



Deuschmann

your ticket to all buses

**Bedienerhandbuch
Universal Feldbus-Gateway
UNIGATE® CL - CANopen**



1	Hinweise zur CE-Kennzeichnung der Baugruppe	8
1.1	EU-Richtlinie EMV	8
1.2	Einsatzbereich	8
1.3	Aufbaurichtlinien beachten	8
1.4	Einbau des Gerätes	8
1.5	Arbeiten an Schaltschränken	8
2	Hinweise für den Hersteller von Maschinen	9
2.1	Einleitung	9
2.2	EU-Richtlinie Maschinen	9
3	Einführung	10
3.1	UNIGATE® CL Software Flussdiagramm	11
3.2	UNIGATE® Blockdiagramm	12
3.3	UNIGATE® Applikationsdiagramm	12
4	Die Betriebsmodi des Gateway	13
4.1	Konfigurationsmodus (config mode)	13
4.2	Testmodus (test mode)	13
4.3	Datenaustauschmodus (data exchange mode)	13
5	RS-Schnittstelle	14
5.1	RS-Schnittstellen beim UNIGATE® CL	14
5.2	Puffergrößen beim UNIGATE® CL	14
5.3	Framing Check	14
6	SSI-Schnittstelle	15
6.1	Inbetriebnahme der SSI-Schnittstelle	15
6.2	Parameter	15
6.2.1	Resolution	15
6.2.2	SSI Encoder Type	15
6.2.3	Parameter Abtastfrequenz (Clock stretch)	15
6.2.4	Parameter Geberüberwachung (Check Encoder)	16
6.3	Hardware- Beschaltung	16
7	Die Debug Schnittstelle	17
7.1	Übersicht über die Debug Schnittstelle	17
7.2	Start im Debugmode	17
7.3	Kommunikationsparameter für die Debugschnittstelle	17
7.4	Möglichkeiten mit der Debug Schnittstelle	17
7.5	Befehle der Debug Schnittstelle	17
8	Funktionsweise des Systems	18
8.1	Allgemeine Erläuterung	18
8.2	Schnittstellen	18
8.3	Datenaustausch CANopen® V3	18
8.3.1	SDO-Zugriff	18
8.3.2	PDO-Zugriff	18
8.4	Mögliche Datenlängen	18
9	Erstellung eines Scripts	19
9.1	Was ist ein Script	19
9.2	Speichereffizienz der Programme	19

9.3	Was kann man mit einem Script Gerät machen	19
9.4	Unabhängigkeit von Bussen	19
9.5	Weitere Einstellungen am Gateway	19
9.6	Die Benutzung des Protocol Developer	20
9.7	Genauigkeiten der Baudraten	20
9.8	Scriptarbeitszeiten	21
10	Implementierte Protokolle mit Universalscript - Konfiguration	22
11	Implementierte Protokolle mit Universalscript -	
	Protokolle Applikation	22
11.1	Protokoll:Transparent	22
11.1.0.1	Datenaufbau	22
11.2	Protokoll: Universal 232	23
11.2.1	Datenaufbau	23
11.2.2	Parameter Feldbus	23
11.2.3	Parametertabelle RS232	23
11.2.3.1	Startzeichen (232 Start character).	23
11.2.3.2	Länge232 (232 Length)	23
11.2.3.3	Timeout	23
11.2.3.4	Datenbereich	24
11.2.3.5	Endezeichen (232 End character).	24
11.2.4	Kommunikationsablauf	24
11.3	Protokoll: 3964(R)	24
11.3.1	Datenaufbau 3964R	24
11.3.2	Protokollfestlegungen	24
11.3.3	Datenverkehr	25
11.3.3.1	Einleitung des Datenverkehrs durch den niederprioren Teilnehmer.	25
11.3.4	Konfliktfälle	25
11.3.4.1	Überwachungszeiten	25
11.3.4.2	Wiederholungen	25
11.3.4.3	Einleitung des Datenverkehrs durch den hochprioren Teilnehmer	25
11.3.5	Protokolltyp 3964	25
11.4	Protokoll: Modbus-RTU	25
11.4.1	Hinweise	25
11.4.2	UNIGATE® als Modbus-Master	26
11.4.2.1	Vorbereitung	26
11.4.2.2	Datenaufbau	26
11.4.2.3	Kommunikationsablauf	26
11.4.3	UNIGATE® als Modbus-Slave	27
11.4.3.1	Vorbereitung	27
11.4.3.2	Datenaufbau	27
11.4.3.3	Kommunikationsablauf	27
11.4.4	UNIGATE® als Modbus-ASCII Master	27
11.5	Protokoll „Universal Modbus RTU Slave“	27
11.5.1	Datenaufbau Feldbusseite z.B.: PROFIBUS	28
11.5.1.1	Beispiel: FC1 + FC2	28
11.5.1.2	Beispiel: FC3 (Read Holding Register) + FC4 (Read Input Register)	29
11.5.1.3	Beispiel: Schreibe Single Coil FC5	30

11.5.1.4	Beispiel: Write Single Register FC6	31
11.5.1.5	Beispiel: Force multiple coils FC 15.	32
11.5.1.6	Beispiel: Preset multiple register FC16	32
11.6	Protokoll „Universal Modbus RTU Master“	33
11.6.1	Datenaufbau Feldbusseite (z.B. PROFIBUS):	33
11.6.2	Datenaufbau Applikationsseite:	33
11.6.3	Konfiguration: über Wingate ab wcf Datei Version 396	34
11.6.3.1	Beispiel: Read coil status FC1	35
11.6.3.2	Beispiel: Read input status FC2	36
11.6.3.3	Beispiel: Read multiple register FC3	37
11.6.3.4	Beispiel: Read input registers FC4	38
11.6.3.5	Beispiel: Force single coil FC5	38
11.6.3.6	Beispiel: Preset single register FC6.	39
11.6.3.7	Beispiel: Force multiple coils FC15	39
11.6.3.8	Beispiel: Preset multiple register FC16	40
11.7	Protokoll „Universal Modbus ASCII Master/Slave“	41
11.8	Protokoll Delta exchange	41
11.9	Protokoll SSI	41
12	Implementierte Protokolle mit Universalscript - Protokolle CANopen/	
	CAN Layer 2	42
12.1	Protokoll CANopen (default)	42
12.2	Protokoll CAN 2.0A	42
12.3	Protokoll CAN 2.0A with ID-filter	42
12.4	Protokoll CAN 2.0B	43
12.5	Protokoll CAN 2.0B with ID-filter	45
13	Auslieferungszustand	45
14	Das Triggerbyte	45
15	Das Längenbyte	45
16	Hardware-Anschlüsse, Schalter und Leuchtdioden	46
16.1	Gerätebeschriftung	46
16.2	Stecker	46
16.2.1	Stecker zum externen Gerät (RS-Schnittstelle)	46
16.2.2	Stecker Versorgungsspannung und DEBUG-Schnittstelle	47
16.2.3	CANopen®-Stecker	47
16.2.4	Stromversorgung	47
16.3	Leuchtanzeigen	47
16.3.1	LED “(Bus) Power“	48
16.3.2	LED “(Bus) State“	48
16.3.3	LED “Power“	49
16.3.4	LED “State“	49
16.3.5	LEDs 1 / 2 / 4 / 8 (Error No / Select ID)	49
16.4	Schalter	49
16.4.1	Termination Rx 422 + Tx 422 (serielle Schnittstelle)	50
16.4.2	Drehcodierschalter S4 + S5 (serielle Schnittstelle)	50
16.4.3	Termination (CANopen®)	50
16.4.4	DIP-Switch	50
16.5	Debug-Kabel für UNIGATE® CL	51

17 Fehlerbehandlung	52
17.1 Fehlerbehandlung beim UNIGATE® CL	52
18 Aufbaurichtlinien	54
18.1 Montage der Baugruppe	54
18.1.1 Montage	54
18.1.2 Demontage	54
18.2 Verdrahtung	54
18.2.1 Anschlusstechniken	54
18.2.1.1 Stromversorgung	55
18.2.1.2 Anschluss des Potentialausgleichs	55
18.2.2 Kommunikationsschnittstelle CANopen®	55
18.2.2.1 Busleitung mit Kupferkabel	55
18.2.3 Leitungsführung, Schirmung und Maßnahmen gegen Störspannung	55
18.2.4 Allgemeines zur Leitungsführung	55
18.2.4.1 Schirmung von Leitungen	56
19 CANopen®	57
19.1 Beschreibung CANopen®	57
19.1.1 CANopen® V3	57
19.1.2 CANopen® V4	58
20 Technische Daten	60
20.1 Gerätedaten	60
20.1.1 Schnittstellendaten	61
21 Inbetriebnahmeleitfaden	62
21.1 Beachte	62
21.2 Komponenten	62
21.3 Montage	62
21.4 Maßzeichnung UNIGATE® CL-CANopen®	62
21.5 Inbetriebnahme	63
21.6 CANopen®-Adresse und Baudrate einstellen	63
21.7 CANopen®-Anschluss	63
21.8 Anschluss des Prozessgerätes	63
21.9 Versorgungsspannung anschließen	63
21.10 Schirmanschluss	63
21.11 Projektierung	63
22 Service	64
22.1 Einsendung eines Gerätes	64
22.2 Download von PC-Software	64
23 Anhang	65
23.1 Erläuterung der Abkürzungen	65
23.2 Hexadezimal-Tabelle	66

Haftungsausschluss

Wir haben den Inhalt der Druckschrift auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so dass wir für die vollständige Übereinstimmung keine Gewähr übernehmen. Die Angaben in der Druckschrift werden jedoch regelmäßig überprüft. Notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten. Für Verbesserungsvorschläge sind wir dankbar.

Copyright

Copyright (C) Deutschmann Automation GmbH & Co. KG 1997 – 2022. All rights reserved.

Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung Ihres Inhalts sind nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte vorbehalten, insbesondere für den Fall der Patenterteilung oder PM-Eintragung.

1 Hinweise zur CE-Kennzeichnung der Baugruppe

1.1 EU-Richtlinie EMV

Für die in dieser Betriebsanleitung beschriebenen Baugruppe gilt:

Produkte, die das CE-Kennzeichen tragen, erfüllen die Anforderungen der EU Richtlinie "Elektromagnetische Verträglichkeit" und die dort aufgeführten harmonisierten europäischen Normen (EN).

Die EU-Konformitätserklärungen werden gemäß der EU-Richtlinie, Artikel 10, für die zuständigen Behörden zur Verfügung gehalten bei:

Deutschmann Automation GmbH & Co. KG, Carl-Zeiss-Str. 8, 65520 Bad Camberg

1.2 Einsatzbereich

Die Baugruppen sind ausgelegt für den Einsatz im Industriebereich und erfüllen die folgenden Anforderungen.

Einsatzbereich	Anforderung an	
	Störaussendung	Störfestigkeit
Industrie	EN 55011, cl. A (2007)	EN 61000-6-2 (2005)

1.3 Aufbaurichtlinien beachten

Die Baugruppe erfüllt die Anforderungen, wenn Sie

1. bei Installation und Betrieb die in der Betriebsanleitung beschriebenen Aufbaurichtlinien einhalten.
2. zusätzlich die folgenden Regeln zum Einbau des Gerätes und zum Arbeiten an Schaltschränken beachten.

1.4 Einbau des Gerätes

Baugruppen müssen in elektrischen Betriebsmittelräumen oder in geschlossen Gehäusen (z.B. Schaltkästen aus Metall oder Kunststoff) installiert werden. Ferner müssen Sie das Gerät und den Schaltkasten (Metallkasten), oder zumindest die Hutschiene (Kunststoffkasten), auf die die Baugruppe aufgeschnappt wurde, erden.

1.5 Arbeiten an Schaltschränken

Zum Schutz der Baugruppen vor Entladung von statischer Elektrizität muss sich das Personal vor dem Öffnen von Schaltschränken bzw. Schaltkästen elektrostatisch entladen.

2 Hinweise für den Hersteller von Maschinen

2.1 Einleitung

Die Baugruppe UNIGATE® stellt keine Maschine im Sinne der EU-Richtlinie "Maschinen" dar. Für die Baugruppe gibt es deshalb keine Konformitätserklärung bezüglich der EU-Richtlinie Maschinen.

2.2 EU-Richtlinie Maschinen

Die EU-Richtlinie Maschinen regelt die Anforderungen an eine Maschine. Unter einer Maschine wird hier eine Gesamtheit von verbundenen Teilen oder Vorrichtungen verstanden (siehe auch EN 292-1, Absatz 3.1)

Die Baugruppe ist ein Teil der elektrischen Ausrüstung einer Maschine und muss deshalb vom Maschinenhersteller in das Verfahren zur Konformitätserklärung einbezogen werden.

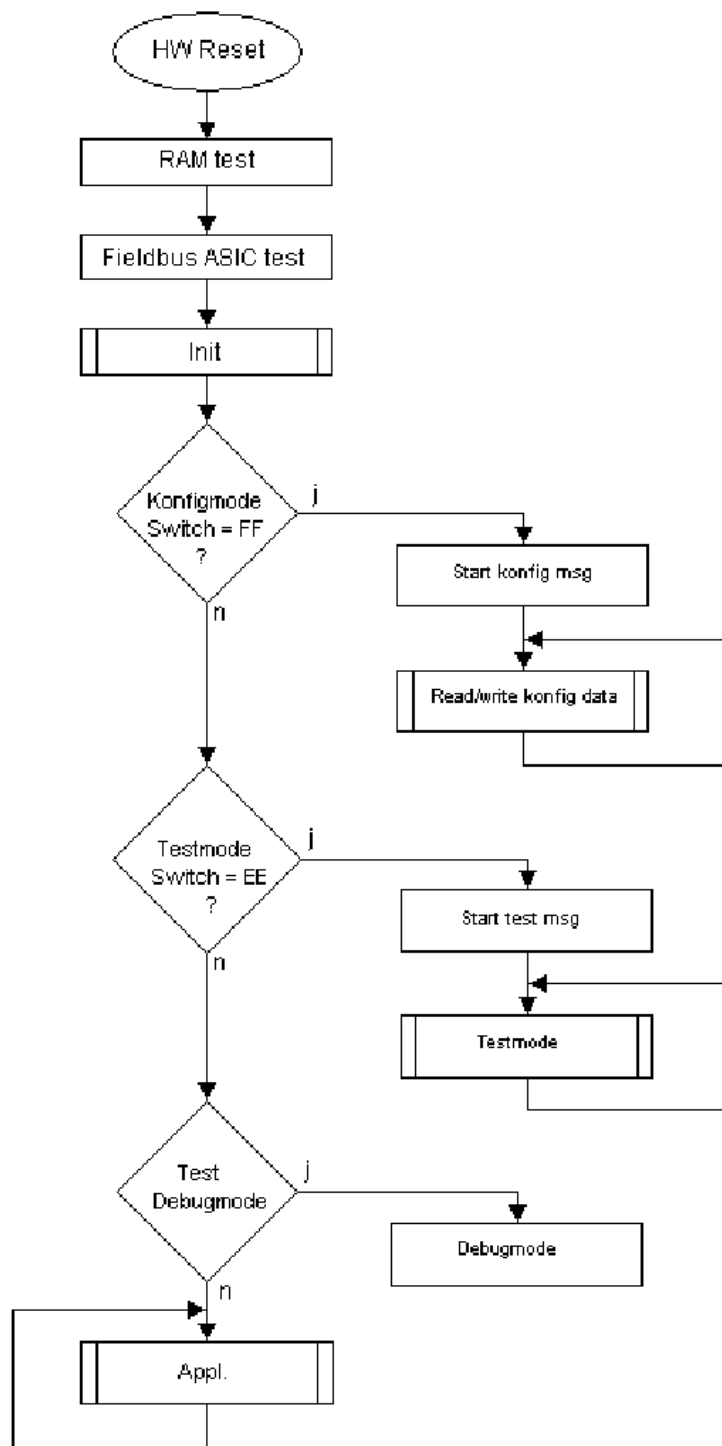
3 Einführung

Die Baugruppe UNIGATE CL-CANopen® dient als Anpassung einer seriellen Schnittstelle an CANopen®. Es fungiert in diesem Anwendungsfall als Gateway und arbeitet als CANopen® Slave. Es kann von jedem normkonformen Master betrieben werden.

