Inhaltsverzeichnis

5.22 TesiMod - CAN/CANopen ..... HB 5.22-3

5.22.1 Allgemeines ..... HB 5.22-3
5.22.2 Technische Beschreibung ..... HB 5.22-3
5.22.2.1 Datenobjekte ..... HB 5.22-3
5.22.2.2 Identifier ..... HB 5.22-3
5.22.3 Indirekte Prozeßdaten-Kommunikation ..... HB 5.22-4
5.22.3.1 Ablauf vom Datenaustausch ..... HB 5.22-4
5.22.4 Aufbau von Request- und Responsobjekten ..... HB 5.22-4
5.22.4.1 Aufbau vom Statusbyte ..... HB 5.22-5
5.22.4.2 Indexbytes ..... HB 5.22-6
5.22.4.3 Subindexbyte ..... HB 5.22-6
5.22.4.4 Nutzdatenbytes vom Request- / Responseobjekt ..... HB 5.22-6
5.22.4.5 Responseobjekt mit Fehler ..... HB 5.22-7
5.22.5 Aufgaben vom Kommunikationspartner ..... HB 5.22-7
5.22.5.1 Ablauf bei fehlerfreier Kommunikation ..... HB 5.22-7
5.22.5.2 Ablauf bei fehlerhafter Kommunikation ..... HB 5.22-8
5.22.6 Protokollparameter ..... HB 5.22-8
5.22.6.1 Baudrate ..... HB 5.22-8
5.22.6.2 Wartezeit für Antwort ..... HB 5.22-8
5.22.6.3 Zeit bis zu neuem Verbindungsaufbau ..... HB 5.22-8
5.22.6.4 Terminal-Modulnummer ..... HB 5.22-8
5.22.6.5 CANopen Netzdienste ..... HB 5.22-9
5.22.7 Kommunikationsbeziehungen ..... HB 5.22-9
5.22.7.1 Modus ..... HB 5.22-9
5.22.7.2 Verbindungspartner ..... HB 5.22-9
5.22.7.3 Modulnummer ..... HB 5.22-9
5.22.7.4 Requestidentifier ..... HB 5.22-10
5.22.7.5 Responseidentifier ..... HB 5.22-10
5.22.7.6 Adreßtyp ..... HB 5.22-10
5.22.8 Pollbereich ..... HB 5.22-10
5.22.9 Physikalische Ankopplung ..... HB 5.22-12
5.22.9.1 Verbindungskabel TesiMod Bedienterminal - CAN-Bus ..... HB 5.22-13
5.22.10 Fehlermeldungen ..... HB 5.22-14
5.22.11 Beispiele für Kommunikationsbeziehungen ..... HB 5.22-15

/001-7210/ Technische Änderungen vorbehalten

5.22 TesiMod - CAN/CANopen

Abkürzungen und Begriffe:
CAL CAN-Applikation-Layer, standardisiert durch CiA (CiA Draft Standard 102, 201, 205, 207)
CiA CAN in Automation International Users and Manufacturers Group e.V.
CANopen-Communication-Profile, CiA Draft Standard 301

5.22.1 Allgemeines

TesiMod Bedienterminals lassen sich auf einfache Weise an CAN-Bussysteme ankoppeln. Deshalb sind TesiMod Bedienterminals als Mensch-Maschine-Schnittstelle (MMI) für ihre CAN/CANopen-Applikation bestens geeignet.
Der CAN-Bus ist als High-Speed-Bus nach ISO-DIS 11898 ausgelegt. Alle Verbindungen sind galvanisch getrennt.
Die Anwendung von CAN in TesiMod Bedienterminals lehnt sich an die Spezifikation **CANopen Communication Profile** an.
CANopen verwendet eine Untermenge der Kommunikations-Dienste von CAL.
Bisher wurde von CiA noch kein Kommunikationsprofil für Mensch-Maschine-Schnittstellen (MMI) standardisiert. Deshalb hat die Fa. Sütron electronic GmbH eine eigene **Indirekte Prozeßdaten-Kommunikation** entwickelt.

5.22.2 Technische Beschreibung

Der Datenverkehr wird durch das Kommunikationsprofil des CANopen bestimmt. Basierend darauf werden Profile definiert, über die gleichartige Module gleichartig angesprochen werden können.
Mit Hilfe des Projektiersystems können acht verschiedene Kommunikationsbeziehungen definiert werden.

5.22.2.1 Datenobjekte

Alle Daten, die im CANopen-Netz auszutauschen sind, werden als Datenobjekte über ein Objektverzeichnis angesprochen. Bei CANopen werden Prozeßdatenobjekte (PDO) zusammengestellt. Diese Übertragungspakete können mehrere Datenobjekte (bis 8 Byte) enthalten. PDOs werden für das Senden und Empfangen von Daten zusammengestellt.
PDOs für das Senden von Daten heißen Request-PDO (PDO tx), die für das Empfangen heißen Response-PDO (PDO rx).
Für die Übertragung der PDOs werden Dienste der untersten Ebene der CAL verwendet, die keinerlei Protokoll-Overhead benötigen.