Die Baugruppe CL-CANopen® besteht im wesentlichen aus folgenden Hardware-Komponenten:

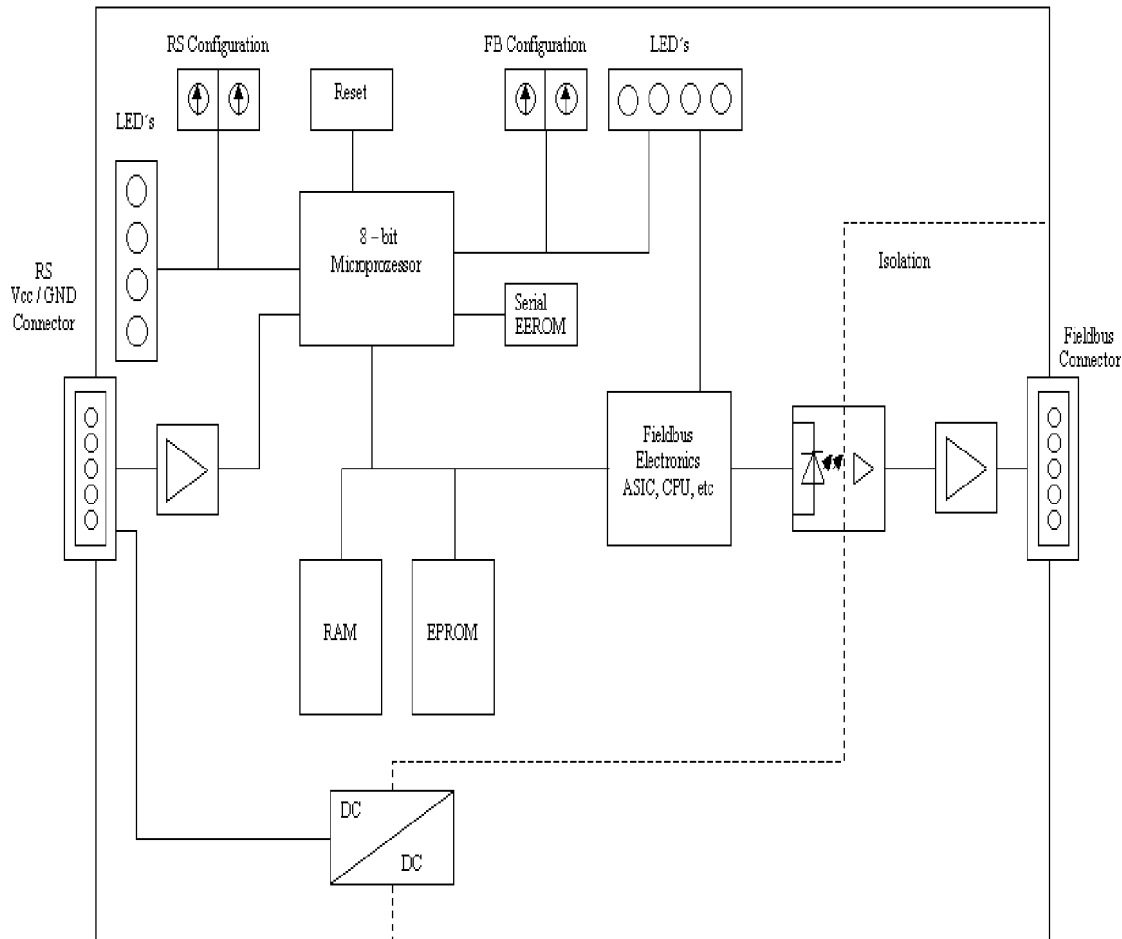
- Potentialgetrennte Schnittstelle zum CANopen®
- CAN-Controller SJA 1000
- Mikroprozessor 89C51RD2
- RAM und EPROM
- Optional potentialgetrennte RS-Schnittstelle
- Serielle Schnittstelle (RS232, RS485 und RS422) zum extern angeschlossenen Gerät

3.1 UNIGATE® CL Software Flussdiagramm



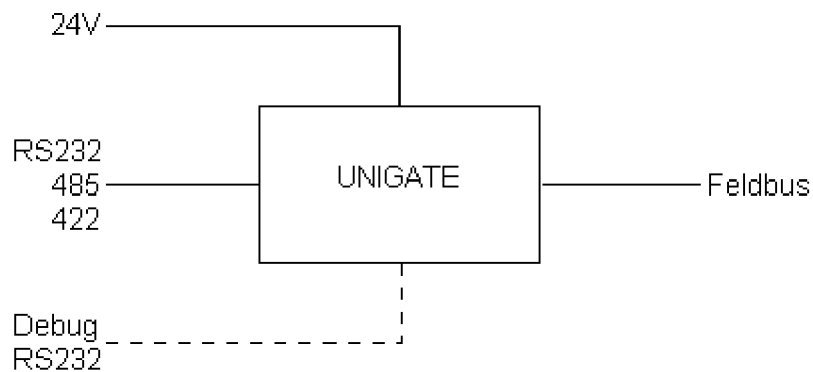
3.2 UNIGATE® Blockdiagramm

Das nachfolgende Bild zeigt ein typisches UNIGATE® Modul-Design.



3.3 UNIGATE® Applikationsdiagramm

Das nachfolgende Bild zeigt ein typisches Anschaltschema.



4 Die Betriebsmodi des Gateway

4.1 Konfigurationsmodus (config mode)

Der Konfigurationsmodus dient der Konfiguration des Gateways. In diesem Modus sind folgende Einstellungen möglich:

- Script einspielen
- Firmware updaten
- Konfiguration des Gateways

Das Gateway wird in diesem Modus starten wenn die Schalter S4 und S5 beim Start des Gateways beide die Stellung "F" haben. Das Gateway sendet unmittelbar nach dem Einschalten im Konfigurationsmodus seine Einschaltmeldung, die analog zu folgender Meldung aussieht:

"RS-COV4-CL (232/422/485) V2.0 [29] (c)dA Switch=0xC1FF Script(8k)="Leer"
Author="Deutschmann Automation GmbH" Version="1.0" Date=21.08.2001 SN=47110001".

Im Konfigurationsmodus arbeitet das Gateway immer mit den Einstellungen 9600 Baud, keine Parity, 8 Datenbits und 1 Stopbit, die RS-State LED wird immer rot blinken, die "Error No/Select ID" LEDs sind für den Benutzer ohne Bedeutung. Der Konfigurationsmodus ist in allen Software Revisionen enthalten.

4.2 Testmodus (test mode)

Einstellung des Testmodes

Der Testmode wird eingestellt, indem die Schalter S4 und S5 beide in die Stellung "E" gebracht werden. Alle anderen Schalter werden für die Einstellung des Testmodus nicht berücksichtigt. Mit diesen Einstellungen muss das Gateway neu gestartet werden (durch kurzzeitiges Trennen von der Spannungsversorgung).

Im Testmodus arbeitet das Gateway immer mit den Einstellungen 9600 Baud, keine Parity, 8 Datenbits und 1 Stopbit.

Er kann hilfreich sein, um das Gateway in die jeweilige Umgebung zu integrieren, z. B. um die Parameter der RS-Schnittstellen zu testen.

Funktionsweise des Testmodus

Nach dem Neustart im Testmodus wird das Gateway auf der seriellen Seite im Rhythmus von 1 Sekunde die Werte 0-15 in hexadezimaler Schreibweise ("0".."F") in ASCII-Kodierung senden. Gleichzeitig werden auf der Feldbus-Schnittstelle die gleichen Werte binär ausgegeben.

Die State-LED auf der RS-Seite wird in diesem Modus rot blinken, die "Error No/Select ID" LEDs werden den Wert, der z. Zt. ausgegeben wird, binär darstellen. Zusätzlich wird jedes Zeichen, das auf einer der Schnittstellen empfangen wird auf derselben Schnittstelle, als ein lokales Echo wieder ausgegeben. Auf der Feldbusseite wird nur das erste Byte für das lokale Echo benutzt, d. h. sowohl beim Empfang als auch beim Senden wird nur auf das erste Byte der Busdaten geschaut, die anderen Busdaten verändern sich gegenüber den letzten Daten nicht.

4.3 Datenaustauschmodus (data exchange mode)

Das Gateway muss sich im Datenaustauschmodus befinden, damit ein Datenaustausch zwischen der RS-Seite des Gateways und dem Feldbus möglich ist. Dieser Modus ist immer dann aktiv, wenn das Gateway sich nicht im Konfigurations- oder Test- bzw. Debugmodus befindet. Im Datenaustauschmodus wird das Gateway das eingespielte Script ausführen.

5 RS-Schnittstelle

5.1 RS-Schnittstellen beim UNIGATE® CL

Das UNIGATE CL - CANopen® verfügt über die Schnittstellen RS232, RS422 und RS485. Die Hardware hat auch immer eine DEBUG-Schnittstelle (siehe Kapitel 7).

5.2 Puffergrößen beim UNIGATE® CL

Dem UNIGATE® CL steht auf der seriellen Seite ein Puffer von jeweils 1024 Byte für Eingangs- und Ausgangsdaten zur Verfügung.

Der FIFO der Applikationsschnittstelle (RS-Schnittstelle) kann in jedem script-fähigen UNIGATE® ab Scriptrevision 26 geändert werden. Siehe dazu im Protocol Developer unter "Device Control" - "Hardware".

5.3 Framing Check

Über die Funktion "Framing Check" wird die Länge des Stopbits, das das Gateway empfängt überprüft. Hierbei ist das vom Gateway erzeugte Stopbit immer lang genug, damit angeschlossene Teilnehmer das Stopbit auswerten können.

Zu beachten ist, dass die Funktion "Framing Check" nur bei 8 Datenbits und der Einstellung "No parity" wirksam wird.

Weist das Stopbit bei aktivierter Prüfung die Länge 1 Bit nicht auf, wird ein Fehler erkannt und durch die Error LEDs angezeigt.

Die mögliche Einstellung für diesen Parameter kann vom Script gesteuert werden (siehe dazu die Online-Hilfe vom Protocol Developer). Die Voreinstellung für den "Stop Bit Framing Check" ist "enabled".

6 SSI-Schnittstelle

Das UNIGATE® unterstützt auch den Anschluss von Applikationen bzw. Produkten, die über SSI kommunizieren.

6.1 Inbetriebnahme der SSI-Schnittstelle

Die Konfiguration der SSI-Schnittstelle erfolgt im Konfigurationsmodus mit der Software WING-ATE, Protokoll SSI. Über die Parameter „Resolution“ (1 bit ... 15 bit, 24 bit ... 25 bit), „SSI Encoder Type“ (Binary oder Graycode) und „Clock stretch“ muss der Gebertyp und die Abtastfrequenz definiert werden.

6.2 Parameter

6.2.1 Resolution

Der Bereich erstreckt sich von 1 Bit bis 25 Bit. Damit können Singel-Turn SSI-Geber und Multi-Turn SSI-Geber konfiguriert werden.

6.2.2 SSI Encoder Type

Dieser kann zwischen Binär und Gray-Code gewählt werden.

6.2.3 Parameter Abtastfrequenz (Clock stretch)

Man kann die Abtastfrequenz verändern. Dazu wird ein "Stretch-Wert" übergeben, der eine Wartezeit nach jeder Clockflanke einfügt.

Wird eine 0 übergeben, gibt es keine Wartezeit.

Somit ergeben sich nun folgende SSI-Abtastfrequenzen, die geringfügig schwanken können:

Wartezeit = 0	→ SSI-Clock ~ 333kHz (No Stretch)
Wartezeit = 1	→ SSI-Clock ~ 185kHz
Wartezeit = 2	→ SSI-Clock ~ 150kHz
Wartezeit = 3	→ SSI-Clock ~ 125kHz
Wartezeit = 4	→ SSI-Clock ~ 110kHz
Wartezeit = 5	→ SSI-Clock ~ 100kHz
Wartezeit = 6	→ SSI-Clock ~ 88kHz
Wartezeit = 7	→ SSI-Clock ~ 80kHz
Wartezeit = 8	→ SSI-Clock ~ 72kHz
Wartezeit = 9	→ SSI-Clock ~ 67kHz
Wartezeit = A	→ SSI-Clock ~ 62kHz
Wartezeit = B	→ SSI-Clock ~ 58kHz
Wartezeit = C	→ SSI-Clock ~ 54kHz
Wartezeit = D	→ SSI-Clock ~ 50kHz
Wartezeit = E	→ SSI-Clock ~ 48kHz
Wartezeit = F	→ SSI-Clock ~ 45kHz

Die Bitzeit, aus der diese Frequenzen abgeleitet wurden, errechnet sich folgendermaßen:

$t = 3\mu s + (2 * (0,6\mu s + (n * 0,6\mu s)))$, wobei n dem "Stretch-Wert" (1..F) entspricht.

Ohne Clock-Verlängerung (n=0) bleibt es bei $3\mu s \rightarrow 333kHz$!

Bei der max. Bitlänge von 32 Bit und dem langsamsten Clock ergibt sich somit ein Gesamteinlesezeit von $32 * 22\mu s \approx 700\mu s$.

6.2.4 Parameter Geberüberwachung (Check Encoder)

Mit dem Parameter „Check Encoder“ kann eine Geberüberwachung aktiviert werden, sofern der verwendete SSI-Geber diese Funktion unterstützt. Es wird überprüft, ob nach dem letzten gelesenen Geberbit die Datenleitung noch für mindestens ein Bit auf Low gezogen bleibt. Erkennt das UNIGATE® dieses Bit NICHT auf Low, wird ein Error 12 ausgegeben. Das erkennt z.B. einen Kabelbruch oder einen nicht angeschlossenen Geber. Es kann aber auch eine falsch konfigurierte Bitlänge, oder ein zu langsamer Auslesetakt sein.

6.3 Hardware- Beschaltung

Die Taktleitungen der SSI-Schnittstelle werden auf die Tx-Leitungen der RS422-Schnittstelle gelegt, die Datenleitungen auf die Rx-Leitungen am UNIGATE®.

X1 (3pol. + 4pol. Schraub-Steckverbinder):

Pin Nr.	Name	Funktion bei SSI
1	Rx 232	n. c.
2	Tx 232	n. c.
3	AP-GND	n. c.
4	Rx 422+	SSI DAT+
5	Rx 422-	SSI DAT-
6	Tx 422+	SSI CLK+
7	Tx 422-	SSI CLK-

7 Die Debug Schnittstelle

7.1 Übersicht über die Debug Schnittstelle

Das UNIGATE® bietet eine Debug Schnittstelle, die es ermöglicht, ein Script schrittweise auszuführen. Diese Schnittstelle ist im Normalfall nur für die Entwicklung eines Scripts nötig.

7.2 Start im Debugmode

Mit dem Anlegen von Spannung an das UNIGATE® (Power up) wird die Firmware nach Ausführung eines Selbsttests auf dieser Schnittstelle das binäre Zeichen 0 (0x00) ausgeben. Wenn das UNIGATE® innerhalb von 500 ms eine Quittung über diese Schnittstelle erhält, befindet es sich im Debugmode. Die Quittung ist das ASCII-Zeichen O (0x4F).

Mit dem Start im Debugmode wird die weitere Ausführung von Scriptbefehlen unterbunden.

7.3 Kommunikationsparameter für die Debugschnittstelle

Die Debugschnittstelle arbeitet immer mit 9600 Baud, no Parity, 8 Datenbit, 1 Stopbit. Eine Änderung dieser Parameter ist im Protocol Developer nicht möglich. Bitte achten Sie darauf, dass diese Einstellungen mit denen der PC-COM-Schnittstelle übereinstimmen und dort die Flusssteuerung (Protokoll) auf „keine“ steht.

7.4 Möglichkeiten mit der Debug Schnittstelle

Normalerweise wird an der Debug Schnittstelle der Protocol Developer angeschlossen. Mit ihm ist es möglich ein Script schrittweise auszuführen, Sprünge und Entscheidungen zu beobachten sowie Speicherbereiche anzusehen. Außerdem können Haltepunkte gesetzt werden, er besitzt also im Prinzip alle Eigenschaften die ein Software-Entwicklungswerkzeug typischerweise besitzt. Es ist jedoch auch möglich über diese Schnittstelle ein Script Update durchzuführen.

Ab der Scriptversion [27] kann man mit dem Scriptbefehl „SerialOutputToDebugInterface“ auch Daten ausgeben. Bitte beachten Sie dazu den Hinweis im Handbuch 'Protocol Developer'.

7.5 Befehle der Debug Schnittstelle

Die Befehle zum Benutzen der Debug Schnittstelle sind dem Handbuch Protocol Developer zu entnehmen.

8 Funktionsweise des Systems

8.1 Allgemeine Erläuterung

Nach dem ISO/OSI-Modell kann eine Kommunikation in sieben Schichten, Layer 1 bis Layer 7, aufgeteilt werden.

Die Gateways der DEUTSCHMANN AUTOMATION setzen die Layer 1 und 2 vom kundenspezifischen Bussystem (RS485 / RS232 / RS422) auf das entsprechende Feldbussystem um. Layer 3 bis 6 sind leer, der Layer 7 wird gemäß Kapitel 8.3 umgesetzt.

8.2 Schnittstellen

Das Gateway ist mit den Schnittstellen RS232, RS422 und RS485 ausgerüstet.

8.3 Datenaustausch CANopen® V3

Alle Daten werden vom Gateway in Abhängigkeit des eingespielten Scripts übertragen.

Für den Datenaustausch auf der CANopen®-Seite existieren im Gateway folgende drei Objekte:

- Defaulteinstellung, solange der Scriptbefehl CO-InitChannel nicht ausgeführt wird.
- Adr. 2000H (Typ DOMAIN): Vom Gateway empfangene Daten
- Adr. 2001H (Typ DOMAIN): Vom Gateway gesendete Daten
- Adr. 2002H (Typ BYTE): Länge der gesendeten Daten

Die Länge der Empfangs- und Sendepuffer (Obj. 2000 + 2001) wird über WINGATE® konfiguriert.

8.3.1 SDO-Zugriff

Die Daten können generell immer über SDOs (Obj. 2000 - 2002) ausgetauscht werden.

Ebenso besteht über SDO ein Zugriff auf alle Mandatory-Objekte (gemäß CiA® DS 301).

8.3.2 PDO-Zugriff

Abhängig von der konfigurierten Länge werden PDOs gemäß folgender Tabelle unterstützt, wobei die PDO-Länge dynamisch auf den richtigen Wert eingestellt wird:

Gateway-Empfangs-Daten	Gateway-Sende-Daten	Empf.-PDO1 (Adr = 512 + ID)	Sende PDO1 (Adr = 384 + ID)
max. 8 Byte	max. 8 Byte	Empf. Daten	Sende Daten
max. 8 Byte	>8 Byte	Empf. Daten	Länge Sendedaten
>8 Byte	max. 8 Byte	-	Sende Daten
>8 Byte	>8 Byte	-	Länge Sendedaten

8.4 Mögliche Datenlängen

In der folgenden Tabelle sind die maximal in CANopen® zu übertragenden Daten tabellarisch dargestellt:

Eingangsdaten	max. 255 Bytes	variabel: hier Maximalwert
Ausgangsdaten	max. 255 Bytes	variabel: hier Maximalwert
Emergency-Daten	1 Byte	siehe Kap. Fehlerbehandlung

9 Erstellung eines Scripts

9.1 Was ist ein Script

Ein Script ist eine Anreihung von Befehlen, die in exakt dieser Reihenfolge ausgeführt werden. Dadurch, dass auch Mechanismen gegeben sind, die den Programmfluss im Script kontrollieren, kann man auch komplexere Abläufe aus diesen einfachen Befehlen zusammenbauen.

Das Script ist speicherorientiert. Das bedeutet, dass alle Variablen sich immer auf einen Speicherbereich beziehen. Allerdings brauchen Sie sich beim Entwickeln eines Scripts nicht um die Verwaltung des Speichers zu kümmern; das übernimmt der Protocol Developer für Sie.

9.2 Speichereffizienz der Programme

Ein Scriptbefehl kann z. B. eine komplexe Checksumme wie eine CRC-16 Berechnung über Daten ausführen. Für die Codierung dieses Befehls sind als Speicherbedarf (für den Befehl selbst) lediglich 9 Byte nötig. Dies ist nur möglich, indem diese komplexen Befehle in einer Bibliothek enthalten sind.

Ein weiterer Vorteil dieser Bibliothek ist, dass die zu Grunde liegenden Funktionen bereits seit Jahren im praktischen Einsatz sind und deswegen als fehlerarm bezeichnet werden können. Da diese Befehle auch im für den Controller nativen Code vorhanden sind, ist auch das Laufzeitverhalten des Scripts an dieser Stelle günstig.

9.3 Was kann man mit einem Script Gerät machen

Unsere Script Geräte sind in der Lage, eine Menge von Befehlen abzuarbeiten. Ein Befehl ist dabei immer eine kleine fest umrissene Aufgabe. Alle Befehle lassen sich in Klassen oder Gruppen einsortieren. Eine Gruppe von Befehlen beschäftigt sich mit der Kommunikation im allgemeinen, die Befehle dieser Gruppe befähigen das Gateway, Daten sowohl auf der seriellen Seite als auch auf der Busseite zu senden und zu empfangen.

9.4 Unabhängigkeit von Bussen

Prinzipiell sind die Scripts nicht vom Bus abhängig, auf dem sie arbeiten sollen, d. h. ein Script, das auf einem PROFIBUS Gateway entwickelt wurde, wird ohne Änderung auch auf einem Interbus Gateway laufen, da sich diese Busse von der Arbeitsweise sehr stark ähneln. Um dieses Script auch auf einem Ethernet Gateway abzuarbeiten, muss man evtl. noch weitere Einstellungen im Script treffen, damit das Script sinnvoll ausgeführt werden kann.

Es gibt keine festen Regeln, welche Scripts wie richtig arbeiten. Beim Schreiben eines Scripts sollten Sie beachten, auf welcher Zielhardware Sie das Script ausführen wollen, um die nötigen Einstellungen für die jeweiligen Busse zu treffen.

9.5 Weitere Einstellungen am Gateway

Die meisten Geräte benötigen keine weiteren Einstellungen außer denen, die im Script selbst getroffen sind. Allerdings gibt es auch Ausnahmen hierzu. Diese Einstellungen werden mit der Software WINGATE getroffen. Wenn Sie bereits unsere Serie UNIGATE® kennen, wird Ihnen die Vorgehensweise hierbei bereits bekannt sein. Beispielhaft sei hier die Einstellung der IP-Adresse und der Net-Mask eines Ethernet-Gateways genannt. Diese Werte müssen fix bekannt sein und sind auch zur Laufzeit nicht vorhanden. Ein weiterer Grund für die Konfiguration dieser Werte in WINGATE ist folgender: Nach einem Update des Scripts bleiben diese Werte unangetastet, d. h. die einmal getroffenen Einstellungen sind auch nach einer Änderung des Scripts weiterhin vorhanden.

Nur so ist es auch möglich, dass das gleiche Script auf verschiedenen Ethernet-Gateways

arbeitet, die alle eine unterschiedliche IP-Adresse haben.

9.6 Die Benutzung des Protocol Developer

Das Softwaretool Protocol Developer kann von unserer Internetseite <http://www.deutschmann.de> heruntergeladen werden.

Es ist als Werkzeug zum einfachen Erstellen eines Scripts für unsere Script Gateways gedacht; seine Bedienung ist genau darauf ausgerichtet. Nach dem Start des Programms wird das zuletzt geladene Script erneut geladen, sofern es nicht der erste Start ist.

Windows typisch können Script Befehle per Maus oder Tastatur hinzugefügt werden. Soweit für den entsprechenden Befehl definiert und notwendig wird der Dialog zu dem entsprechenden Befehl angezeigt, und nach dem Eingeben der Werte wird automatisch der richtige Text in das Script eingefügt. Das Einfügen von neuen Befehlen durch den Protocol Developer erfolgt so, dass niemals ein existierender Befehl überschrieben wird. Generell wird ein neuer Befehl vor dem eingefügt, auf dem momentan der Cursor positioniert ist. Selbstverständlich können die Befehle auch einfach per Tastatur geschrieben werden, oder bereits geschriebene Befehle bearbeitet werden.

9.7 Genauigkeiten der Baudraten

Die Baudrate der seriellen Schnittstelle wird aus der Quarzfrequenz des Prozessors abgeleitet. Zwischenzeitlich arbeiten alle Script-Gateways außer dem MPI-Gateway (20 MHz) mit einer Quarzfrequenz von 40 MHz.

Im Script lässt sich jede beliebige ganzzahlige Baudrate eingeben. Die Firmware stellt daraufhin die Baudrate ein, die am genauesten aus der Quarzfrequenz abgeleitet werden kann.

Die Baudrate, mit der das Gateway tatsächlich arbeitet (BaudIst) kann folgendermaßen ermittelt werden:

$$\text{BaudIst} = (F32 / K)$$

$$F32 = \text{Quarzfrequenz [Hz]} / 32$$

$$K = \text{Round}(F32 / \text{BaudSoll});$$

bei Round () handelt es sich um eine kaufmännische Rundung

Beispiel:

Es soll die genaue Ist-Baudrate berechnet werden, wenn 9600 Baud vorgegeben werden, wobei das Gateway mit 40 MHz betrieben wird:

$$F32 = 40000000 / 32 = 1250000$$

$$K = \text{Round}(1250000 / 9600) = \text{Round}(130.208) = 130$$

$$\text{BaudIst} = 1250000 / 130 = 9615.38$$

D. h.: Die Baudrate, die das Gateway tatsächlich einstellt beträgt 9615.38 Baud

Der entstandene Fehler in Prozent lässt sich folgendermaßen berechnen:

$$\text{Fehler}[\%] = (\text{abs}(\text{BaudIst} - \text{BaudSoll}) / \text{BaudSoll}) * 100$$

In unserem Beispiel ergibt sich somit ein Fehler von:

$$\text{Fehler} = (\text{abs}(9615.38 - 9600) / 9600) * 100 = 0.16\%$$

Fehler, die unter 2% liegen können in der Praxis toleriert werden!

Nachfolgend finden Sie eine Auflistung von Baudraten bei 40 MHz-Quarzfrequenz mit den dazugehörigen Fehlern:

4800 Baud:0,16%
9600 Baud:0,16%
19200 Baud:0,16%
38400 Baud:1,35%
57600 Baud:1,35%
62500 Baud:0%
115200 Baud:1,35%
312500 Baud:0%
625000 Baud: 0%

9.8 Scriptabarbeitungszeiten

Das Script wird vom Protocol Developer übersetzt, und der dabei erzeugte Code in das Gateway geladen. Der Prozessor im Gateway interpretiert nun diesen Code. Dabei gibt es Befehle, die sehr schnell abgearbeitet werden können (z. B. "Set Parameter"). Es gibt aber auch Befehle, die länger dauern (z. B. das Kopieren von 1000 Bytes). Somit differiert die Abarbeitungszeit zunächst einmal durch die Art des Scriptbefehls. Wesentlich stärker wird die Abarbeitungszeit der Scriptbefehle aber bestimmt durch die Prozessorzeit, die für diesen Prozess zur Verfügung steht. Da der Prozessor mehrere Aufgaben gleichzeitig ausführen muss (Multitasking-System), steht nur ein Teil der Prozessorleistung für die Scriptabarbeitung zur Verfügung. Folgende Tasks - in der Reihenfolge der Priorität - werden auf dem Prozessor ausgeführt:

- Senden und Empfangen von Daten an der Debug-Schnittstelle (nur wenn Protocol Developer auf PC gestartet ist)
- Senden und Empfangen von Daten an der RS-Schnittstelle
- Senden und Empfangen von Daten an der Feldbus-Schnittstelle
- Durch Systemtakt (1ms) gesteuerte Aufgaben (z. B. Blinken einer LED)
- Abarbeitung des Scriptes

Aus der Praxis heraus kann man ganz grob mit 0,5 ms pro Scriptzeile rechnen. Dieser Wert hat sich über viele Projekte hinweg immer wieder als Richtwert bestätigt. Er stimmt immer dann recht gut, wenn der Prozessor noch genügend Zeit für die Scriptabarbeitung zur Verfügung hat. An Hand der oben aufgelisteten Tasks kann man folgende Empfehlungen formulieren, um eine möglichst schnelle Scriptabarbeitung zu bekommen:

- Debugschnittstelle deaktivieren (ist im Serieneinsatz der Normalfall)
- Datenlast auf der RS-Schnittstelle möglichst klein halten. Dabei ist nicht die Baudrate v das Problem, sondern die Anzahl Zeichen, die pro Sekunde übertragen werden.
- Datenlast auch auf der Feldbusseite nicht unnötig gross machen. Insbesondere bei azyclischen Busdaten, diese möglichst nur bei Änderung schicken. Die Datenlänge bei Bussen die auf eine feste Länge konfiguriert werden (z.B. Profibus) nur so gross wählen, wie unbedingt notwendig.

Sollte trotz dieser Massnahmen die Abarbeitungszeit zu gross sein, besteht die Möglichkeit, einen kundenspezifischen Scriptbefehl zu generieren, der dann mehrere Aufgaben in einem Scriptbefehl abarbeitet. Wenden Sie sich dazu bitte an unsere Support-Abteilung.

10 Implementierte Protokolle mit Universalscript - Konfiguration

Das UNIGATE® CL wird mit dem Script "Universalscript Deutschmann" ausgeliefert. Die Konfiguration der Protokolle erfolgt mit der Software WINGATE. Siehe dazu die „Anleitung UNIGATE® CL - Konfiguration mit WINGATE“. Sie finden das PDF auch auf unserer Homepage unter Support/Downloads/Handbücher.



Achtung: Wird ein Reset Device durchgeführt, geht möglicherweise (je nach Firmware-Version des UNIGATE®) das "Universalscript" verloren und muss neu eingespielt werden.

Sollte Ihnen das kompilierte Script nicht mehr vorliegen, muss eine entsprechende Anfrage an den Deutschmann Support gestellt werden.

<https://www.deutschmann.de/de/support/anfrage/>

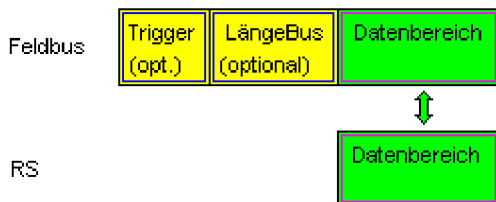
11 Implementierte Protokolle mit Universalscript - Protokolle Applikation

Nachfolgend finden Sie die Beschreibungen zu den implementierten Protokollen die für die Applikationsseite verwendet werden können.

11.1 Protokoll:Transparent

Die Daten werden bidirektional vom UNIGATE® übertragen.

11.1.0.1 Datenaufbau



Auf der RS-Eingangsseite ist eine Timeoutzeit von 2 ms fest eingestellt. Werden innerhalb der Timeoutzeit keine weiteren Daten empfangen, werden die bis dahin empfangenen Daten auf den Bus übertragen.

Werden weniger Daten über Rx empfangen, als über die GSD-Datei konfiguriert (I/O-Länge), dann wird der Rest mit NULL aufgefüllt.

Zuviel empfangene Daten werden abgeschnitten.

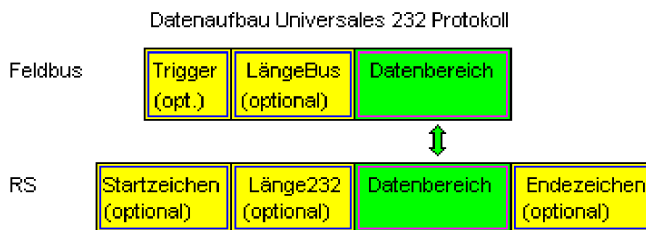
Die benötigte Länge der Eingangs- und Ausgangs-Daten (I/O-Länge) können, je nach Feldbus, über die Gerätekonfiguration des UNIGATES oder über die Gerätebeschreibungsdatei in der übergeordneten Steuerung eingestellt werden.

11.2 Protokoll: Universal 232



Die Protokollbezeichnung "Universal 232" und auch der Bezug auf die "RS232-Schnittstelle" in der Beschreibung sind historisch gewachsen. Das Protokoll funktioniert aber auch genauso mit RS422 und RS485!

11.2.1 Datenaufbau



11.2.2 Parameter Feldbus

Triggerbyte: Siehe „Das Triggerbyte“ Kapitel 14.

Längenbyte: Siehe „Das Längenbyte“ Kapitel 15.

11.2.3 Parametertabelle RS232

11.2.3.1 Startzeichen (232 Start character)

Ist dieses Zeichen definiert, wertet das Gateway nur die Daten an der RS232-Schnittstelle aus, die nach diesem Startzeichen folgen. Jede Sendung vom Gateway über die RS232-Schnittstelle wird in diesem Fall mit dem Startzeichen eingeleitet.

11.2.3.2 Länge232 (232 Length)

Ist dieses Byte aktiviert, erwartet das Gateway empfangsseitig sovielen Bytes Nutzdaten, wie in diesem Byte von dem RS232-Sendegerät angegeben werden. Sendeseitig setzt das Gateway dieses Byte dann auf die Anzahl der von ihm übertragenen Nutzdaten. Ist das Byte „Länge232“ nicht definiert, wartet das Gateway beim Empfang auf der RS232-Schnittstelle auf das Endekriterium, wenn dieses definiert ist. Ist auch kein Endekriterium definiert, werden sovielen Zeichen über die RS232-Schnittstelle eingelesen, wie im Feldbus-Sendepuffer übertragen werden können. Als Sonderfall kann für diesen Parameter auch ein Längenbyte mit zusätzlicher Timeoutüberwachung in WINGATE eingestellt werden. In diesem Fall werden die empfangenen Zeichen bei einem Timeout verworfen.



Achtung:

Ist als Endezeichen „Timeout“ gewählt, ist dieses Byte ohne Bedeutung.

11.2.3.3 Timeout

Wird das Endezeichen auf "FF" gesetzt, wird der Wert, der im Parameter RX_Timeout eingestellt wurde, aktiviert und die dort eingetragene Zeit beim seriellen Empfang abgewartet, bzw. getriggert bei neu einkommenden Zeichen. Ist die eingestellte Zeit ohne Ereignis überschritten, ist das Endekriterium erreicht und die Zeichen werden auf den Bus kopiert.

11.2.3.4 Datenbereich

In diesem Feld werden die Nutzdaten übertragen.

11.2.3.5 Endezeichen (232 End character)

Wenn dieses Zeichen definiert ist, empfängt das Gateway Daten von der RS232- Schnittstelle bis zu diesem Zeichen. Als Sonderfall kann hier das Kriterium „Timeout“ definiert werden. Dann empfängt das Gateway solange Zeichen, bis eine definierte Pause auftritt. Im Sonderfall „Timeout“ ist das „Länge 232-Byte“ ohne Bedeutung. Sendeseitig fügt das Gateway als letztes Zeichen einer Sendung das Endezeichen an, wenn es definiert ist.

11.2.4 Kommunikationsablauf

Die Nutzdaten (Datenbereich) die über den Feldbus ankommen, werden gemäß Kapitel 11.2.1 transparent in das RS232-Datenfeld kopiert, und über die RS-Schnittstelle übertragen, wobei das Protokoll gem. der Konfiguration (Startzeichen, Endezeichen...) ergänzt wird. Eine Quittung erfolgt NICHT!

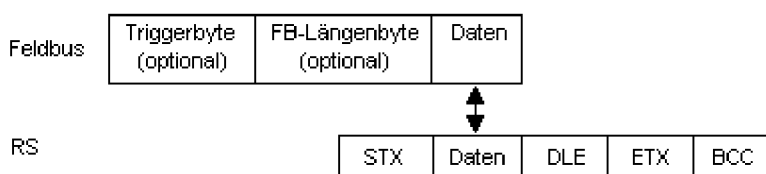
Ist das „Triggerbyte“ (siehe Kapitel 14) aktiv, werden Daten nur bei einem Wechsel dieses Bytes gesendet. Ist das „LaengenByte“ (siehe Kapitel 15) aktiv, werden nur so viele der nachfolgenden Bytes, wie dort spezifiziert sind, übertragen.

Empfangsdaten an der RS-Schnittstelle werden gem. dem konfigurierten Protokoll ausgewertet, und das Datenfeld (Datenbereich (siehe Kapitel 11.2.1)) an den Feldbusmaster gesendet. Sind mehr Zeichen empfangen worden, als Feldbusblocklänge, werden die hinteren Bytes abgeschnitten und ein Rx-Overrun angezeigt, sind weniger empfangen worden, wird mit 0 aufgefüllt. Ist das „LaengenByte“ aktiv, wird dort die Anzahl der empf. Nutzdaten eingetragen. Ist das „Triggerbyte“ aktiv, wird dieses nach jedem vollständigem Empfang an der RS-Schnittstelle um eins erhöht.

11.3 Protokoll: 3964(R)

Mit dem 3964-Protokoll werden Daten zwischen 2 seriellen Geräten übertragen. Zum Auflösen von Initialisierungskonflikten muss ein Partner hochprior und der andere niederprior sein.

11.3.1 Datenaufbau 3964R



11.3.2 Protokollfestlegungen

Das Telegrammformat ist:

STX	Daten	DLE	ETX	BCC
-----	-------	-----	-----	-----

- Die empfangenen Nettodaten werden in beiden Richtungen unverändert durchgereicht (transparent).
- **Achtung:** Davon ausgenommen ist die DLE-Verdoppelung; d. h. ein DLE (10H) auf der Busseite wird zweifach auf der RS-Seite gesendet, ein doppeltes DLE auf der RS-Seite wird nur einmal an den Busmaster gesendet.
- Eine Datenblockung ist nicht vorgesehen.
- Die Nettodatenlänge ist auf 236 Bytes pro Telegramm beschränkt.

- Die Kommunikation läuft immer zwischen hoch- und niederprioren Kommunikationspartner ab.

11.3.3 Datenverkehr

11.3.3.1 Einleitung des Datenverkehrs durch den niederprioren Teilnehmer

Empfängt der niederpriore Teilnehmer auf ein ausgesendetes STX ebenfalls ein STX, dann unterbricht er seinen Sendewunsch, geht in den Empfangsmodus über und quittiert das empfangene STX mit DLE.

Ein DLE im Datenstring wird verdoppelt und in die Prüfsumme mit einbezogen. Der BCC errechnet sich aus der XOR Verknüpfung aller Zeichen.

11.3.4 Konfliktfälle

11.3.4.1 Überwachungszeiten

Die Überwachungszeiten sind durch die Definition des 3964R-Protokolls vorgegeben und können nicht überschrieben werden !!!

tq = Quittungsüberwachungszeit (2 s).

Die Quittungsüberwachungszeit wird nach Senden des Steuerzeichens STX gestartet. Trifft innerhalb der Quittungsüberwachungszeit keine positive Quittung ein, wird der Auftrag wiederholt (max. 2 x). Konnte der Auftrag nach 2 maligem Wiederholen nicht positiv abgeschlossen werden, versucht das hochpriore Gerät trotzdem Kontakt mit dem niederprioren Partner aufzunehmen durch Senden von STX (Zyklus entspricht tq).

tz = Zeichenüberwachungszeit (200 ms)

Empfängt der 3964 R Treiber Daten, überwacht er das Eintreffen der einzelnen Zeichen innerhalb der Zeit tz. Wird innerhalb der Überwachungszeit kein Zeichen empfangen, beendet das Protokoll die Übertragung. Zum Kopplungspartner wird keine Quittung gesendet.