5.22.2.2 Identifier

Bei CANopen wird jedem PDO ein Identifier zugeordnet.
Jeder CANopen-Teilnehmer enthält maximal zwei Sende-PDOs und zwei Empfangs-PDOs, also vier Identifier.
Durch diese Einschränkung können auf einfachem Wege maximal 16 Byte gesendet und 16 Byte empfangen werden.

/001-7210/ Technische Änderungen vorbehalten

5.22.3 Indirekte Prozeßdaten-Kommunikation

Bei der Kommunikation mit Bedienterminals werden im CANopen-Netz Datenmengen ausgetauscht, die größer als 16 Byte sind. In zwei PDOs können diese Datenmengen dann nicht mehr untergebracht werden.
Deshalb wird vom Bedienterminal ein Datenobjekt nicht direkt angesprochen, sondern über spezielle Kommunikationsobjekte, die Request- und Responseobjekte. Diese teilen dem Partner mit, welches Datenobjekt gelesen oder beschrieben werden soll.

5.22.3.1 Ablauf vom Datenaustausch

Die für den Betrieb des Terminals notwendigen Dienste gehen alle vom Bedienterminal aus, das heißt das **Bedienterminal besitzt Client-Funktionalität**.
Der Kommunikationspartner reagiert lediglich auf die Anforderung des Terminals, das heißt der **Kommunikationspartner besitzt Server-Funktionalität**.
Das Bedienterminal sendet also Requestobjekte und der Kommunikationspartner sendet Responseobjekte.

Empfängt der Kommunikationspartner ein Requestobjekt vom Bedienterminal, so interpretiert er die ersten 4 Byte nach Datenrichtung und angesprochenem Datenobjekt.
Bei einer Schreibanforderung speichert er die angekommenen Daten und quittiert mit einem leeren Responseobjekt.
Bei einer Leseanforderung sendet er im Responseobjekt die angeforderten Daten an das Bedienterminal.

Sende-PDO und Empfangs-PDO sind immer paarweise vorhanden. Ein Paar steht für eine Kommunikationsbeziehung zwischen Bedienterminal und **einem** Teilnehmer.
Mit TesiMod-Bedienterminals können bis zu acht Kommunikationsbeziehungen aufgebaut werden.

5.22.4 Aufbau von Request- und Responsobjekten

Die Request- und Responseobjekte sind je 8 Byte groß. In den ersten 4 Byte dieser Objekte wird das angesprochene Datenobjekt definiert und in den restlichen 4 Byte stehen die Daten.

1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte	7. Byte	8. Byte
Status	Index-Low	Index-High	Subindex	Nutzdaten	Nutzdaten	Nutzdaten	Nutzdaten

Bild 1: Aufbau von Request- und Responseobjekten

5.22.4.1 Aufbau vom Statusbyte

Innerhalb von Request- und Responseobjekt hat das Statusbyte folgenden Aufbau:

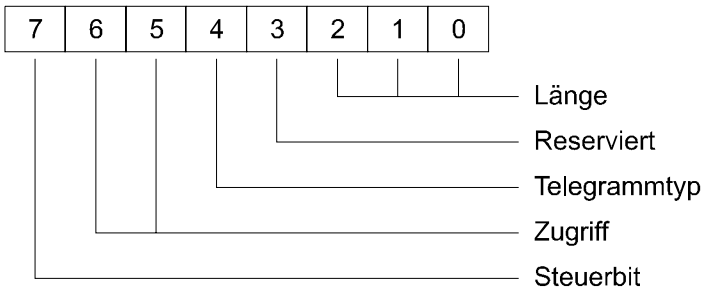


Bild 2: Aufbau vom Steuerbyte

Länge - Bit 0 bis Bit 2

1, 2	Bitzugriff	Länge der Bitmaske zum Setzen oder Löschen von Bits
1, 2, 3, 4	Bytezugriff	Anzahl in Bytes
2, 4	Wortzugriff	Anzahl in Bytes
4	Doppelwortzugriff	Anzahl in Bytes

Telegrammtyp - Bit 4

0	Request - Anforderungstelegramm
1	Response - Antworttelegramm

Zugriff - Bit 5 bis Bit 6

0	Bitzugriff	Schreibender Bitzugriff (setzen / löschen)
1	Bytezugriff	Byte lesen oder schreiben
2	Wortzugriff	Wort lesen oder schreiben
3	Doppelwortzugriff	Doppelwort lesen oder schreiben

Steuerbit - Bit 7

Requestobjekt

0	Schreiben
1	Lesen

Responseobjekt

0	O.K.
1	Fehler aus Steuerung bei Telegrammbearbeitung Die Fehlernummer steht im ersten Datenbyte

/001-7210/ Technische Änderungen vorbehalten

5.22.4.2 Indexbytes

Der Wert spezifiziert den Index aus dem Objektverzeichnis des Kommunikationspartners.
Der Index zeigt auf
- eine Variable ohne Subindex
- den Anfang eines Arrays
- den Anfang eines Records.