11.3.4.2 Wiederholungen

Bei negativer Quittung oder Zeitüberschreitung wird ein vom hochprioren Teilnehmer gesendetes Telegramm 2 x wiederholt. Danach meldet das Gateway die Kommunikation als gestört, versucht aber weiterhin, die Verbindung wieder aufzubauen.

11.3.4.3 Einleitung des Datenverkehrs durch den hochprioren Teilnehmer

Bei negativer Quittung oder Zeitüberschreitung wird ein vom externen Gerät gesendetes Telegramm 2 x wiederholt, bevor eine Störung gemeldet wird.

11.3.5 Protokolltyp 3964

Der Unterschied zum Protokolltyp 3964R ist:

1. tq = Quittungsüberwachungszeit (550 ms)
2. Es fehlt das Checksummenbyte BCC.

11.4 Protokoll: Modbus-RTU

11.4.1 Hinweise

- Im folgenden Text wird für „Modbus-RTU“ der Einfachheit halber immer „Modbus“ geschrieben.
- Die Begriffe „Input“ und „Output“ sind immer aus der Sicht des Gateways gesehen; d. h. Feldbus-Input-Daten sind die Daten, die vom Feldbus-Master an das Gateway geschickt werden.

11.4.2 UNIGATE® als Modbus-Master

11.4.2.1 Vorbereitung

Vor dem Beginn des Datenaustausches müssen die Parameter „Baudrate“, „Parity“, „Start-“, „Stop-“ und „Datenbits“ sowie gegebenenfalls „Triggerbyte“ und „Längenbyte“ eingestellt werden.

Außerdem muss eine „Responsetime“ vorgegeben werden, die der max. Zeit entspricht, bis der Modbus-Slave nach einer Anfrage antwortet. Der in WINGATE eingetragene Wert wird vom UNIGATE® mit 10 ms multipliziert.

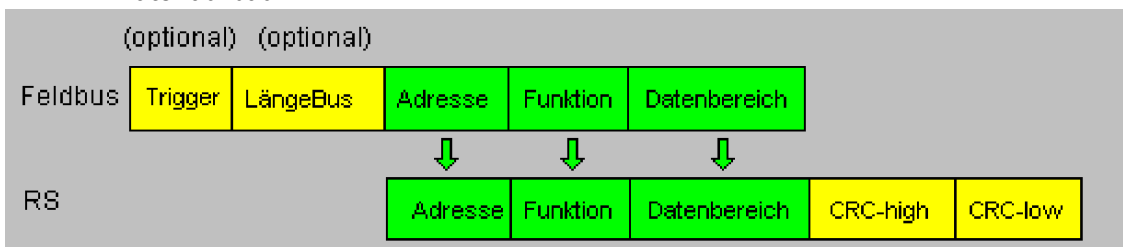
Da der Modbus mit einem variablen Datenformat arbeitet - abhängig von der gewünschten Funktion und Datenlänge - der Feldbus aber eine feste Datenlänge benötigt, muss diese über die Konfiguration mit WINGATE (Blocklength fieldbus input und Blocklength fieldbus output) vorgegeben werden. Diese Länge sollte vom Anwender so gewählt werden, dass die längste Modbus-Anfrage bzw. Antwort bearbeitet werden kann.

Der Anwender kann wählen, ob die Feldbusanfragen ereignisorientiert (On Event) an den Modbus weitergegeben werden oder auf Anforderung (On Trigger).

Der Modus „Modbusanfrage auf Anforderung“ bedingt, dass das erste Byte im Feldbus ein Triggerbyte enthält (siehe Kapitel 14). Dieses Byte wird nicht zum Modbus übertragen und dient nur dazu, eine Modbussendung zu starten. Dazu überwacht das Gateway ständig dieses Triggerbyte und sendet nur dann Daten an den Modbus, wenn sich dieses Byte geändert hat. In der umgekehrten Richtung (zum Feldbus) überträgt das Gateway in diesem Byte die Anzahl der empfangenen Modbusdatensätze; d. h. nach jedem Datensatz wird dieses Byte vom Gateway inkrementiert.

Ist das „Längenbyte“ aktiviert (siehe Kapitel 15), werden vom Gateway nur die Anzahl Bytes, die dort spezifiziert sind, übertragen. Zum Feldbus-Master hin wird dort die Anzahl der empfangenen Modbusdaten hinterlegt. Die Länge bezieht sich dabei immer auf die Bytes „Adresse“ bis „Dat n“ (jeweils incl.) immer ohne CRC-Checksumme.

11.4.2.2 Datenaufbau



11.4.2.3 Kommunikationsablauf

Das Gateway verhält sich zum Feldbus immer als Slave und auf der Modbus-Seite immer als Master. Somit muss ein Datenaustausch immer vom Feldbusmaster gestartet werden. Das Gateway nimmt diese Daten vom Feldbusmaster, die gemäß Kapitel „Datenaufbau“ angeordnet sein müssen, ermittelt die gültige Länge der Modbusdaten, wenn das Längenbyte nicht aktiviert ist, ergänzt die CRC-Checksumme, und sendet diesen Datensatz als Anfrage auf dem Modbus.

Die Antwort des selektierten Slaves wird vom Gateway daraufhin - ohne CRC-Checksumme - an den Feldbusmaster geschickt. Erfolgt innerhalb der festgelegten „Responsetime“ keine Antwort, meldet das Gateway einen „TIMEOUT-ERROR“.

11.4.3 UNIGATE® als Modbus-Slave

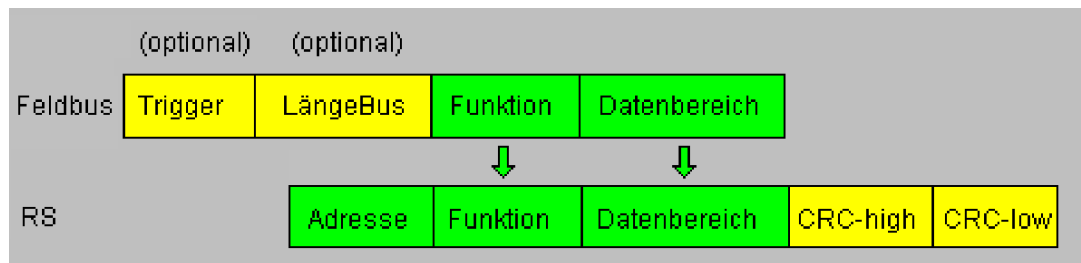
11.4.3.1 Vorbereitung

Vor dem Beginn des Datenaustausches müssen die Parameter „Trigger-“ und „Längenbyte“, „Baudrate“, „Parity“, „Start-“, „Stop-“ und „Datenbits“ eingestellt werden.

Am Drehschalter auf der RS-Seite muss der Modbus-ID eingestellt werden, unter dem das Gateway im Modbus angesprochen wird.

Da der Modbus mit einem variablen Datenformat arbeitet - abhängig von der gewünschten Funktion und Datenlänge - der Feldbus aber eine feste Datenlänge benötigt, muss diese über die Konfiguration mit WINGATE (Blocklength fieldbus input und Blocklength fieldbus output) vorgegeben werden. Diese Länge sollte vom Anwender so gewählt werden, dass die längste Modbus-Anfrage bzw. Antwort bearbeitet werden kann.

11.4.3.2 Datenaufbau



11.4.3.3 Kommunikationsablauf

Das Gateway verhält sich zum Feldbus immer als Slave und auf der Modbus-Seite ebenfalls als Slave. Ein Datenaustausch wird immer vom Modbus-Master über die RS-Schnittstelle eingeleitet. Ist die vom Modbus-Master ausgesandte Modbus-Adresse (1. Byte) identisch mit der am Gateway eingestellten Adresse, sendet das Gateway die empfangenen Daten (ohne Modbus-Adresse und CRC-Checksumme) an den Feldbusmaster (siehe Bild oben). Dabei ergänzt das Gateway als Vorspann optional ein Trigger- und ein Längenbyte.

Durch das Triggerbyte, das vom Gateway bei jeder Anfrage inkrementiert wird, erkennt der Feldbusmaster, wann er einen Datensatz auswerten muss.

Im Längenbyte befindet sich die Anzahl der nachfolgenden Modbusdaten.

Der Feldbusmaster muss nun die Modbusanfrage auswerten und die Antwort im gleichen Format (optional mit führendem Trigger- und Längenbyte) über den Feldbus an das Gateway zurücksenden.

Das Gateway nimmt dann diese Antwort, ergänzt Modbus-Adresse und CRC und schickt die Daten über die RS-Schnittstelle an den Modbus-Master.

Damit ist der Datenaustausch abgeschlossen und das Gateway wartet auf eine neue Anfrage des Modbus-Masters.

11.4.4 UNIGATE® als Modbus-ASCII Master

-> Beschreibung: siehe Kapitel 11.4.2, UNIGATE® als Modbus-Master.

11.5 Protokoll „Universal Modbus RTU Slave“

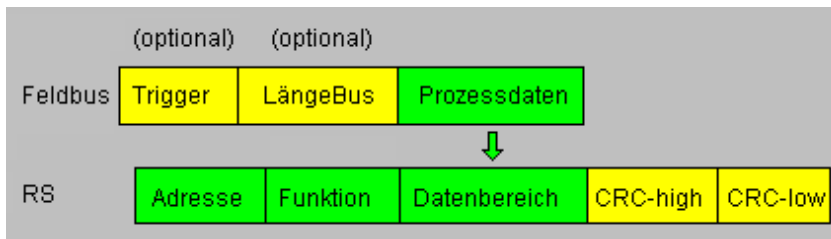
Das UNIGATE® ist auf der Applikationsseite Modbus-Slave. Die Slave ID wird an den Drehschaltern S4 + S5 eingestellt (S4 = High, S5 = Low)

11.5.1 Datenaufbau Feldbusseite z.B.: PROFIBUS

Gilt für In und Out

1. Byte: Trigger-Byte, optional (siehe Kapitel 14, Das Triggerbyte)
2. Byte: Feldbuslängen-Byte, optional (siehe Kapitel 15, Das Längenbyte)
3. Byte: Prozessdaten
4. Byte: Prozessdaten
-

Datenaufbau



11.5.1.1 Beispiel: FC1 + FC2

Ein Modbus Master (externes Gerät) sendet einen Request (Anfrage) mit Funktionscode 1 oder 2.

Hinweis:

Modbus Master Request Adresse (High + Low)
 Adressabfrage 01 .. 08 wird immer auf Adresse 01 sein.
 Adressabfrage 09 .. 16 wird immer auf Adresse 09 sein.
 Adressabfrage 17 .. 24 wird immer auf Adresse 17 sein.
 ...

Konfiguration:

-----FIELDBUS-----	
Fieldbus ID	126
Data exchange	On Change
Fieldbus lengthbyte	active
-----APPLICATION-----	
Protocol	Universal Modbus RTU Slave

Feldbus sendet zum UNIGATE®

08 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A...

Hinweis: Das 1. Byte (0x08) ist das Feldbuslängenbyte. Es werden also nur die folgenden 8 Byte im UNIGATE® gespeichert.

Angeschlossener Modbus Master sendet Request an die RS232/484 Seite des UNIGATE®:

Start-Adresse 0001, Length 56 (38h), FC1 (-Read Coil Status)

[01] [01] [00] [00] [00] [38] [3d] [d8]

UNIGATE® sendet über RS232/485 Response:

[01] [01] [07] [01] [02] [03] [04] [05] [06] [07] [6b] [c5]

Darstellung der Daten im Modbus Master (FC1):

```
00001: <1> 00009: <0> 00017: <1> 00025: <0> 00033: <1> 00041: <0> 00049: <1>
00002: <0> 00010: <1> 00018: <1> 00026: <0> 00034: <0> 00042: <1> 00050: <1>
00003: <0> 00011: <0> 00019: <0> 00027: <1> 00035: <1> 00043: <1> 00051: <1>
00004: <0> 00012: <0> 00020: <0> 00028: <0> 00036: <0> 00044: <0> 00052: <0>
00005: <0> 00013: <0> 00021: <0> 00029: <0> 00037: <0> 00045: <0> 00053: <0>
00006: <0> 00014: <0> 00022: <0> 00030: <0> 00038: <0> 00046: <0> 00054: <0>
00007: <0> 00015: <0> 00023: <0> 00031: <0> 00039: <0> 00047: <0> 00055: <0>
00008: <0> 00016: <0> 00024: <0> 00032: <0> 00040: <0> 00048: <0> 00056: <0>
```

Beispiel: StartAddress 0008, Length 80, FC2 (Read Input Status)

[01] [02] [00] [07] [00] [50] [c9] [f7]

UNIGATE® sendet über RS232/485 Response:

[01] [02] [0a] [02] [03] [04] [05] [06] [07] [08] [00] [00] [00] [8f] [7a]

11.5.1.2 Beispiel: FC3 (Read Holding Register) + FC4 (Read Input Register)

Feldbus sendet zum UNIGATE®

00 30 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 20 20 20...

(Konfiguration ist diesmal mit „Data exchange = On Trigger“, also mit zusätzlichem 1. Steuerbyte in den Feldbusdaten).

„Feldbus lengthbyte = active“, in diesem Beispiel 30h (48d), das UNIGATE® kopiert somit die folgenden 48 Byte vom Feldbus in den internen Speicher.

Angeschlossener Modbus Master sendet Request an die RS232/484 Seite des UNIGATE®

[01] [03] [00] [00] [00] [14] [45] [c5]

UNIGATE® sendet über RS232/485 Response:

[01] [03] [28] [02] [03] [04] [05] [06] [07] [08] [09] [0a] [0b] [0c] [0d] [0e] [0f] [10] [11] [12] [13] [14]...
... [15] [16] [17] [18] [19] [1a]

Darstellung der Prozessdaten im Modbus Master:

```
40001: <0203H>
40002: <0405H>
40003: <0607H>
40004: <0809H>
40005: <0A0BH>
40006: <0C0DH>
40007: <0E0FH>
40008: <1011H>
40009: <1213H>
40010: <1415H>
40011: <1617H>
40012: <1819H>
40013: <1A20H>
40014: <2020H>
40015: <2020H>
40016: <0000H>
40017: <0000H>
40018: <0000H>
40019: <0000H>
40020: <0000H>
```

Funktionsweise FC3 und FC4 im Protokoll "Universal Modbus (RTU/ASCII) Slave":

Ab „Universalscript Deutschmann“ V1.5.1:

- FC3 (0x03): Read Holding Registers greifen auf den Puffer Data To SPS zu.
- FC4 (0x04): Read Input Registers greift auf den Puffer Data From SPS zu.

11.5.1.3 Beispiel: Schreibe Single Coil FC5

Feldbus Master hat z.B. einmal folgende Daten an das UNIGATE® gesendet:

07 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 20 20 20...

1. Byte = Feldbuslängenbyte

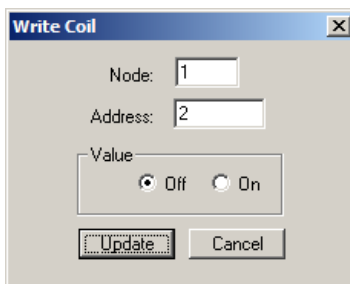
Es werden die folgenden 7 Byte im UNIGATE® gespeichert, der Rest wird nicht überschrieben.

Ein Modbus Master liest mit FC 1 und der Coil-Länge = 80 (10 Byte), folgende Werte aus:

```
00001: <1> 00017: <1> 00033: <1> 00049: <1> 00065: <0>
00002: <0> 00018: <1> 00034: <0> 00050: <1> 00066: <0>
00003: <0> 00019: <0> 00035: <1> 00051: <1> 00067: <0>
00004: <0> 00020: <0> 00036: <0> 00052: <0> 00068: <0>
00005: <0> 00021: <0> 00037: <0> 00053: <0> 00069: <0>
00006: <0> 00022: <0> 00038: <0> 00054: <0> 00070: <0>
00007: <0> 00023: <0> 00039: <0> 00055: <0> 00071: <0>
00008: <0> 00024: <0> 00040: <0> 00056: <0> 00072: <0>
00009: <0> 00025: <0> 00041: <0> 00057: <0> 00073: <0>
00010: <1> 00026: <0> 00042: <1> 00058: <0> 00074: <0>
00011: <0> 00027: <1> 00043: <1> 00059: <0> 00075: <0>
00012: <0> 00028: <0> 00044: <0> 00060: <0> 00076: <0>
00013: <0> 00029: <0> 00045: <0> 00061: <0> 00077: <0>
00014: <0> 00030: <0> 00046: <0> 00062: <0> 00078: <0>
00015: <0> 00031: <0> 00047: <0> 00063: <0> 00079: <0>
00016: <0> 00032: <0> 00048: <0> 00064: <0> 00080: <0>
```

Die Feldbusausgangsdaten werden erst aktualisiert wenn sie über einen Schreibbefehl von der RS Seite angetriggert werden.

Dies geschieht z.B. über den FC 5 :



Adresse 0002 bleibt unverändert auf 0, aber die Feldbusausgangsdaten werden aktualisiert:

Nach einem Reset sind sie erst einmal NULL (1. Zeile) und werden dann aktualisiert (2. Zeile):

00 ...

1F 01 02 03 04 05 06 07 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00...

Das 1. Byte ist das Feldbuslängenbyte. Es enthält die Anzahl Nutzzeichen, danach folgen die Nutzdaten. Der Nutzdatenbereich (interner Buffer) ist maximal 1024 Byte groß.

Im folgenden Beispiel wird das Bit (Coil) in Adresse 0002 auf High (1) gesetzt:

Write Coil

Node: 1

Address: 2

Value

☐ Off ☒ On

Update Cancel

Die Feldbusdaten werden aktualisiert:

1F 03 02 03 04 05 06 07 00 00 00 00 00

Der interne Buffer behält sich auch diesen Wert, daher kann er vom Master per FC1 Read Coil Status zurück gelesen werden:

```
00001: <1>
00002: <1>
00003: <0>
00004: <0>
00005: <0>
00006: <0>
00007: <0>
```

11.5.1.4 Beispiel: Write Single Register FC6

Modbus Master sendet in Adresse 0008 den Wert 1234H.

Write Register

Node: 1

Address: 8

Value, (HEX): 1234

Update Cancel

Der Modbus Master sendet den Request an das UNIGATE®:

[01] [06] [00] [07] [12] [34] [35] [7c]

Das UNIGATE® sendet einen Response:

[01] [06] [00] [07] [12] [34] [35] [7c]

Die 1. Zeile zeigt die Feldbusdaten VOR dem Schreibbefehl.

```
1F 03 02 03 04 05 06 07 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00...
```

```
1F 03 02 03 04 05 06 07 00 00 00 00 00 00 00 12 34 00 00 00 00 00 00 00 00 00 00 00 00...
```

Die 2. Zeile zeigt die Feldbusdaten NACH dem Schreibbefehl.