Bei Anwendung ohne CANopen kann hier eine beliebige Numerierung zwischen den Kommunikationspartnern vereinbart werden.

5.22.4.3 Subindexbyte

Der Wert spezifiziert den Subindex aus dem Objektverzeichnis des Kommunikationspartners.
Der Subindex zeigt genau auf eine Variable (Basic-CAN-Variable) von der Größe des Zugriffs.

Bei Anwendung ohne CANopen kann hier eine beliebige Numerierung zwischen den Kommunikationspartnern vereinbart werden.

5.22.4.4 Nutzdatenbytes vom Request- / Responseobjekt

Nutzdaten bitorientiert

In einer Byteadresse:

5. Byte	6. Byte	7. Byte	8. Byte
AND-Maske		OR-Maske	

Bild 3: Nutdaten in einer Byteadresse

In einer Wortadresse:

5. Byte	6. Byte	7. Byte	8. Byte
AND-Maske LOW-Byte	AND-Maske HIGH-Byte	OR-Maske LOW-Byte	OR-Maske HIGH-Byte

Bild 4: Nutzdaten in einer Wortadresse

Die Bitmasken sind in der Steuerung mit der angegebenen Adresse zu verODERn und zu verUNDen, wobei die AND-Maske Bits löscht und die ODER-Maske Bits setzt.

Nutzdaten byteorientiert

5. Byte	6. Byte	7. Byte	8. Byte
1. Byte	2. Byte	3. Byte	4. Byte

Bild 5: Nutzdaten byteorientiert

Nutzdaten wortorientiert

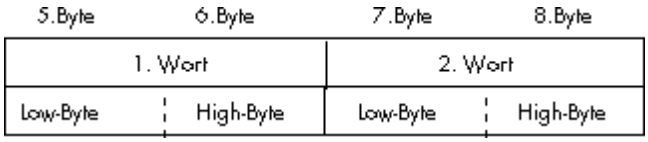


Bild 6: Nutzdaten wortorientiert

Nutzdaten doppelwortorientiert

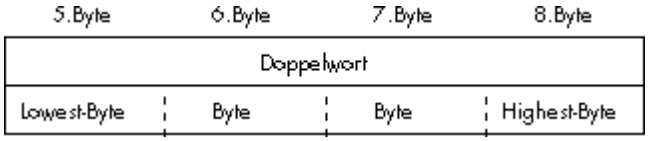


Bild 7: Nutzdaten doppelwortorientiert

5.22.4.5 Responseobjekt mit Fehler

Im Falle eines Kommunikationsfehlers wird ein Responseobjekt mit folgendem Aufbau zurückgesendet:

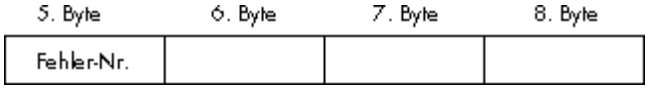


Bild 8: Aufbau Responseobjekt mit Fehler

5.22.5 Aufgaben vom Kommunikationspartner

Der Kommunikationspartner des Bedienterminals hat Serverfunktionalität. Als Server muß der Kommunikationspartner die eingehenden Telegramme interpretieren und mit einem Responseobjekt beantworten. Dabei muß die in den Protokollparametern eingestellte Wartezeit eingehalten werden. Jedes Requestobjekt muß mit einem Responseobjekt beantwortet werden.
Im Fehlerfall sendet der Kommunikationspartner gleichzeitig eine Fehlernummer an das Bedienterminal.

5.22.5.1 Ablauf bei fehlerfreier Kommunikation

- Das Statusbyte des Requestobjekts wird mit folgender Änderung in das Responseobjekt umkopiert:
  Bit 7 Steuerbit = 0 fehlerfreie Aktion
  Bit 4 Telegrammtyp = 1 Responseobjekt
- Index und Subindex werden vom Requestobjekt in das Responseobjekt umkopiert.
- Schreibt das Bedienterminal, so sind die Nutzdaten auf die angesprochene Adresse zu kopieren
- Liest das Bedienterminal Daten, so sind die Daten von der angesprochenen Adresse an die Adresse der Nutzdatenbytes zu kopieren.

/001-7210/ Technische Änderungen vorbehalten

5.22.5.2 Ablauf bei fehlerhafter Kommunikation

Stellt der Kommunikationspartner einen Fehler im Inhalt vom Requestobjekt fest, so kann dies durch eine Fehlermeldung im Responseobjekt dem Bedienterminal mitgeteilt werden.
Die Fehlernummern können vom Anwender frei definiert werden. Grundsätzlich zeigt das Bedienterminal einen Communication-Error 100 am Display an. Der Subcode der Fehlermeldung enthält die Fehlernummer.

- Das Statusbyte des Requestobjekts wird mit folgender Änderung in das Responseobjekt umkopiert:
  Bit 7 Steuerbit = 1 Fehler erkannt
  Bit 4 Telegrammtyp = 1 Responseobjekt
- Index und Subindex werden vom Requestobjekt in das Responseobjekt umkopiert.
- Fehlernummer in Byte 5 eintragen (wird als Subcode am Bedienterminal angezeigt)

5.22.6 Protokollparameter

5.22.6.1 Baudrate

Für die Baudrate können folgende Einstellungen ausgewählt werden:
- 20 kBaud
- 125 kBaud
- 500 kBaud
- 1 MBaud

Grundsätzlich wird die Baudrate von 20 kBit/s vorgegeben.

5.22.6.2 Wartezeit für Antwort

Innerhalb der Wartezeit für Anwort muß der Kommunikationspartner mit einem Response-PDO geantwortet haben. Andernfalls wird eine entsprechende Fehlermeldung ausgegeben.
Die Wartezeit kann zwischen 0 ms und 65536 ms liegen.
Als Grundeinstellung wird der Wert 1000 ms eingetragen.

5.22.6.3 Zeit bis zu neuem Verbindungsaufbau

Das TesiMod-Bedienterminal wartet mit der Aufnahme der Kommunikation, bis die eingestellte Zeit verstrichen ist. Dadurch wird vermieden, daß eine fehlerhafte Kommunikation aufgebaut wird, bevor das Bussystem betriebsbereit ist.
Die Wartezeit kann zwischen 1 s und 255 s eingestellt werden.
Als Grundwert wird eine Zeit von 5 s eingetragen.

5.22.6.4 Terminal-Modulnummer

Im Modus CANopen wird mit Hilfe der Modulnummer die Vergabe der Identifier für Request und Response berechnet.
Für die Modulnummern können Werte zwischen 0 und 127 eingetragen werden.
Als Grundwert wird der Wert 1 vorgegeben.

5.22.6.5 CANopen Netzdienste

Der Anwender kann im Projektiersystem definieren, ob CANopen Netzdienste wahrgenommen werden sollen.

- NMT-Dienst mit Identifier = 0
  - Byte 1 = 1 Aktivieren des Teilnehmers (Knotens)
  - Byte 1 = 2 Deaktivieren des Teilnehmers (Knotens)
  - Byte 1 = 3 Deaktivieren des Teilnehmers (Knotens)
  - Byte 1 = 128 Deaktivieren des Teilnehmers (Knotens)
  - Byte 1 = 129 Reset des Teilnehmers (Knotens)
  - Byte 1 = 130 Kommunikation beim Teilnehmer (Knoten) zurücksetzen
- Das Bedienterminal sendet beim Hochlaufen eine Emergency-Nachricht mit Identifier = 128 + Terminal-Modulnummer

5.22.7 Kommunikationsbeziehungen

Für TesiMod Bedienterminals können acht unabhängige Kommunikationsbeziehungen definiert werden. Sehen Sie dazu das Bild 10 im Kapitel 5.22.11.

5.22.7.1 Modus

Die Vergabe der Identifier für die Kommunikationsbeziehungen kann auf zwei verschiedenen Wegen vorgenommen werden:

- Manuell: Der Request- und Responseidentifier wird direkt in den entsprechende Feldern eingetragen.
- CANopen: In diesem Fall richtet sich die Vergabe der Identifier nach dem Verteilschlüssel der Definition nach CANopen-Communikation-Profile DS301.
  Zu einer Basisnummer wird die Modulnummer des Slave, des Kommunikationspartners im CANopen-Netz, addiert und als Identifier eingetragen.

5.22.7.2 Verbindungspartner

Die Auswahl eines Verbindungspartners kann nur im CANopen-Modus erfolgen.
Durch die Auswahl des Verbindungspartners wird der Basis-Identifier bestimmt. Als Verbindungspartner kann entweder **Master** oder **Slave** ausgewählt werden.