1F 03 02

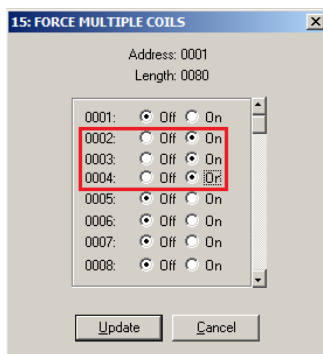
| +---- 1. Prozesswert

11.5.1.5 Beispiel: Force multiple coils FC 15

Hinweis: Für die Adresse kann nur ein Vielfaches von 8 übergeben werden incl. Null.

Beispiel: Startadresse = 0001.

Geändert wurde Adr 0002 ... 0004 von Low auf High:



Zeile eins zeigt die Feldbusdaten VOR dem Request:

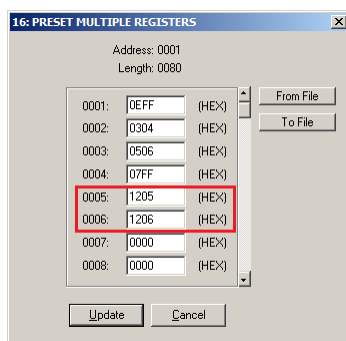
```
1F 00 FF 03 04 05 06 07 FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00...
```

```
1F 0E FF 03 04 05 06 07 FF 12 05 12 06 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ...
```

Zeile zwei NACH dem Request.

Geändert hat sich somit der 1. Prozessdatenwert von 00h nach 0Eh.

11.5.1.6 Beispiel: Preset multiple register FC16



Geändert wurde nur der Inhalt von Registeradresse 0005 und 0006.

Die 1. Zeile zeigt den Feldbusdateninhalt VOR dem Update.

1F 0E FF 03 04 05 06 07 FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00...

1F 0E FF 03 04 05 06 07 FF 12 05 12 06 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00...

Die 2. Zeile zeigt den Feldbusdateninhalt NACH dem Update.

11.6 Protokoll „Universal Modbus RTU Master“

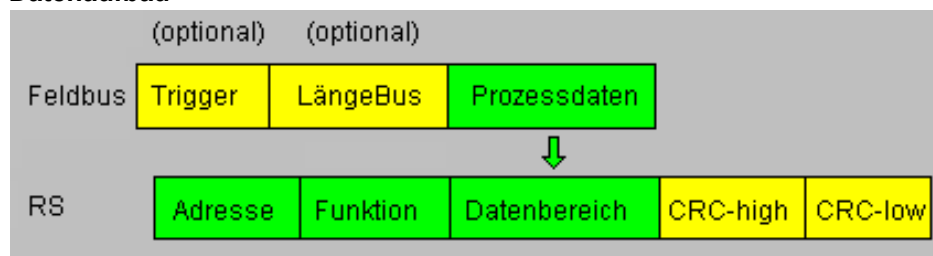
Das UNIGATE® ist auf der Applikationsseite Modbus-Master

11.6.1 Datenaufbau Feldbusseite (z.B. PROFIBUS):

Gilt für In und Out

1. Byte: Trigger-Byte , optional (siehe Kapitel 14, Das Triggerbyte)
2. Byte: Feldbuslängen-Byte, optional (siehe Kapitel 15, Das Längenbyte)
3. Prozessdaten

Datenaufbau



11.6.2 Datenaufbau Applikationsseite:

Nach Modbus RTU Master Definition.

Unterstützte Funktionen:

Read coil status FC1	(No. of Points = Bit)
Read input status FC2	(No. of Points = Bit)
Read multiple register FC3	(No. of Points = Word)
Read input registers FC4	(No. of Points = Word)
Force single coil FC5	(No. of Points – not used = fix 1 Bit)
Preset single register FC6	(No. of Points – not used = fix 1 Word)
Force multiple coils FC15	(No. of Points = Bit)
Preset multiple register FC16	(No. of Points = Word)

Hinweis:

status and coil = 1 Bit, register = 16 Bit.

FC 1 + 2 sowie FC 3 + 4 sind im Prinzip gleich, der einzige Unterschied ist die Definition der Startadresse.

Bei FC1 fängt sie bei Null an, bei FC2 bei 10 000.

Bei FC3 fängt sie bei 40 000 an, bei FC4 bei 30 000

11.6.3 Konfiguration: über Wingate ab wcf Datei Version 396

Parameter Name	Wertebereich	Erklärung
Modbus Timeout (10ms)	1 ... 255 (10ms ... 2550ms)	Maximale Wartezeit auf den "Response" bevor ein Error 9 für Timeout generiert wird. Ist „RX Poll Retry“ > 0 wird erst nach den Wiederholversuchen ein Error generiert.
RX Poll Retry		Wiederholversuche des letzten, ungültig beantworteten, "Request"
RX Poll Delay (10ms)		Pause vor dem nächsten "Request"

Konfigurationsparameter für einen Modbus Request:

Req. 1 Slave ID: Slave ID des Modbuslaveteilnehmers

Req. 1 Modbus Function: s. "Unterstützte Funktionen"

Req. 1 StartAdr (hex): Startadresse (High / Low) der Modbusregister ab der gelesen/geschrieben werden soll.

Req. 1 No. of Points (dec): Anzahl der zu lesenden/schreibenden Register/Coils

Req. 1 Fieldbus Map Adr(Byte): Position des zu kopierenden Prozesswertes aus/zu dem Feldbusbereich, je nach Schreib/Lesebefehl. Ist der Wert NULL, werden die Prozessdaten automatisch hintereinander gereiht.

Es können bis zu 24 Request's konfiguriert werden.

Zusätzliche konfigurationsmöglichkeiten in der Einstellung „Req. ... Modbus Function“:

jump to Req. 1: springe zum 1. Requesteintrag

disable this Req.: überspringe diesen Request und führe den nächsten Request-Eintrag aus.

„(10ms)“: einstellbar in 10ms Schritten

„(hex)“: Eingabe in hexadezimaler Schreibweise.

„(dec)“: Eingabe in dezimaler Schreibweise.

„(Byte)“: Zählweise in Byte, angefangen bei Position Null. !!! Achtung: Bei Lesebefehle, z.B. FC3, ist nach dem Trigger- und Längenbyte der erste Prozesswert die Position Null, die auf den Feldbus zur SPS kopiert wird.

Bei Schreibbefehlen, z.B. FC16, ist die Position Null das Triggerbyte.

AD 07 00 00 00 00 00 00 01 00 00 00 00 00 00 00
 AE 07 00 00 00 00 00 00 03 00 00 00 00 00 00 00

Hier sieht man die Änderung:

9. Byte = Fieldbus Map Adr 6 (Wert = 0x01) => 0x03

Eine Änderung von Adresse 7 im Modbus Slave hat keine Auswirkung auf die Feldbusausgangsseite, weil in der Konfiguration „No. Of Points“ = 2 eingestellt ist.

```
00001: <0>
00002: <0>
00003: <0>
00004: <0>
00005: <1>
00006: <1>
00007: <1>
00008: <0>
```

Der Wert bleibt unverändert auf 0x03:
 1F 07 00 00 00 00 00 00 03 00 00 00 0

11.6.3.2 Beispiel: Read input status FC2

Im folgenden Beispiel der Inhalt von Adresse 10007 ... 10009 in das 8. Feldbusausgangsbyte gemapped (kopiert/übertragen).

Req. 1 Slave ID	1
Req. 1 Modbus Function	Read input status FC2
Req. 1 StartAdr (hex)	0006
Req. 1 No. of Points (dec)	3
Req. 1 Fieldbus Map Adr(Byte)	8

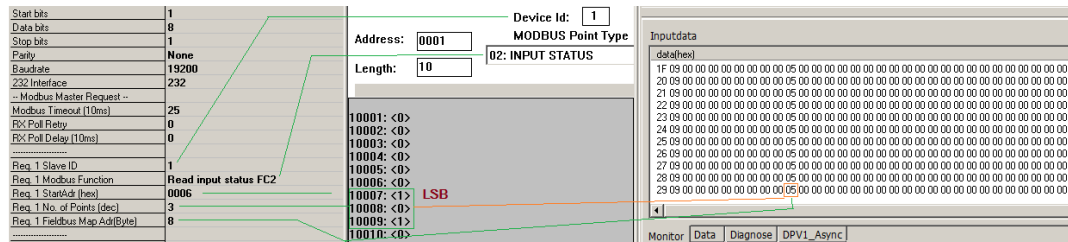
Address:
 Device Id:

Length:
 MODBUS Point Type:

```
10001: <0>
10002: <0>
10003: <0>
10004: <0>
10005: <0>
10006: <0>
10007: <1>
10008: <0>
10009: <0>
10010: <0>
```

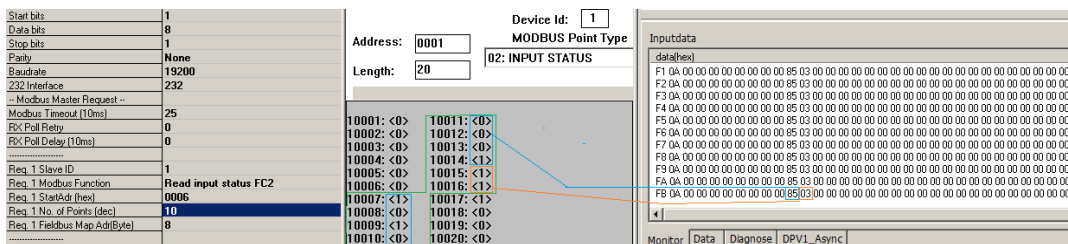
76 09 00 00 00 00 00 00 00 01 00 00 00 00

Im folgenden Beispiel ändert sich der Inhalt von Adresse 10009 von 0 -> 1



Im nächsten Beispiel wurde nur die „No. Of Points auf 10 geändert.

D.h. es werden nun 10 Bits => 2 Byte ausgelesen. Aus diesem Grund hat sich auch das Feldbus-längen Byte (2. Feldbusbyte) auf 0x0A, also um 1 Byte, erhöht.



11.6.3.3 Beispiel: Read multiple register FC3

Protocol	Universal Modbus RTU Master
-- Modbus Master Request --	
Modbus Timeout (10ms)	25
RX Poll Retry	0
RX Poll Delay (10ms)	0
Req. 1 Slave ID	1
Req. 1 Modbus Function	Read multiple register FC3
Req. 1 StartAdr (hex)	0001
Req. 1 No. of Points (dec)	2
Req. 1 Fieldbus Map Adr(Byte)	0

RX Poll Delay = 0 wird von der Firmware automatisch auf 1 gesetzt.

Modbus-Request:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Slave ID	Modbus Function	StartAdr High	StartAdr Low	No. of Points High	No. of Points Low	CRC High	CRC Low
1	3	0x00	0x01	0	2	x	y

Der CRC-Wert wird automatisch vom UNIGATE® errechnet

Das UNIGATE® sendet 1 mal (RX Poll Retry = 0) den Request über die RS Schnittstelle raus und wartet maximal 250 ms (Modbus Timeout = 25) auf den Response.

Fieldbus Map Adr = 0 -> nicht aktiv

Dabei hält der angesprochene Slave folgende Daten in seinen Registern vor:

Register	
Adresse	Wert(hex)
40000	0x0000
40001	0x0202
40002	0x0303
40003	0x0000
40004	0x0000

Register = 1 Word = 2 Byte



In manchen Anwendungen wird in der Dokumentation ein Offset + 1 bei der Adresse vorausgesetzt. Die Schreibweise für Adresse „40000“ steht für „Holding Register“. Real ist aber Adresse 0x0000 gemeint. Dies ist in den Modbus-Slave Dokumentationen nicht einheitlich. (Z.B. hat das PC Simulationstool „ModSim32“ diesen Offset).

Wird ein gültiger Response empfangen, werden die vier Byte (No. Of Points = 2) Prozesswerte (Modbus-Data) auf den Feldbus, ab dem „Fieldbus Map Adr(Byte)“ = 0 kopiert.

Feldbusdaten vom UNIGATE® -> SPS:

51 13 02 02 03 03 30 04 01 00 01 00 00 00 02 57 00 01 03 00 00 00 00 00 00 00 ...

Byte 0 = Trigger-Byte „0x51“

Byte 1 = Feldbuslängen-Byte „0x13“

Byte 2 = Prozesswert (High) aus StartAdr „0x02“

Byte 3 = Prozesswert (Low) aus StartAdr „0x02“

Byte 4 = Prozesswert (High) aus StartAdr + 1 „0x03“

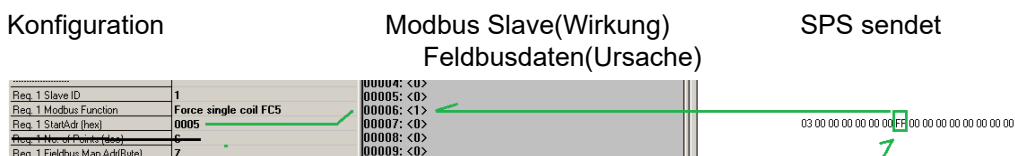
Byte 5 = Prozesswert (Low) aus StartAdr + 1 „0x03“

11.6.3.4 Beispiel: Read input registers FC4

(siehe Kapitel 11.6.3.3, Beispiel: Read multiple register FC3)

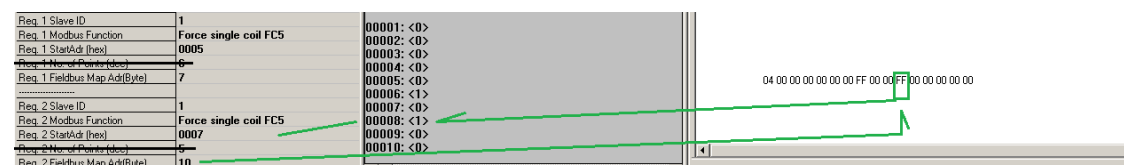
11.6.3.5 Beispiel: Force single coil FC5

Bei FC5 wird ein Bit im Modbus Slave gesetzt, wenn das gemappte Feldbusbyte größer (>) NULL ist.



Hinweis: No. of Points wird nicht benötigt

Ein weiteres Beispiel wenn ein zweiter Request konfiguriert wird:



11.6.3.6 Beispiel: Preset single register FC6

Konfiguration

Req. 1 Slave ID	1
Req. 1 Modbus Function	Preset single register FC6
Req. 1 StartAdr (hex)	0005
Req. 1 Fieldbus Map Adr(Byte)	7

SPS sendet zum UNIGATE®

01 00 00 00 00 00 00 00 FF23 00 FF 00 00 00 00 00 FF 00 00 00 00 00 00 00 00 00 00 00 00 00 ...

UNIGATE® sendet Modbus RTU Request

[01] [06] [00] [05] [ff] [23] [99] [e2]

Modbus Slave sendet Response

[01] [06] [00] [05] [ff] [23] [99] [e2]

Speicherinhalt von Modbus Slave nach Response:

40001: <0000H>
40002: <0000H>
40003: <0000H>
40004: <0000H>
40005: <0000H>
40006: <FF23H>
40007: <0000H>
40008: <0000H>
40009: <0000H>
40010: <0000H>

11.6.3.7 Beispiel: Force multiple coils FC15

Konfiguration

Req. 1 Slave ID	1
Req. 1 Modbus Function	Force multiple coils FC15
Req. 1 StartAdr (hex)	0002
Req. 1 No. of Points (dec)	10
Req. 1 Fieldbus Map Adr(Byte)	2

Feldbus Master sendet:

0E 00 FF 05 00...

UNIGATE® sendet Request:

[01] [0f] [00] [02] [00] [0a] [02] [ff] [05] [65] [29]

Modbus Slave sendet Response:

[01] [0f] [00] [02] [00] [0a] [74] [0c]

Speicherinhalt von Modbus Slave nach Response:

```
00001: <0> 00011: <1>
00002: <0> 00012: <0>
00003: <1> 00013: <0>
00004: <1> 00014: <0>
00005: <1> 00015: <0>
00006: <1> 00016: <0>
00007: <1> 00017: <0>
00008: <1> 00018: <0>
00009: <1> 00019: <0>
00010: <1> 00020: <0>
```

Hex	FF	05
Bin		00000 0
Position	8 7 6 5 4 3 2 1	11 10 9

Bitte beachten sie das No. Of coils = 10 ist, deswegen wird bei dem Wert 0x05 nur das untere Bit in Adresse 0011 geschrieben, Adress 0013 wäre schon das Bit Nr 11 das nicht mehr übertragen wird.

11.6.3.8 Beispiel: Preset multiple register FC16

Konfiguration

Req. 1 Slave ID	1
Req. 1 Modbus Function	Preset multiple register FC16
Req. 1 StartAdr (hex)	0002
Req. 1 No. of Points (dec)	10
Req. 1 Fieldbus Map Adr(Byte)	2

Feldbus Master sendet:

BA 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 ...

UNIGATE® sendet Request:

[01] [10] [00] [02] [00] [0a] [14] [01] [02] [03] [04] [05] [06] [07] [08] [09] [0a] [0b] [0c] [0d] [0e] [0f]...
... [10] [11] [12] [13] [14] [3d] [e4]

Modbus Slave sendet Response:

[01] [10] [00] [02] [00] [0a] [e1] [ce]

Speicherinhalt von Modbus Slave nach Response:

```
40001: <0000H>
40002: <0000H>
40003: <0102H>
40004: <0304H>
40005: <0506H>
40006: <0708H>
40007: <090AH>
40008: <0B0CH>
40009: <0D0EH>
40010: <0F10H>
40011: <1112H>
40012: <1314H>
40013: <0000H>
```


11.7 Protokoll „Universal Modbus ASCII Master/Slave“

Der Feldbusdatenaustausch für Modbus ASCII ist identisch mit RTU. Das UNIGATE® überträgt automatisch auf der seriellen Seite die Daten im ASCII Format.

Protokollbeschreibung siehe Kapitel 11.5, Protokoll „Universal Modbus RTU Slave“ bzw. Kapitel 11.6., Protokoll „Universal Modbus RTU Master“.

11.8 Protokoll Delta exchange

Das Protokoll Delta exchange kann erst ab der Version 1.9.0 des Universalscript Deutschmann genutzt werden.

Mit diesem Protokoll können Daten gezielt in einen Bereich der Feldbusausgangseite, also vom UNIGATE® zur SPS, kopiert werden.

Dazu werden zwei Steuerparameter, dem AdressOffset und der Anzahl Bytes, am Anfang der Übertragung benötigt, bevor dann die eigentlichen Nutzdaten folgen. Zuerst kommt der AdressOffset (Word), dann die Anzahl Bytes (Word), dann die Nutzdaten als Byte-Array.

Die beiden Parameter (AdressOffset und Anzahl Bytes) müssen bei jeder Datenübertragung mitgesendet werden.

1. Byte AdressOffset High
2. Byte AdressOffset Low
3. Byte Anzahl Bytes High
4. Byte Anzahl Bytes Low
5. Byte Nutzdaten 1. Byte
6. Byte " 2. Byte
7. " ...

Beispiel:

1. Byte 0x00
2. Byte 0x02
3. Byte 0x00
4. Byte 0x05
5. Byte 0x01
6. Byte 0x02
7. Byte 0x03
8. Byte 0x04
9. Byte 0x05

In den Feldbusausgangsdaten (z.B. PROFIBUS) werden die Daten wie folgt an die SPS gesendet: 0x00 0x00 0x01 0x02 0x03 0x04 0x05 0x00 ...

11.9 Protokoll SSI

Mit dem Protokoll SSI können z.B. SSI-Geber mit dem UNIGATE® ausgewertet werden und diese Informationen an die übergeordnete Steuerung weitergeleitet werden. Über Parameter können, der Geber-Type, die Geber-Auflösung, die Taktfrequenz und eine ERROR-Bit (wenn unterstützt) entsprechend dem eingesetzten SSI-Geber konfiguriert werden. Siehe auch Kapitel 6 (SSI-Schnittstelle).

12 Implementierte Protokolle mit Universalscript - Protokolle CANopen/CAN Layer 2

Nachfolgend finden Sie die Beschreibungen zu den implementierten Protokollen, die für den Bus verwendet werden können.

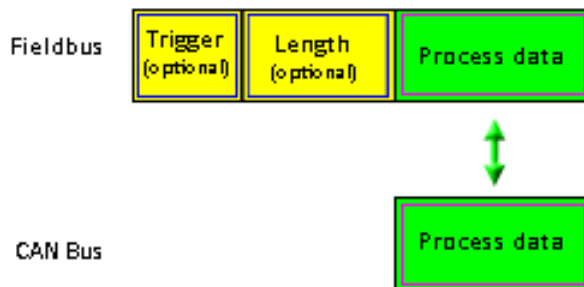
12.1 Protokoll CANopen (default)

siehe Kapitel 8, Funktionsweise des Systems.

Hinweis: Die Protokolle zu CAN Layer 2 können erst ab der Version 1.9.0 des Universalscript Deutschmann genutzt werden.

12.2 Protokoll CAN 2.0A

Datenaufbau



Feldbus Triggerbyte (d.h. Fieldbus Data exchange = On Trigger), Feldbuslängenbyte und Swap word.

Die Scriptdurchlaufzeit erhöht sich da zusätzliche Routinen durchlaufen werden.

12.3 Protokoll CAN 2.0A with ID-filter

Zusätzliche Konfigurationen sind möglich:

Es können bis zu 16 COB-ID's konfiguriert werden, auf die die CAN Schnittstelle reagieren soll. Alle anderen werden gefiltert (nicht bearbeitet).

Steht im ersten Eintrag „Accepted ID 1“ der Wert „0000“ wird auch auf Nachrichten mit COB-ID NULL reagiert.

Steht in einem der anderen Einträge „Accepted ID 2 ... 16“ der Wert NULL, gilt dies als Ende der Liste, alle folgenden Einträge werden nicht bearbeitet.

Beispiel:

-----FELDBUS-----	
CAN Transport protocol	CAN 2.0A with ID filter
Accepted ID 1	0000
Accepted ID 2	0181
Accepted ID 3	0201
Accepted ID 4	0000
Accepted ID 5	0000
Accepted ID 6	0000
Accepted ID 7	0000
Accepted ID 8	0000
Accepted ID 9	0000
Accepted ID 10	0000
Accepted ID 11	0000
Accepted ID 12	0000
Accepted ID 13	0000
Accepted ID 14	0000
Accepted ID 15	0000
Accepted ID 16	0000

In diesem Beispiel werden nur die COB-ID's „0000“ (hex), „0181“ (hex), „0201“ (hex) und „0080“ (hex) bearbeitet, d.h. auf den Feldbus durchgereicht.

12.4 Protokoll CAN 2.0B

Es werden maximal 13 Byte Daten auf der Feldbusseite abgebildet (1 Byte Frameinfo, 4 Byte COB-ID, 8 Byte Nutzdaten).

Es sind zusätzliche Konfigurationen auf der Feldbusseite möglich:

Feldbus Triggerbyte (d.h. Fieldbus Data exchange = On Trigger), Feldbuslängenbyte und Swap word.

Datenabbildung auf der Feldbusseite (SPS empfängt von CAN) “11 Bit Mode”:

1. Byte Anzahl empfangener Daten von CAN(Low Nibble)
MSB = 0 => 11Bit Frame received
2. Byte not used
3. Byte not used
4. Byte COB-ID High Byte
5. Byte COB-ID Low Byte
6. Byte Data n
7. Byte Data n+1
8. ...

Beispiel 1. Zeile: 8 Nutzdaten, COB-ID 181, Nutzdaten 11 ... 88

Beispiel 2. Zeile: 3 Nutzdaten, COB-ID 203, Nutzdaten 01 02 03

08 00 00 01 81 11 22 33 44 55 66 77 88
03 00 00 02 03 01 02 03 00 00 00 00 00

Datenabbildung auf der Feldbusseite (SPS empfängt von CAN) "29 Bit Mode":

1. Byte Anzahl empfangener Daten von CAN (Low Nibble)
 Bit 7 (MSB) = 1 => 29Bit Frame received
 Bit 6 = RTR
 Bit 5 ... 0 = not used
2. Byte COB-ID High Byte
3. Byte COB-ID
4. Byte COB-ID
5. Byte COB-ID Low Byte
6. Byte Data n
7. Byte Data n+1
8. ...

Beispiel 1. Zeile: MSB = 1 => 29 Bit Frame, 8 Nutzdaten, COB-ID 181, Nutzdaten 11 ... 88

Beispiel 2. Zeile: MSB = 1 => 29 Bit Frame, 3 Nutzdaten, COB-ID 203, Nutzdaten 01 02 03

88 00 00 01 81 11 22 33 44 55 66 77 88
83 00 00 02 03 01 02 03 00 00 00 00 00

SPS sendet ein 29 Bit Datenframe an COB-ID 456, 5 Nutzdaten, Daten 01 02 03 04 05

Beispiel:

SPS (Out): 85 00 00 04 56 01 02 03 04 05

Auf CAN Bus:

ID (hex)	Daten (hex)
456	01 02 03 04 05 06 20 20

12.5 Protokoll CAN 2.0B with ID-filter

Identisch dem Protokoll CAN 2.0B, siehe Kapitel 12.4, mit zusätzlicher Konfigurationsmöglichkeit. Es können bis zu 16 COB-ID's konfiguriert werden, auf die die CAN Schnittstelle reagieren soll wie in Kapitel 12.3 beschrieben.

13 Auslieferungszustand

Im Auslieferungszustand ist das UNIGATE® auf der Feldbusseite auf das Protokoll CANopen (default) und auf der Applikationsseite auf das Protokoll Transparent eingestellt.

14 Das Triggerbyte

Werden Daten über den Bus zyklisch übertragen, was normalerweise nicht der Fall ist, muss das Gateway erkennen, wann der Anwender neue Daten über die serielle Schnittstelle verschicken will. Aus diesem Grund kann der Anwender einstellen, dass er über ein Triggerbyte das Senden steuern will (Data exchange -> On Trigger). In diesem Modus sendet das Gateway immer (und nur dann), wenn das Triggerbyte verändert wird.

Wenn der Triggerbyte-Modus eingeschaltet ist, inkrementiert das Gateway das Triggerbyte jedesmal, wenn ein Telegramm empfangen wurde.

Als Triggerbyte wird das erste Byte im Bus-Ein-/Ausgangsdatenpuffer verwendet, wenn dieser Modus eingeschaltet ist.

15 Das Längenbyte

Es kann konfiguriert werden, ob die Sendelänge als Byte im Ein-/Ausgangsdatenbereich mit abgelegt wird (Fieldbus lengthbyte -> active). In Senderichtung werden soviele Bytes verschickt, wie in diesem Byte angegeben sind. Beim Empfang eines Telegramms trägt das Gateway die Anzahl empfangener Zeichen ein.

16 Hardware-Anschlüsse, Schalter und Leuchtdioden

16.1 Gerätebeschriftung

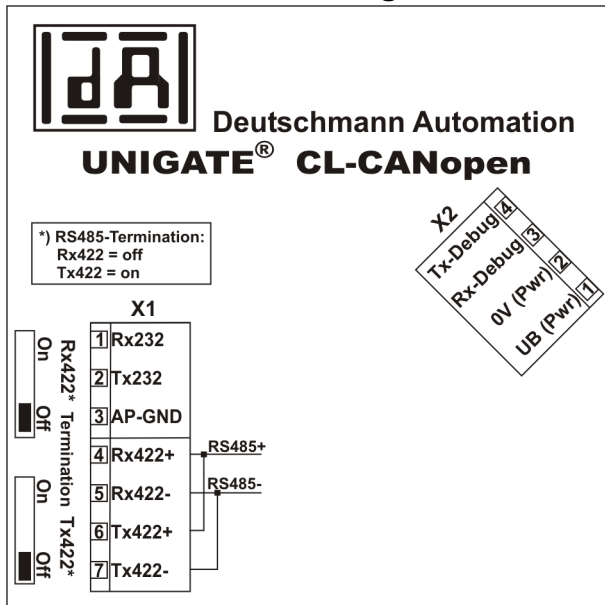


Bild 1: Anschlussbeschriftung und Terminierung

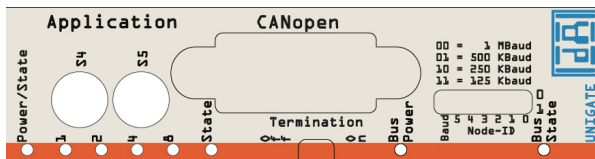


Bild 2: Frontblende: Drehschalter, DIP-Switch, Leuchtanzeigen und Terminierung CO



Sollte die Frontblende herausspringen, so hat dies keinen Einfluss auf die Funktion oder die Qualität des Gerätes. Sie kann einfach wieder eingesetzt werden.

16.2 Stecker

16.2.1 Stecker zum externen Gerät (RS-Schnittstelle)

An dem an der Oberseite des Gerätes zugänglichen Stecker ist die serielle Schnittstelle verfügbar.

Pinbelegung X1 (3-pol. + 4-pol. Schraub-Steckverbinder)

Pin Nr.	Name	Funktion
1	Rx 232	Empfangssignal
2	Tx 232	Sendesignal
3	AP-GND	Applikation Ground
4	Rx 422+ (485+)	Empfangssignal
5	Rx 422- (485-)	Empfangssignal
6	Tx 422+ (485+)	Sendesignal
7	Tx 422- (485-)	Sendesignal



Für den Betrieb an einer 485-Schnittstelle müssen die beiden Pins mit der Bezeichnung "485-" zusammen angeschlossen werden. Ebenso die beiden Pins "485+".

16.2.2 Stecker Versorgungsspannung und DEBUG-Schnittstelle

Pinbelegung X2 (4-pol. Schraub-Steckverbinder; an der Unterseite, hinten)

Pin Nr.	Name	Funktion
1	UB (Pwr)	10..33 Volt Versorgungsspannung / DC
2	0 V (Pwr)	0 Volt Versorgungsspannung / DC
3	Rx-Debug	Empfangssignal Debug
4	Tx-Debug	Sendesignal Debug



Achtung:

Der Ground für die DEBUG-Schnittstelle muss bei Geräten mit Potentialtrennung (Option GT) mit Pin 3 (AP-GND) der RS-Schnittstelle verbunden werden!

Bei Geräten ohne Potentialtrennung kann als Bezug auch das 0V (Pwr) DC-Signal verwendet werden.

16.2.3 CANopen®-Stecker

An der Vorderseite des Gerätes ist der Stecker (Beschriftung: CANopen®) zum Anschluss an CANopen®.

Pinbelegung (9pol. D-SUB Stecker)

Pin Nr.	Name	Funktion
1		
2	CAN-L	Dominant Low
3	CAN-GND	CAN Ground
4		
5		
6		
7	CAN-H	Dominant High
8		
9		

16.2.4 Stromversorgung

Das Gerät ist mit 10-33 VDC zu versorgen. Die Spannungsversorgung erfolgt über den 4pol. Schraub-/Steckverbinder an der Unterseite.

Bitte beachten Sie, dass Geräte der Serie UNIGATE® nicht mit Wechselspannung (AC) betrieben werden sollten.

16.3 Leuchtanzeigen

Das Gateway UNIGATE® CL - CANopen® verfügt über 8 LEDs mit folgender Bedeutung:

LED (Bus) Power	grün	Versorgungsspannung CANopen®
LED (Bus) State	rot/grün	Schnittstellenzustand CANopen®
LED Power	grün	Versorgungsspannung serielle Schnittstelle
LED State	rot/grün	benutzerdefiniert / allgemeiner Gatewayfehler
LED 1 / 2 / 4 / 8 (Error No / Select ID)	grün	benutzerdefiniert / allgemeiner Gatewayfehler

16.3.1 LED “(Bus) Power“

Diese LED ist direkt mit der (potentialgetrennten) Versorgungsspannung der CANopen®-Seite verbunden.

16.3.2 LED “(Bus) State“

Anzeigezustände und Blinkfrequenzen

Die folgenden Anzeigezustände werden unterschieden:

LED an	Konstant an
LED aus	Konstant aus
LED flackernd	Iso-Phase an und aus mit einer Frequenz von ungefähr 10 Hz: an für ca. 50 ms und aus für ca. 50 ms.
LED blinkend	Iso-Phase an und aus mit einer Frequenz von ungefähr 2,5 Hz: an für ca. 200 ms, danach aus für ca. 200 ms.
LED Einzel-Blinken	Kurzes Blinken (ca. 200 ms) gefolgt von einer langen Aus-Phase (ca. 1000 ms)
LED Doppel-Blinken	Eine Folge von zweimaligem kurzen Blinken (ca. 200 ms), getrennt durch eine Aus-Phase (ca. 200 ms). Die Folge wird beendet mit einer langen Aus-Phase (ca. 1000 ms).
LED Dreifach-Blinken	Eine Folge von dreimaligem kurzen Blinken (ca. 200 ms), getrennt durch eine Aus-Phase (ca. 200 ms). Die Folge wird beendet mit einer langen Aus-Phase (ca. 1000 ms).

Anstatt zwei einfarbiger LEDs wird eine zweifarbige Status-LED verwendet, die sowohl den physischen BUS Status als auch den Status der CANopen® Zustandsmaschine anzeigen soll. Diese zweifarbige LED ist rot und grün.

CANopen® Fehler LED (rot)

Die CANopen® Fehler LED zeigt den Zustand des physikalischen CAN-Layers und Fehler auf Grund von fehlenden CAN Nachrichten an (SYNC, GUARD oder HEARTBEAT).

Fehler LED	Zustand	Beschreibung
Aus	kein Fehler	Das Gerät befindet sich im betriebsfähigen Zustand.
Einzel-Blinken	Warn-Grenze erreicht	Mindestens einer der Fehlerzähler des CAN-Controllers hat die Warn-Grenze erreicht oder überschritten (zu viele Fehlerrahmen).
Flackernd	AutoBaud/LSS	Auto Baudraten-Erkennung in Bearbeitung oder LSS Service in Bearbeitung (abwechselnd flackernd mit RUN LED).
Doppel-Blinken	Fehlerkontroll-Ereignis	Ein "Guard"-Vorgang (NMT-Slave oder NMT-Master) oder ein heartbeat-Ereignis (Heartbeat "Consumer") fand statt.
Dreifach-Blinken	Sync Fehler	Die SYNC Mitteilung wurde nicht innerhalb der konfigurierten Kommunikationszyklus Auszeit erhalten (siehe Objekt Lexikon Eintrag 0x1006).
An	Bus aus	Der CAN Controller ist "Bus aus".

Wenn bei einer vorgegebenen Zeit mehrere Fehler vorhanden sind, wird der Fehler mit der höchsten Nummer angegeben (z. B. wenn NMT Fehler und Sync Fehler erscheinen, wird der SYNC Fehler angezeigt).

CANopen® RUN LED (grün)

Die CANopen® RUN LED zeigt den Zustand der CANopen®-Netzwerk Zustandsmaschine an.

CAN RUN LED	Zustand	Beschreibung
Flackernd	AutoBaud/LSS	Auto Baudraten-Erkennung in Bearbeitung oder LSS Service in Bearbeitung (abwechselnd flackernd mit Fehler LED)
Einzel-Blinken	Gestoppt	Das Gerät befindet sich im Stopzustand.
Blinkend	PREOPERATIONAL	Das Gerät befindet sich im Zustand "PREOPERATIONAL".
An	OPERATIONAL	Das Gerät befindet sich im Zustand "OPERATIONAL".

Während das Gerät einen Neustart durchführt, sollte die CANopen® RUN LED aus sein.

Sollte der Fall auftreten, dass sich die grüne und rote LED entgegenstehen, wird die LED auf rot wechseln. Abgesehen von diesem Fall sollte die zweifarbige Status LED das Verhalten der CAN Fehler LED und der CAN RUN LED miteinander verknüpfen.

Alte Hardware (mit RD2-Prozessor) - geliefert bis Februar 2009:

grün leuchtend	CAN-Status = OPERATIONAL
grün blinkend	CAN-Status = PREOPERATIONAL oder PREPARED
rot blinkend	Guarding-Error
rot leuchtend	CAN-Busfehler

16.3.3 LED "Power"

Diese LED ist direkt mit der (optional auch potentialgetrennten) Versorgungsspannung der seriellen Schnittstelle (RS232/422/485) verbunden.