- Master: Das Terminal kommuniziert mit dem CANopen-Netzmaster.
  Es wird der erste PDO-Kanal verwendet.
  Zur Basisnummer wird die Modulnummer vom Bedienterminal addiert.
- Slave: Das Bedienterminal kommuniziert mit einem anderen Slave im CANopen-Netz.
  Es wird der zweite PDO-Kanal verwendet.
  Zum Basisidentifier wird die Modulnummer des anderen Slaves addiert.

5.22.7.3 Modulnummer

Die Eingabe einer Modulnummer kann nur im CANopen-Modus in Verbindung mit einer Kommunikationsbeziehung zu einem anderen Slave erfolgen.
Hier kann direkt die Modulnummer des Kommunikationspartners eingegeben werden.

/001-7210/ Technische Änderungen vorbehalten

5.22.7.4 Requestidentifier

Die Eingabe eines Requestidentifiers kann nur im Manuell-Modus erfolgen.
Während im CANopen-Modus der Requestidentifier aus Basisidentifier und Modulnummer berechnet wird, kann im Manuell-Modus direkt der Identifier für das Requestobjekt eingegeben werden.

5.22.7.5 Responseidentifier

Die Eingabe eines Responseidentifiers kann nur im Manuell-Modus erfolgen.
Während im CANopen-Modus der Responseidentifier aus Basisidentifier und Modulnummer berechnet wird, kann im Manuell-Modus direkt der Identifier für das Responseobjekt eingegeben werden.

5.22.7.6 Adreßtyp

Beim Zugriff auf Daten des Kommunikationspartners muß dem Bedienterminal für interne Adreßberechnungen bekannt sein, welche Adressierungsart der Datenbereich hat, auf den zugegriffen wird.
Folgende Werte können für den Adreßtyp eingetragen werden:
- Byteadresse 1
- Wortadresse 2
- Doppelwortadresse 4

5.22.8 Pollbereich

Für den Datenbereich des Pollbereichs sind folgende Bedingungen zu beachten:
- Die Steuerung muß einzelne Bits manipulieren können (Ansteuerung der LEDs)
- Das Bedienterminal muß diesen Datenbereich wortweise lesen können
- Vom Bedienterminal ist **nur Wortzugriff** erlaubt
- Der Adreßtyp des Datenbereichs in der Steuerung muß eine **Byte- oder Wortadresse** sein

Beispiel für die Adressierung des Pollbereichs:

Byteadressiert

Wort-Adresse + 0	Index 100	Subindex 0
Wort-Adresse + 1	Index 100	Subindex 2
Wort-Adresse + 2	Index 100	Subindex 4
Wort-Adresse + 3	Index 100	Subindex 6
Wort-Adresse + 4	Index 100	Subindex 8
Wort-Adresse + 5	Index 100	Subindex 10

Wortadressiert

Wort-Adresse + 0	Index 100	Subindex 0
Wort-Adresse + 1	Index 100	Subindex 1
Wort-Adresse + 2	Index 100	Subindex 2
Wort-Adresse + 3	Index 100	Subindex 3
Wort-Adresse + 4	Index 100	Subindex 4
Wort-Adresse + 5	Index 100	Subindex 5

	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Wort-Adr. + 0	frei	frei	frei	DDF	LM	PL	RQ	ED	frei	frei	frei	frei	frei	frei	frei	frei
Wort-Adr. + 1	Sequentieller Meldekanal High-Byte								Sequentieller Meldekanal Low-Byte							
Wort-Adr. + 2	LED 1 ein/aus	LED 1 blinken	LED 2 ein/aus	LED 2 blinken	LED 3 ein/aus	LED 3 blinken	LED 4 ein/aus	LED 4 blinken	LED 5 ein/aus	LED 5 blinken	LED 6 ein/aus	LED 6 blinken	LED 7 ein/aus	LED 7 blinken	LED 8 ein/aus	LED 8 blinken
Wort-Adr. + 3	LED 9 ein/aus	LED 9 blinken	LED 10 ein/aus	LED 10 blinken	LED 11 ein/aus	LED 11 blinken	LED 12 ein/aus	LED 12 blinken	LED 13 ein/aus	LED 13 blinken	LED 14 ein/aus	LED 14 blinken	LED 15 ein/aus	LED 15 blinken	LED 16 ein/aus	LED 16 blinken
Wort-Adr. + 4	LED 17 ein/aus	LED 17 blinken	LED 18 ein/aus	LED 18 blinken	LED 19 ein/aus	LED 19 blinken	LED 20 ein/aus	LED 20 blinken	LED 21 ein/aus	LED 21 blinken	LED 22 ein/aus	LED 22 blinken	LED 23 ein/aus	LED 23 blinken	LED 24 ein/aus	LED 24 blinken
Wort-Adr. + 5	LED 25 ein/aus	LED 25 blinken	LED 26 ein/aus	LED 26 blinken	LED 27 ein/aus	LED 27 blinken	LED 28 ein/aus	LED 28 blinken	LED 29 ein/aus	LED 29 blinken	LED 30 ein/aus	LED 30 blinken	LED 31 ein/aus	LED 31 blinken	LED 32 ein/aus	LED 32 blinken
Wort-Adr. + 6	LED 33 ein/aus	LED 33 blinken	LED 34 ein/aus	LED 34 blinken	LED 35 ein/aus	LED 35 blinken	LED 36 ein/aus	LED 36 blinken	LED 37 ein/aus	LED 37 blinken	LED 38 ein/aus	LED 38 blinken	LED 39 ein/aus	LED 39 blinken	LED 40 ein/aus	LED 40 blinken
Wort-Adr. + 7	LED 41 ein/aus	LED 41 blinken	LED 42 ein/aus	LED 42 blinken	LED 43 ein/aus	LED 43 blinken	LED 44 ein/aus	LED 44 blinken	LED 45 ein/aus	LED 45 blinken	LED 46 ein/aus	LED 46 blinken	LED 47 ein/aus	LED 47 blinken	LED 48 ein/aus	LED 48 blinken