16.3.4 LED "State"

grün leuchtend	Datenaustausch aktiv über RS422/RS485/RS232; über Script steuerbar
grün blinkend	RS422/RS485/RS232 O.K., aber kein ständiger Datenaustausch; über Script steuerbar
grün/rot blinkend	Noch kein Datenaustausch seit Einschalten; über Script steuerbar
rot leuchtend	Allgemeiner Gatewayfehler (s. LED's Error No.), über Script steuerbar
rot blinkend	UNIGATE befindet sich im Konfigurations-/Testmodus, über Script steuerbar

16.3.5 LEDs 1 / 2 / 4 / 8 (Error No / Select ID)

Blinken diese 4 LED's und die LED "State" leuchtet gleichzeitig rot, wird binärcodiert (Umrechnungstabelle siehe Anhang) gemäß der Tabelle im Kapitel "Fehlerbehandlung" die Fehlernummer angezeigt. Zusätzlich sind diese LEDs über Script steuerbar.

16.4 Schalter

Das Gateway verfügt über 6 Schalter mit folgenden Funktionen:

Termination Rx 422	schaltbarer Rx 422-Abschlusswiderstand für die serielle Schnittstelle
Termination Tx 422	schaltbarer Tx 422- bzw. RS485-Abschlusswiderstand für die serielle Schnittstelle
Drehcodierschalter S4	ID High für serielle Schnittstelle z. B. Konfigmode
Drehcodierschalter S5	ID Low für serielle Schnittstelle z. B. Konfigmode
Termination (CANopen®)	schaltbarer CANopen®-Abschlusswiderstand
DIP-Switch	Node-ID und Baudrate

16.4.1 Termination Rx 422 + Tx 422 (serielle Schnittstelle)

Wird das Gateway als physikalisch erstes oder letztes Gerät in einem RS485-Bus bzw. als 422 betrieben, muss an diesem Gateway ein Busabschluss erfolgen. Dazu wird der Terminationschalter auf ON gestellt. Der im Gateway integrierte Widerstand (150Ω) wird aktiviert. In allen anderen Fällen bleibt der Schalter auf der Position OFF.

Nähere Information zum Thema Busabschluss finden Sie in der allgemeinen RS485 Literatur.

Wird der integrierte Widerstand verwendet ist zu berücksichtigen, dass damit automatisch ein Pull-Down-Widerstand (390Ω) nach Masse und ein Pull-Up-Widerstand (390Ω) nach VCC aktiviert wird.



**Bei RS485 darf nur der Tx 422-Schalter auf ON gestellt werden.
Der Rx 422-Schalter muss auf OFF stehen.**

16.4.2 Drehcodierschalter S4 + S5 (serielle Schnittstelle)

Diese beiden Schalter können über den Scriptbefehl `Get (RS_Switch, Destination)` ausgelesen und der Wert für weitere Funktionen weiter verwendet werden. Dieser Wert wird beim Einschalten des Gateways bzw. immer wieder nach Ausführen des Scriptbefehls eingelesen. Die Schalterstellung „EE“ (testmode) und „FF“ (config mode) sind bei der RS422- oder RS485-Betrieb nicht möglich.

Hinweis: Die Schalterstellung „DD“ (d.h. S4 und S5 beide in Stellung "D") ist für interne Zwecke reserviert.

16.4.3 Termination (CANopen®)

Wird das Gateway als physikalisch erstes oder letztes Gerät im CANopen® betrieben, muss an diesem Gateway ein Busabschluss erfolgen. Dazu muss entweder ein Busabschlusswiderstand im Stecker oder der im Gateway integrierte Widerstand (220Ω) aktiviert werden. Dazu wird der Schiebeschalter auf die Position ON geschoben. In allen anderen Fällen bleibt der Schiebeschalter auf der Position OFF. Nähere Information zum Thema Busabschluss finden Sie in der allgemeinen Feldbus Literatur.

Hinweis: Um den Busabschluss zu aktivieren bzw. deaktivieren, bitte den Bus-Stecker abziehen und den Schalter vorsichtig in die gewünschte Position bringen.

16.4.4 DIP-Switch

Über diesen DIP-Switch wird die Node-ID und die Baudrate (gemäß Bild 3) eingestellt.

Wird auf dem DIP-Switch die Node-ID 0 eingestellt, die in CANopen® nicht erlaubt ist, dann wird in diesem Fall die Node-ID verwendet, die im EEROM gespeichert ist und über das Script bzw. WINGATE dorthin geschrieben wurde. Damit hat man einen Weg auch Node-IDs > 63 einzustellen und es gibt keine Unklarheiten, ob nun die Node-ID des DIP-Switch oder des Scriptes gültig ist.

Im Auslieferungszustand ist die Node-ID 1 im EEROM gespeichert.

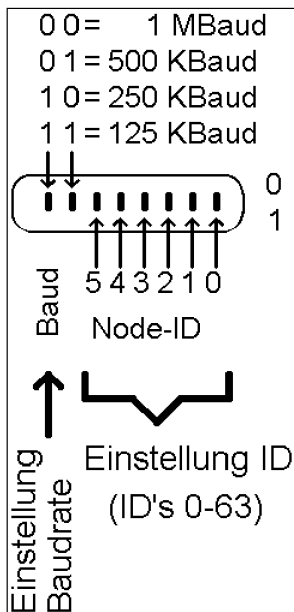


Bild 3: DIP-Switch

16.5 Debug-Kabel für UNIGATE® CL

Als Zubehör ist ein vorkonfiguriertes Debugkabel erhältlich. Das Debug Kabel verbindet das Gateway zu Debug und RS.

17 Fehlerbehandlung

17.1 Fehlerbehandlung beim UNIGATE® CL

Erkennt das Gateway einen Fehler, so wird dieser dadurch signalisiert, dass die LED "State" rot leuchtet und gleichzeitig die Fehlernummer gemäß nachfolgender Tabelle über die LED's "Error No" angezeigt werden. Zusätzlich wird im default Zustand über CANopen® diese Fehlernummer als Emergency-Telegramm gesendet. Dazu wird der Code 61xx (Hex) verwendet, der gemäß DS301 einen internen Firmwarefehler kennzeichnet. Für "xx" wird der aktuelle Fehlercode eingesetzt gemäß anhängender Tabelle. Im Byte 3 und 4 der Emergency-Message kann noch ein Detail-Errorcode enthalten sein, der für interne Zwecke genutzt wird.

Es können zwei Fehlerkategorien unterschieden werden:

Schwere Fehler (1-5): In diesem Fall muss das Gateway aus- und wieder neu eingeschaltet werden. Tritt der Fehler erneut auf, muss das Gateway getauscht und zur Reparatur eingeschickt werden.

Warnungen (6-15): Diese Warnungen werden lediglich zur Information 1 Minute lang angezeigt und werden dann automatisch zurückgesetzt. Treten diese Warnungen häufiger auf, ist der Kundendienst zu verständigen.

Beim Heartbeat-Error wird zusätzlich der Fehler 10 aktiviert!

Im Konfigurationsmodus sind diese Anzeigen nicht gültig und nur für interne Zwecke bestimmt.

LED8	LED4	LED2	LED1	Fehlernr. bzw. ID	Fehlerbeschreibung
0	0	0	0	0	Reserviert
0	0	0	1	1	Hardwarefehler
0	0	1	0	2	EEROM-Fehler
0	0	1	1	3	Interner Speicherfehler
0	1	0	0	4	FeldbusHardwarefehler oder falsche Feldbus ID
0	1	0	1	5	Script-Fehler
0	1	1	0	6	Reserviert
0	1	1	1	7	RS Sende-Puffer-Überlauf
1	0	0	0	8	RS Empfangs-Puffer-Überlauf
1	0	0	1	9	RS Timeout
1	0	1	0	10	Allgemeiner Feldbusfehler
1	0	1	1	11	Parity- oder Frame-Check-Fehler
1	1	0	0	12	Reserviert
1	1	0	1	13	Feldbus Konfigurationsfehler
1	1	1	0	14	Feldbus Datenpuffer-Überlauf
1	1	1	1	15	Reserviert

Tabelle 1: Fehlerbehandlung beim UNIGATE® CL - Systemfehler

Blinkfrequenz 2-Mal pro Sekunde (Systemfehler)

LED8	LED4	LED2	LED1	Fehlernr.	Protokoll	Fehlerbeschreibung
0	0	1	1	3	alle Protokolle	keine Universalscript Unterstützung
0	1	0	1	5	alle Protokolle	Unbekanntes Protokoll
1	0	0	1	9	Modbus RTU Master Modbus ASCII Master	Timeout-Modbus Slave Teilnehmer hat nicht innerhalb der festgesetzten Zeit (Reponsetime) geantwortet.
					Modbus RTU Slave Universal Modbus RTU Slave	Timeout bei Response-Sendung
					3964(R)	Timeout - keine Antwort von Teilnehmer
1	0	1	1	11	Universal 232 (mit 232 Checksum)	Checksumme von Empfang stimmt nicht mit nachgerechneter überein.
					Modbus RTU Slave Universal Modbus RTU Slave	Unbekannter Fehler nach Response-Sendung
					3964(R)	Fehler bei Datenaustausch (z.B. Checksummenfehler)
1	1	0	0	12	Universal Modbus RTU Master	Error in Response von Funktionscode
1	1	0	0	12	SSI	Fehler bei SSI Kommunikation
1	1	1	1	15	Modbus RTU Master Modbus ASCII Master	Allgemeiner Empfangsfehler bei Modbus (ASCII) Exchange, z.B. Checksummenfehler
1	1	1	0	14	Modbus RTU Slave	Exception Response
1	1	1	1	15	alle Protokolle	interner Fehler bei Prozessdatenverarbeitung

Tabelle 2: Protokollbezogene Fehler

Blinkfrequenz 1-Mal pro Sekunde (benutzerdefinierte Fehler bzw. protokollbezogene Fehler).

Hinweis: Der Fehler wird solange angezeigt wie mit „Set Warning Time“ definiert ist.

18 Aufbaurichtlinien

18.1 Montage der Baugruppe

Die Baugruppe mit den max. Abmessungen (23x117x111mm BxTxH) ist für den Schaltschrank-einsatz (IP20) entwickelt worden und kann deshalb nur auf einer Normprofilschiene (tiefe Hutschiene nach EN50022) befestigt werden.

18.1.1 Montage

- Die Baugruppe wird von oben in die Hutschiene eingehängt und nach unten geschwenkt bis die Baugruppe einrastet.
- Links und rechts neben der Baugruppe dürfen andere Baugruppen aufgereiht werden.
- Oberhalb und unterhalb der Baugruppe müssen mindestens 5 cm Freiraum für die Wärmeabfuhr vorgesehen werden.
- Die Normprofilschiene muss mit der Potentialausgleichschiene des Schaltschranks verbunden werden. Der Verbindungsdraht muss einen Querschnitt von mindestens 10 mm² haben.

18.1.2 Demontage

- Zuerst müssen die Versorgungs- und Signalleitungen abgesteckt werden.
- Danach muss die Baugruppe nach oben gedrückt und die Baugruppe aus der Hutschiene herausgeschwenkt werden.

Senkrechter Einbau

Die Normprofilschiene kann auch senkrecht montiert werden, so dass die Baugruppe um 90° gedreht montiert wird.

18.2 Verdrahtung

18.2.1 Anschlusstechniken

Folgende Anschlusstechniken müssen bzw. können Sie bei der Verdrahtung der Baugruppe einsetzen:

- Standard-Schraub-/Steck-Anschluss (Versorgung + RS)
- 9pol. D-SUB Steckverbinder (CANopen®)

a) Bei den Standard-Schraubklemmen ist eine Leitung je Anschlusspunkt klemmbar. Zum Festschrauben benutzen Sie am besten einen Schraubendreher mit Klingenbreite 3,5 mm.

Zulässige Querschnitte der Leitung:

- Flexible Leitung mit Aderendhülse: 1 x 0,25 ... 1,5 mm²
- Massive Leitung: 1 x 0,25 ... 1,5 mm²
- Anzugsdrehmoment: 0,5 ... 0,8 Nm

b) Die steckbare Anschlussklemmleiste stellt eine Kombination aus Standard-Schraubanschluss und Steckverbinder dar. Der Steckverbindungsteil ist kodiert und kann deshalb nicht falsch aufgesteckt werden.

c) Der 9-polige D-SUB Steckverbinder wird mit zwei Schrauben mit "4-40-UNC"-Gewinde gesichert. Zum Festschrauben benutzen Sie am besten einen Schraubendreher mit Klingenbreite 3,5 mm.

Anzugsdrehmoment: 0,2 ... 0,4 Nm

18.2.1.1 Stromversorgung

Das Gerät ist mit 10..33VDC zu versorgen.

- Schließen Sie die Versorgungsspannung an die 4-polige Steckschraubklemme entsprechend der Beschriftung auf dem Gerät an.

18.2.1.2 Anschluss des Potentialausgleichs

Die Verbindung zum Potentialausgleich erfolgt automatisch beim Aufsetzen auf die Hutschiene.

18.2.2 Kommunikationsschnittstelle CANopen®

18.2.2.1 Busleitung mit Kupferkabel

Diese Schnittstelle finden Sie auf der Baugruppe in Form eines 9-poligen D-SUB-Steckers an der Frontseite des Gehäuses.

- Stecken Sie den CANopen®-Verbindungsstecker auf den D-SUB-Stecker mit der Beschriftung "CANopen®".
- Schrauben Sie die Sicherungsschrauben des Verbindungsstecker mit einem Schraubendreher fest.
- Befindet sich die Baugruppe am Anfang oder am Ende der CANopen®-Leitung, so müssen Sie den im Gateway integrierten Busabschlusswiderstand zuschalten. Schieben Sie dazu den Schiebeschalter in die Stellung mit der Beschriftung ...on...
- Befindet sich die Baugruppe nicht am Anfang oder am Ende, so müssen Sie den Schiebeschalter in die Stellung "off" schieben.

18.2.3 Leitungsführung, Schirmung und Maßnahmen gegen Störspannung

Gegenstand dieses Kapitels ist die Leitungsführung bei Bus-, Signal- und Versorgungsleitungen mit dem Ziel, einen EMV-gerechten Aufbau Ihrer Anlage sicherzustellen.

18.2.4 Allgemeines zur Leitungsführung

- innerhalb und außerhalb von Schränken

Für eine EMV-gerechte Führung der Leitungen ist es zweckmäßig, die Leitungen in folgende Leitungsgruppen einzuteilen und diese Gruppen getrennt zu verlegen.

⇒ Gruppe A: • geschirmte Bus- und Datenleitungen z.B. für PROFIBUS DP, RS232C, Drucker, etc.

- geschirmte Analogleitungen
- ungeschirmte Leitungen für Gleichspannungen ≥ 60 V
- ungeschirmte Leitungen für Wechselspannung ≥ 25 V
- Koaxialleitungen für Monitore

⇒ Gruppe B: • ungeschirmte Leitungen für Gleichspannungen ≥ 60 V und ≥ 400 V
• ungeschirmte Leitungen für Wechselspannung ≥ 24 V und ≥ 400 V

⇒ Gruppe C: • ungeschirmte Leitungen für Gleichspannungen > 400 V

Anhand der folgenden Tabelle können Sie durch die Kombination der einzelnen Gruppen die Bedingungen für das Verlegen der Leitungsgruppen ablesen.

	Gruppe A	Gruppe B	Gruppe C
Gruppe A	1	2	3
Gruppe B	2	1	3
Gruppe C	3	3	1

Tabelle: Leitungsverlegevorschriften in Abhängigkeit der Kombination von Leitungsgruppen

- 1) Leitungen können in gemeinsamen Bündeln oder Kabelkanälen verlegt werden.
- 2) Leitungen sind in getrennten Bündeln oder Kabelkanälen (ohne Mindestabstand) zu verlegen.
- 3) Leitungen sind innerhalb von Schränken in getrennten Bündeln oder Kabelkanälen und außerhalb von Schränken aber innerhalb von Gebäuden auf getrennten Kabelbahnen mit mindestens 10 cm Abstand zu verlegen.

18.2.4.1 Schirmung von Leitungen

Das Schirmen ist eine Maßnahme zur Schwächung (Dämpfung) von magnetischen, elektrischen oder elektromagnetischen Störfeldern.

Störströme auf Kabelschirmen werden über die mit dem Gehäuse leitend verbundene Schirmschiene zur Erde abgeleitet. Damit diese Störströme nicht selbst zu einer Störquelle werden, ist eine impedanzarme Verbindung zum Schutzleiter besonders wichtig.

Verwenden Sie möglichst nur Leitungen mit Schirmgeflecht. Die Deckungsdichte des Schirmes sollte mehr als 80% betragen. Vermeiden Sie Leitungen mit Folienschirm, da die Folie durch Zug- und Druckbelastung bei der Befestigung sehr leicht beschädigt werden kann; die Folge ist eine Verminderung der Schirmwirkung.

In der Regel sollten Sie die Schirme von Leitungen immer beidseitig auflegen. Nur durch den beidseitigen Anschluss der Schirme erreichen Sie eine gute Störunterdrückung im höheren Frequenzbereich.

Nur im Ausnahmefall kann der Schirm auch einseitig aufgelegt werden. Dann erreichen Sie jedoch nur eine Dämpfung der niedrigeren Frequenzen. Eine einseitige Schirmanbindung kann günstiger sein, wenn,

- die Verlegung einer Potentialausgleichsleitung nicht durchgeführt werden kann
- Analogsignale (einige mV bzw. mA) übertragen werden
- Folienschirme (statische Schirme) verwendet werden.

Benutzen Sie bei Datenleitungen für serielle Kopplungen immer metallische oder metallisierte Stecker. Befestigen Sie den Schirm der Datenleitung am Steckergehäuse.

Bei Potentialdifferenzen zwischen den Erdungspunkten kann über den beidseitig angeschlossenen Schirm ein Ausgleichsstrom fließen. Verlegen Sie in diesem Fall eine zusätzliche Potentialausgleichsleitung.

Beachten Sie bei der Schirmbehandlung bitte folgende Punkte:

- Benutzen Sie zur Befestigung der Schirmgeflechte Kabelschellen aus Metall- Die Schellen müssen den Schirm großflächig umschließen und guten Kontakt ausüben.
- Legen Sie den Schirm direkt nach Eintritt der Leitung in den Schrank auf eine Schirmschiene auf. Führen Sie den Schirm bis zur Baugruppe weiter; legen Sie ihn dort jedoch nicht erneut auf!

19 CANopen®

19.1 Beschreibung CANopen®

Diese Spezifikation baut auf der Norm CiA® Draft Standard 301 (DS301) auf.

CANopen® unterstützt den Standard CAN-Rahmen mit 11-bit Identifier.

Es ist nicht notwendig, den erweiterten CAN-Rahmen mit 29-bit Identifier zu unterstützen.

19.1.1 CANopen® V3



Nur für Altprodukte, aktuell liefern wir CANopen V4.

Ein CANopen V3 Script kann mit geringfügigen Änderungen auf CANopen V4 angepasst werden.