Bild 9: Pollbereich wortorientiert

/001-7210/ Technische Änderungen vorbehalten

5.22.9 Physikalische Ankopplung

Für eine Integration eines Bedienterminals in eine CAN-Struktur stehen die optoentkoppelten Schnittstellen X2.1 und X2.2 zur Verfügung. Der CAN-Bus ist als High-Speed-Bus nach ISO-DIS 11898 ausgelegt.

Für die Verbindungsleitungen kommen abgeschirmte Kabel mit paarverseilten Adern in Frage. Der CAN-Bus muß an beiden Enden mit Terminationswiderständen abgeschlossen werden.
Die Belegung der Schnittstellenstecker am Bedienterminal entsprechen CiA Draft Standard 102.

Steckverbinder im Terminal: 9-polige SubminD Buchsenleiste X2.1

Belegung:

Pin	Bel.	Bezeichnung	Funktion
1	x	res	Reserviert
2	x	CANL	CAN_L Busleitung (Dominant LOW)
3	x	0V_C	CAN Ground
4	x	res	Reserviert
5	x	res	Reserviert
6	x	0V_C	CAN Ground
7	x	CANH	CAN_H Busleitung (Dominant HIGH)
8	x	res	Reserviert
9	x	res	Reserviert.

Steckverbinder im Terminal: 9-polige SubminD Stiftleiste X2.2

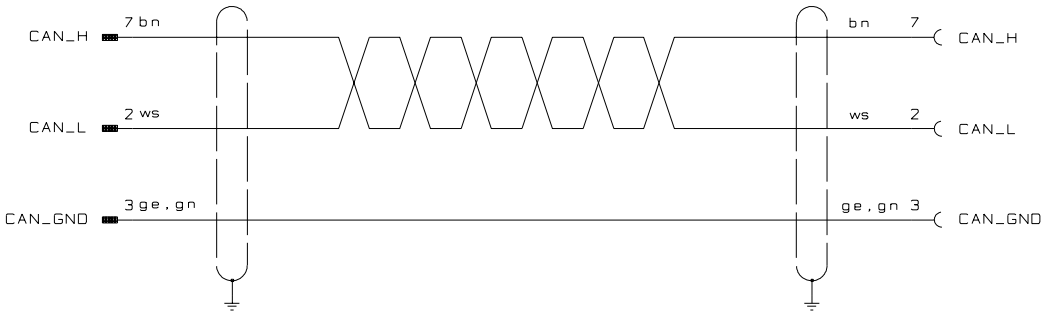
Belegung:

Pin	Bel.	Bezeichnung	Funktion
1	x	res	Reserviert
2	x	CANL	CAN_L Busleitung (Dominant LOW)
3	x	0V_C	CAN Ground
4	x	res	Reserviert
5	x	res	Reserviert
6	x	0V_C	CAN Ground
7	x	CANH	CAN_H Busleitung (Dominant HIGH)
8	x	res	Reserviert
9	x	res	Reserviert

Alle Signalleitungen sind von X2.1 zu X2.2 gebrückt. Die Verbindungskabel sollten an alle Pins, auch die reservierten, angeschlossen werden. Dadurch können die Kabel auch bei zukünftigen Erweiterungen der Busspezifikationen noch eingesetzt werden.

5.22.9.1 Verbindungskabel TesiMod Bedienterminal - CAN-Bus

TesiMod Bedienterminal	Nächster CAN-Bus-Teilnehmer



D-Subminiatur Stiftstecker 9-polig	D-Subminiatur Buchsenstecker 9-polig

Schirm liegt beidseitig auf Metallgehäuse.

Das Kabel ist entgegen den Empfehlungen des CiA Draft Standard 102 nur mit den Verbindungen ausgestattet, die, nach den derzeitigen Anforderungen, für eine Kommunikation nötig sind.

/001-7210/ Technische Änderungen vorbehalten

5.22.10 Fehlermeldungen

Code			
	1	E_SLAVE_NOT_READY	Slave zur Zeit nicht bereit
	2	E_PROTOKOL	Reihenfolge der Pakete
	3	E_FRAME	Protokollrahmen-Fehler
	4	E_TIMEOUT	Timeout-Fehler
	5	E_CRC_BCC	CRC-Fehler
	6	E_PARITY	Parity-Fehler
	7	E_SEND_ABORT	Sendeabbruch
	8	R_REC_ABORT	Empfangsabbruch
	9	E_BUF_SIZE	Zyklischer Puffer zu klein
	10	E_NO_DEFINE	Keine zyklischen Daten definiert
	12	E_DEFINE	Zyklische Daten bereits definiert.
	15	E_NO_PROTOCOL	Gewähltes Protokoll wird nicht unterstützt
	16	E_OVERRUN	Empfangsüberlauf
	40	E_SYS_ADDRESS	Undefinierte Systemvariable
	50	E_CAN_ERROR	Fehler aus CAN-Controller Fehlernummer im Subcode
Subcode	1		Stuff-Error
	3		Terminal hat keine Verbindung zum Bus Am Bus sind keine Teilnehmer angeschlossen
	4, 5		Busleitung hat Schluß nach logisch 0 oder 1
	6		CRC-Error
Code			
	51	E_RESPONSE-TIMEOUT	Keine Antwort vom Kommunikationspartner Kein Partner für Requestobjekt Kein Empfänger für gesendeten Identifier
	53	E_NO_RESPONSE_BIT	Falsches Responseobjekt Nachricht ohne Response-Bit
	54	E_RESPONSE OF NO_REQUEST	Falsches Responseobjekt Response ohne Request
	55	E_NO_HARDWARE	Terminal findet keine CAN-Hardware CAN-Hardware in Terminal fehlt oder ist defekt
	60	E_RESET_FROM_MASTER	NMT 0 Message mit Kommando 129
	100	E_DATA_ERROR	Fehler vom Kommunikationspartner Fehlernummer steht im Subcode Subcode ist frei definierbar.

5.22.11 Beispiele für Kommunikationsbeziehungen

Mit Hilfe des Projektiersystems TSwin lassen sich auf übersichtliche Weise acht Kommunikationsbeziehungen definieren. Das folgende Bild zeigt das Fenster von TSwin, in dem die Beziehungen definiert werden und macht gleichzeitig deutlich, welche Einstellungen möglich sind.

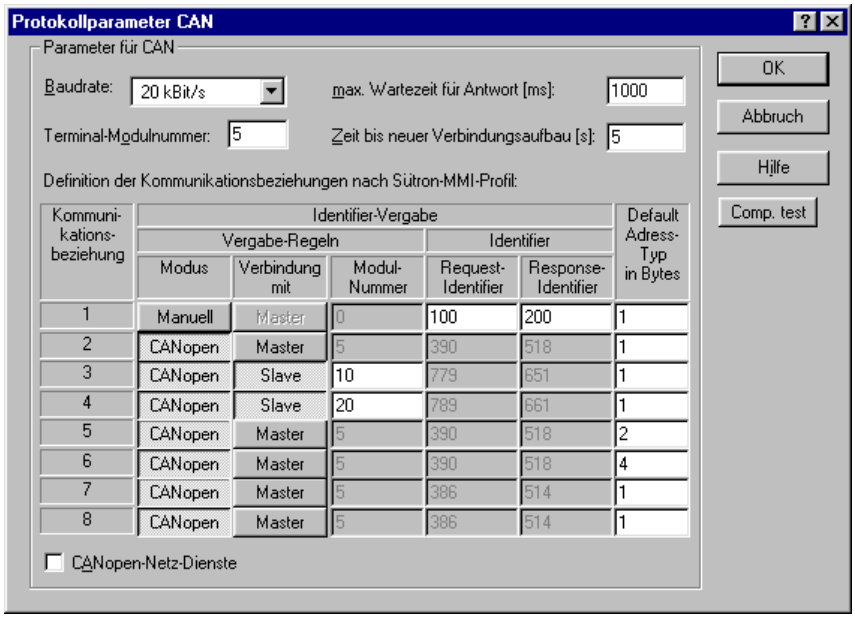
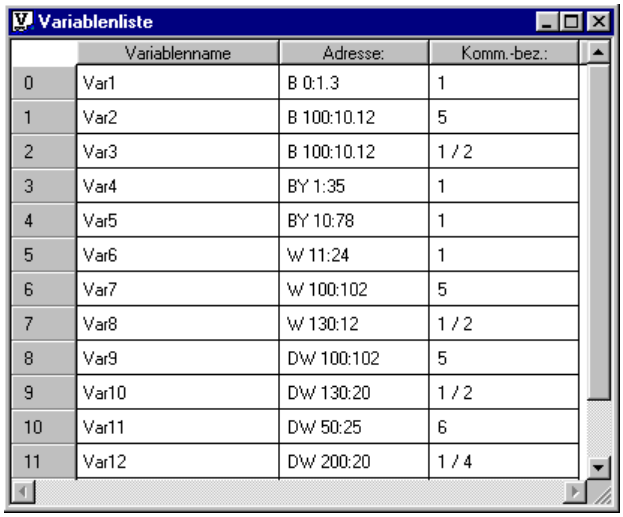