Syntax

CO_InitChannel (vw_Channel , Direction , vw_len , vw_Obj_Nr , vw_COBId)

Beschreibung

Ab Script Rev. 25 und höher erlaubt die CAN Firmware die Definition von Anwenderobjekten und das Mapping von bis zu 16 Rx und 16 Tx PDOs. Für Script Rev. zwischen 22 und 25 können nur bis zu 5 Rx und 5 Tx PDOs benutzt werden.

Vordefinierte Kommunikation

Für manche Anwendungen reicht eine Rx und eine Tx PDO aus.

Es ist möglich, CANopen® ohne die Definition von Kommunikationskanälen zu benutzen. In diesem Fall gibt es folgende Abbildung der Daten:

Data width	Direction	Object	Mapping
1..8 byte	Rx	2000	Default Rx-PDO1 (COB-ID 200 + Node ID)
1..8 byte	Tx	2001	Default Tx-PDO1 (COB-ID 180 + Node ID)
9..255 byte	Rx	2000	Data not mapped (could be read by SDO) no Rx-PDO available
9..255	Tx	2001 (Tx-Data) 2002 (Tx-Length)	Data not mapped (write by SDO) Data width in Tx PDO 1 (COB-ID 180 + Node ID)

Es ist möglich, für den Datenaustausch ReadBus und WriteBus zu benutzen. Dieses Standardverhalten ist nicht mehr aktiv, wenn Sie mindestens einmal CO_InitChannel aufrufen.

Benutzerdefinierte Kommunikation

Dieser Modus ist notwendig, wenn Sie anspruchsvollere CANopen® Funktionen benutzen möchten. Sie müssen für jede PDO oder Objekt einen CANopen®-Kanal initialisieren.

Syntax

CO_InitChannel (vw_Channel , Direction , vw_len , vw_ObjAddress , vw_COBId)

Benutzen Sie folgende Werte als Parameter:

Parameter	Type	Meaning	
vw_Channel	Word	Value	Meaning
		0	Using CAN Layer2, we have no PDO and SDO data access
		1..8	Define PDO 1..8
Direction	-	RX or Tx depending of the data direction. It is seen from the devices view, this means Rx-data is incoming data.	
vw_ByteLen	Byte	Length of the object to use. If the length is > 8 byte only the first 8 bytes are used to transmit by the PDO	
vw_ObjAddress	Word	Allowed values are 0x2000 to 0x5FFF, which is the range of objects to be defined by the user	
vw_COB_ID	Word	Value	Meaning
		0	PDO is not active, data is defined to be used by a SDO transfer only.
		0x181..0x57F	Allowed range for normal Rx and Tx - PDO's
		0xFFFE	Is to be used, if the master defines the COB-ID's when the CANopen network is started by the master (no predefined COB-IDs). The resulting objects used by the device are 0x1800 + (PDO-Nr - 1). Sub-Index 1 of this object contains the COB-ID. After writing a valid value to this object with subindex the requested PDO becomes active. Valid values must be in the range from 0x181 to 0x57F.
		0xFFFF	Is to be used for PDO1 and PDO2. The COB-ID is as defined by the predefined connection set Tx-PDO1: 0x180 + Node-ID Rx-PDO1: 0x200 + Node-ID Tx-PDO2: 0x280 + Node-ID Rx-PDO2: 0x300 + Node-ID

CAN Layer 2

Wenn Sie CAN Layer 2 benutzen wollen, können Sie eine spezielle Script Initialisierung setzen, mit der Sie Zugriff auf jede CAN Nachricht ohne Filter haben. Bitte beachten Sie, dass in diesem Fall das Datenformat für ReadBus und WriteBus von anderen Funktionalitäten abweicht. Ab jetzt muss die COB-ID der Nachricht in den ersten beiden Bytes des Datenbereiches gelesen oder gesendet werden.

Folgende Beispiele finden Sie in dem Ordner "example" nach der Installation der Software "Protocol Developer":

- Example CANopen® 2 PDOs
- Example CAN Layer 2

Schauen Sie sich dazu auch die folgenden Scriptbefehle aus dem "Protocol Developer" an:

- CO_Read PDO
- CO_WriteEmergency
- CO_WritePDO

19.1.2 CANopen® V4

Zusätzlich unterstützte Funktionen

- Heartbeat
- Dynamic mapping

- Onswich message

Folgendes Beispiel finden Sie in dem Ordner "example" nach der Installation der Software "Protocol Developer" (in diesem Beispiel ist die genaue Initialisierung beschrieben):

- Example_CO_V4.dss

Folgende feldbusspezifischen Scriptbefehle werden bei CANopen® V4 unterstützt:

- Init Object Table
- Create Object
- Set PDO Communication
- Set PDO Mapping
- Write Object
- Read New CANopen® Object Data
- Emergency Message

Die Software unterstützt keine Default Objekte, wie bei CANopen® V3 beschrieben.

20 Technische Daten

20.1 Gerätedaten

In der nachfolgenden Tabelle finden Sie die technischen Daten der Baugruppe.

Nr.	Parameter	Daten	Erläuterungen
1	Einsatzort	Schaltschrank	Hutschienenmontage
2	Schutzart	IP20	Fremdkörper und Wasserschutz nach IEC 529 (DIN 40050)
3	Lebensdauer	10 Jahre	
4	Gehäusegröße	23 x 117 x 111 mm (inkl. Schraub-Steckverbinder) 23 x 117 x 100 mm (ohne Schraub-Steckverbinder)	B x T x H
5	Einbaulage	Beliebig	
6	Gewicht	123 g	
7	Betriebstemperatur	-40°C ... +85°C	Die Minustemperaturen gelten nur für die üblichen Bedingungen (nicht kondensierend).
8	Lager-/Transporttemperatur	-40°C ... +85°C	
9	Luftdruck bei Betrieb bei Transport	795 hPa ... 1080 hPa 660 hPa ... 1080 hPa	
10	Aufstellungshöhe	2000 m 4000 m	Ohne Einschränkungen mit Einschränkungen - Umgebungstemperatur ≤ 40°C
11	Relative Luftfeuchte	Max. 80 %	Nicht kondensierend, keine korrosive Atmosphäre
12	Externe Versorgungsspannung	10...33V DC	Standardnetzteil nach DIN 19240
13	Stromaufnahme bei 24VDC	Typ. 50 mA max 60 mA	(bei 10VDC) max. 150 mA
14	Verpolungsschutz	Ja	Gerät funktioniert jedoch nicht!
15	Kurzschlusschutz	Ja	
16	Überlastschutz	Poly-Switch	Thermosicherung
17	Unterspannungserkennung (USP)	≤ 9V DC	
18	Spannungsausfall-Überbrückung	≥ 5 ms	Gerät voll funktionsfähig

Tabelle: Technische Daten der Baugruppe

20.1.1 Schnittstellendaten

In der nachfolgenden Tabelle sind technische Daten der auf dem Gerät vorhandenen Schnittstellen aufgelistet. Die Daten sind den entsprechenden Normen entnommen.

Nr	Schnittstellenbezeichnung physikalische Schnittstelle	CANopen® RS485	RS232-C RS232-C	RS485/RS422 RS485/RS422
1	Norm	CiA® DS 102	DIN 66020	EIA-Standard
2	Übertragungsart	symmetrisch asynchron seriell halbduplex → Differenzsignal	asymmetrisch asynchron seriell vollduplex → Pegel	symmetrisch asynchron seriell halbduplex/ vollduplex bei RS422 → Differenzsignal
3	Übertragungsverfahren	Master / Slave	Master / Slave	Master / Slave
4	Teilnehmerzahl: - Sender - Empfänger	32 32	1 1	32 32
5	Kabellänge: - maximal - baudratenabhängig	1300 m 50 kBd → 1300 m 100 kBd → 640 m 200 kBd → 310 m 500 kBd → 112 m 1 MBd → 40 m	15 m nein	1200 m <93,75 kBd → 1200 m 312, kBd → 500 m 625 kBd → 250 m
6	Bus-Topologie	Linie	Pkt.-zu-Pkt.	Linie
7	Datenrate: - maximal - Standardwerte	1 MBit/s 125 kB 250 kBd 500 kB 1MB	120 kBit/s 2,4 k/B 4,8 k/B 9,6 kBit/s 19,2 kBit/s 38,4 kBit/s	625 kBaud 2,4 kBit/s 4,8 kBit/s 9,6 kBit/s 19,2 kBit/s 57,6 kB 312,5 kB 625 kB
8	Sender: - Belastung - max. Spannung - Signal ohne Belastung - Signal mit Belastung	54 Ω - 7 V ... 12 V ± 5 V ± 1,5 V	3 ... 7 kΩ ± 25 V ± 15 V ± 5 V	54 Ω - 7 V ... 12 V ± 5 V ± 1,5 V
9	Empfänger: - Eingangswiderstand - max. Eingangssignal - Empfindlichkeit	12 Ω - 7 V ... 12 V ± 0,2 V	3 ... 7 Ω ± 15 V ± 3 V	12 Ω - 7 V ... 12 V ± 0,2 V
10	Sendebereich (SPACE): - Spgspegel - Logikpegel	- 0,5 ... + 0,05 V 0	+ 3 ... + 15 V 0	- 0,2 ... + 0,2 V 0
11	Sendepause (MARK): - Spgspegel - Logikpegel	+ 1,5 ... +3 V 1	- 3 ... -15 V 1	+ 1,5 ... +5 V 1

Tabelle: Technische Daten der an der Baugruppe vorhandenen Schnittstellen

21 Inbetriebnahmeleitfaden

21.1 Beachte

Die Inbetriebnahme des UNIGATE® darf nur von geschultem Personal unter Beachtung der Sicherheitsvorschriften durchgeführt werden.

21.2 Komponenten

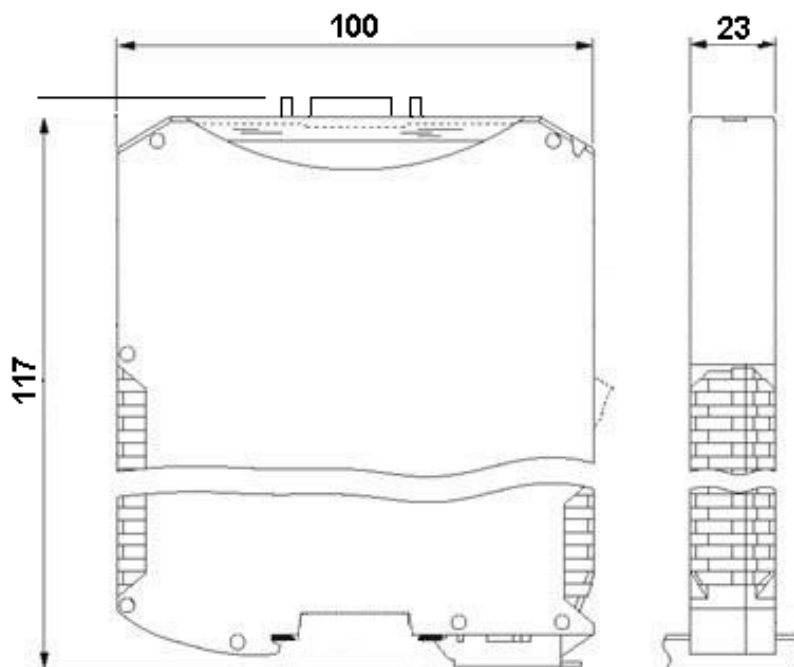
Zur Inbetriebnahme des UNIGATE® benötigen Sie folgende Komponenten:

- UNIGATE®
- Verbindungskabel vom Gateway zum Prozess hin
- Verbindungsstecker für den CANopen®-Anschluss an das Gateway
- CANopen®-Kabel (Dieses Kabel ist in der Regel bereits vorort installiert!)
- 10..33 VDC-Spannungsversorgung (DIN 19240)
- Typ- bzw. EDS-Datei und Betriebsanleitung (eine Muster-EDS-Datei sowie das Handbuch können separat bestellt oder kostenfrei aus dem Internet unter www.deutschmann.de bezogen werden).

21.3 Montage

Die Baugruppe UNIGATE® CL-CO hat die Schutzart IP20 und ist somit für den Schaltschrankeinsatz geeignet. Das Gerät ist für das Aufschnappen auf eine 35 mm Hutprofilschiene ausgelegt.

21.4 Maßzeichnung UNIGATE® CL-CANopen®



21.5 Inbetriebnahme

Um ein ordnungsgemäßes Arbeiten der Baugruppe zu gewährleisten, müssen Sie folgende Schritte bei der Inbetriebnahme unbedingt durchführen:

21.6 CANopen®-Adresse und Baudrate einstellen

Stellen Sie an der Feldbusseite der Baugruppe an dem DIP-Switch die CANopen®-Node-ID und die Baudrate ein (siehe auch Kapitel 16.4.4).



Achtung:

Die eingestellte CANopen®-Adresse muss mit der projektierten Adresse übereinstimmen!

**Alle Teilnehmer im CANopen® müssen die gleiche Baudrate verwenden!
Diese Werte werden nur beim Einschalten des Gateways eingelesen!**

21.7 CANopen®-Anschluss

Verbinden Sie das Gerät mit dem CANopen® an der Schnittstelle mit der Bezeichnung "CANopen®".

21.8 Anschluss des Prozessgerätes.

Zur Inbetriebnahme des Prozessgerätes lesen Sie bitte auch dessen Betriebsanleitung.

21.9 Versorgungsspannung anschließen

Schließen Sie bitte 10..33 V Gleichspannung an die dafür vorgesehenen Klemmen an.

21.10 Schirmanschluss

Erden Sie die Hutschiene, auf der die Baugruppe aufgeschnappt wurde.

21.11 Projektierung

Verwenden Sie zum Projektieren ein beliebiges Projektierungstool.

Falls die benötigte EDS-Datei nicht mit Ihrem Projektierungstool ausgeliefert wurde, kann eine Muster-Datei aus dem Internet (www.deutschmann.de) bezogen werden.

22 Service

Sollten einmal Fragen auftreten, die in diesem Handbuch nicht beschrieben sind, finden Sie im

- FAQ/Wiki Bereich unserer Homepage www.deutschmann.de oder www.wiki.deutschmann.de weiterführende Informationen.

Falls dennoch Fragen unbeantwortet bleiben sollten wenden Sie sich direkt an uns.

Bitte halten Sie für Ihren Anruf folgende Angaben bereit:

- Gerätebezeichnung
- Seriennummer (S/N)
- Artikel-Nummer
- Fehlernummer und Fehlerbeschreibung

Ihre Anfragen werden im Support Center aufgenommen und schnellstmöglich von unserem Support Team bearbeitet. (In der Regel innerhalb von einem Arbeitstag, selten länger als drei Arbeitstage.)

Der technische Support ist erreichbar von Montag bis Donnerstag von 8.00 bis 12.00 und von 13.00 bis 16.00, Freitag von 8.00 bis 12.00 (MEZ).

Deutschmann Automation GmbH & Co. KG
Carl-Zeiss-Straße 8
D-65520 Bad-Camberg

Zentrale und Verkauf	+49 6434 9433-0
Technischer Support	+49 6434 9433-33

Fax Verkauf	+49 6434 9433-40
Fax Technischer Support	+49 6434 9433-44

Email Technischer Support support@deutschmann.de

22.1 Einsendung eines Gerätes

Bei der Einsendung eines Gerätes benötigen wir eine möglichst umfassende Fehlerbeschreibung. Insbesondere benötigen wir die nachfolgenden Angaben:

- Welche Fehlernummer wurde angezeigt
- Wie groß ist die Versorgungsspannung ($\pm 0,5V$) mit angeschlossenem Gateway
- Was waren die letzten Aktivitäten am Gerät (Programmierung, Fehler beim Einschalten, ...)

Je genauer Ihre Angaben und Fehlerbeschreibung, umso exakter können wir die möglichen Ursachen prüfen.

22.2 Download von PC-Software

Von unserem Internet-Server können Sie kostenlos aktuelle Informationen und Software laden.

<http://www.deutschmann.de>

23 Anhang

23.1 Erläuterung der Abkürzungen

Allgemein

CL	= Produktgruppe CL (Compact Line)
CM	= Produktgruppe CM (CANopen Line)
CX	= Produktgruppe CX
EL	= Produktgruppe EL (Ethernet Line)
FC	= Produktgruppe FC (Fast Connect)
GT	= Galvanische Trennung RS-Seite
GY	= Gehäusefarbe grau
MB	= Produktgruppe MB
RS	= Produktgruppe RS
SC	= Produktgruppe SC (Script)
232/485	= Schnittstelle RS232 und RS485 umschaltbar
232/422	= Schnittstelle RS232 und RS422 umschaltbar
DB	= zusätzlich eine RS232 DEBUG-Schnittstelle
D9	= Anschluss der RS über 9pol. D-SUB statt 5pol. Schraub-Steckverbinder
PL	= Nur Platine ohne DIN-Schienenmodul und ohne Gehäusedeckel
PD	= Nur Platine ohne DIN-Schienenmodul mit Gehäusedeckel
AG	= Gateway montiert im Aludruckgussgehäuse
EG	= Gateway montiert im Edelstahlgehäuse
IC	= Produktgruppe IC (IC-Bauform DIL32)
IC2	= Produktgruppe IC2 (IC-Bauform DIL32)
IO8	= Option I/O8
16	= Scriptspeicher auf 16KB erweitert
5V	= Betriebsspannung 5V
3,3V	= Betriebsspannung 3,3V

Feldbus

CO	= CANopen
C4	= CANopen V4
C4X	= CANopen V4-Variante X (siehe Vergleichstabelle UNIGATE® IC beim jeweiligen Produkt)
DN	= DeviceNet
EC	= EtherCAT
EI	= EtherNet/IP
FE	= Ethernet 10/100 MBit/s
FEX	= Ethernet 10/100 MBit/s-Variante X (siehe Vergleichstabelle UNIGATE® IC beim jeweiligen Produkt)
IB	= Interbus
IBL	= Interbus
LN62	= LONWorks62
LN512	= LONWorks512
ModTCP	= ModbusTCP
MPI	= Siemens MPI®
PL	= Powerlink
PN	= Profinet-IO

PBDP = ProfibusDP
PBDPL = ProfibusDP-Variante L (siehe Vergleichstabelle UNIGATE® IC beim jeweiligen Produkt)
PBDPX = ProfibusDP-Variante X (siehe Vergleichstabelle UNIGATE® IC beim jeweiligen Produkt)
PBDPV0 = ProfibusDPV0
PBDPV1 = ProfibusDPV1
RS = Serial RS232/485/422

23.2 Hexadezimal-Tabelle

Hex	Dezimal	Binär
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111