Bild 10: Einstellungen der Kommunikationsbeziehungen in TSwin als Beispiel

Passend zu den oben dargestellten Kommunikationsbeziehungen ist hier eine Variablentabelle dargestellt, wie sie ebenfalls mit TSwin erstellt wird. In der rechten Spalte der Tabelle werden die Kommunikationsbeziehungen eingetragen.



	Variablenname	Adresse:	Komm.-bez.:
0	Var1	B 0:1.3	1
1	Var2	B 100:10.12	5
2	Var3	B 100:10.12	1 / 2
3	Var4	BY 1:35	1
4	Var5	BY 10:78	1
5	Var6	W 11:24	1
6	Var7	W 100:102	5
7	Var8	W 130:12	1 / 2
8	Var9	DW 100:102	5
9	Var10	DW 130:20	1 / 2
10	Var11	DW 50:25	6
11	Var12	DW 200:20	1 / 4

Bild 11: Variablenliste in TSwin als Beispiel

/001-7210/ Technische Änderungen vorbehalten

In der Kommunikations-Beziehung 1 gilt folgendes:
- Index 0 - 99 byteadressiert,
- Index 100 - 199 wortadressiert
- Index 200 -300 doppelwortadressiert.

Variablenname	Zugriff	Index	Subindex	Bitnummer	Komm.-Beziehg.	Adreßtyp
Var1	Bit	0	1	3	1	

Bitzugriff: Bit 3 von Index 0, Subindex 1 von Kommunikations-Beziehung 1.
Ist für Adress-Typ nichts eingetragen, so gilt der Defaultwert in der Tabelle “Definition der Kommunikations-Beziehungen”, also hier bei Index 0 Byteadresse.

Variablenname	Zugriff	Index	Subindex	Bitnummer	Komm.-Beziehg.	Adreßtyp
Var3	Bit	100	10	12	1	2

Bitzugriff: Bit 12 von Index 100, Subindex 10 von Kommunikations-Beziehung 1.
Index 100 in der Kommunikations-Beziehung 1 ist eine Wortadresse, default ist aber Byteadresse eingestellt. Deshalb muß hier eine 2 bei Adress-Typ eingegeben werden.

Variablenname	Zugriff	Index	Subindex	Bitnummer	Komm.-Beziehg.	Adreßtyp
Var2	Bit	100	10	12	5	

Bitzugriff: Bit 12 von Index 100, Subindex 10 von Kommunikations-Beziehung 5.

Variablenname	Zugriff	Index	Subindex	Bitnummer	Komm.-Beziehg.	Adreßtyp
Var5	Byte	10	78		1	

Bytezugriff: Index 10, Subindex 78, Kommunikations-Beziehung 1

Variablenname	Zugriff	Index	Subindex	Bitnummer	Komm.-Beziehg.	Adreßtyp
Var8	Wort	130	12		1	2

Wortzugriff: Index 130, Subindex 12, Kommunikations-Beziehung 1
Index 100- 199 in Komm.-Bez. 1 ist eine Wortadresse, default ist aber Byteadresse eingestellt. Deshalb muß hier eine 2 bei Adress-Typ eingegeben werden.

Variablenname	Zugriff	Index	Subindex	Bitnummer	Komm.-Beziehg.	Adreßtyp
Var12	Doppelwort	200	20	3	1	4

Doppelwortzugriff: Index 200, Subindex 20, Kommunikations-Beziehung 1
Index 200- 300 in Komm.-Bez. 1 ist eine Doppelwortadresse, default ist aber Byteadresse eingestellt. Deshalb muß hier eine 4 bei Adress-Typ eingegeben werden.

Variablenname	Zugriff	Index	Subindex	Bitnummer	Komm.-Beziehg.	Adreßtyp
Var11	Doppelwort	50	25		6	

Doppelwortzugriff: Index 50, Subindex 25, Kommunikations-Beziehung 6