



**Deutschmann**

*your ticket to all buses*

**Bedienerhandbuch  
Universelles Feldbus-Gateway  
UNIGATE® IC - PROFIBUS**



**Deutschmann Automation GmbH & Co. KG**  
**[www.deutschmann.de](http://www.deutschmann.de) | [wiki.deutschmann.de](http://wiki.deutschmann.de)**



<b>1</b>	<b>Allgemeine Einführung</b>	<b>9</b>
<b>2</b>	<b>Das UNIGATE® IC</b>	<b>10</b>
2.1	Technische Einführung	10
2.2	Verfügbarkeit	10
2.3	Firmware	10
2.4	Die serielle Standardschnittstelle	10
2.5	Die synchrone serielle Schnittstelle	10
2.6	Die Debug Schnittstelle	10
2.7	UNIGATE® IC Hardwareübersicht	11
<b>3</b>	<b>Hardware Design</b>	<b>12</b>
3.1	Anschlüsse	12
3.2	Pinout	12
3.2.1	-Boot enable	13
3.2.2	Load out (SPI-Master: SS0-)	13
3.2.3	Data out (SPI-Master: SS1-)	13
3.2.4	Data In (SPI: MISO)	13
3.2.5	Load In (SPI: MOSI)	13
3.2.6	Clock (SPI: SCK)	14
3.2.7	-Reset In	14
3.2.8	LED-PB	14
3.2.9	-Config Mode	14
3.2.10	DbgTX, DbgRx	14
3.2.11	TE	14
3.2.12	TX, RX	14
3.3	Software	14
3.4	Prinzipielle Vorgehensweise	15
3.5	Anschlussbeispiele	15
3.6	Layout-Beispiele	18
3.7	Betrieb mit externem PROFIBUS-Treiber	21
3.8	Termination (PROFIBUS)	21
3.9	Handling (Befestigung des UNIGATE® IC auf der Trägerplatine)	21
3.10	Unterschiede zwischen den verschiedenen Ausführungen	22
3.10.1	Standard Ausführung	22
3.10.2	Ausführung "ohne 485 (PROFIBUS)-Treiber"	22
3.10.3	Ausführung "LWL"	22
<b>4</b>	<b>Die serielle Schnittstelle</b>	<b>23</b>
4.1	Übersicht	23
4.2	Initialisierung der seriellen Schnittstelle	23
4.3	Benutzung der seriellen Schnittstelle	23
4.4	Weitere Betriebsmodi	23

<b>5</b>	<b>Synchrone serielle Schnittstelle</b>	<b>24</b>
5.1	Schieberegisterbetrieb	24
5.1.1	Script-Beispiel	24
5.2	SPI-Betrieb	25
5.2.1	Script-Beispiel	25
<b>6</b>	<b>Die Debug Schnittstelle</b>	<b>26</b>
6.1	Übersicht über die Debug Schnittstelle	26
6.2	Start im Debugmode	26
6.3	Kommunikationsparameter für die Debugschnittstelle	26
6.4	Möglichkeiten mit der Debug Schnittstelle	26
6.5	Befehle der Debug Schnittstelle	26
<b>7</b>	<b>Script und Konfiguration</b>	<b>27</b>
7.1	Übersicht	27
7.2	Der Konfigurationsmodus	27
7.3	Script updaten	27
7.4	Konfiguration des UNIGATE® IC	29
7.4.1	PROFIBUS	29
7.4.2	RS232/RS485/RS422	30
<b>8</b>	<b>Erstellung eines Scripts</b>	<b>31</b>
8.1	Was ist ein Script	31
8.2	Speichereffizienz der Programme	31
8.3	Was kann man mit einem Script Gerät machen	31
8.4	Unabhängigkeit von Bussen	31
8.5	Weitere Einstellungen am IC Gateway	31
8.6	Die Benutzung des Protocol Developer	32
8.7	Genauigkeiten der Baudraten bei UNIGATE® IC	32
8.8	Scriptarbeitungszeiten	33
<b>9</b>	<b>PROFIBUS DP</b>	<b>34</b>
9.1	Beschreibung der DPV1- / DPV2-Funktionen	34
9.1.1	DPV1	34
9.1.2	DPV2	34
9.2	Einstellen der PROFIBUS-Adresse	35
<b>10</b>	<b>Fehlerbehandlung beim UNIGATE® IC</b>	<b>37</b>
<b>11</b>	<b>Firmware Update</b>	<b>38</b>
11.1	Übersicht	38
11.2	Einstellen des Firmware-Update-Modus	38
11.2.1	Einstellung per Hardware	38
11.2.2	Einstellung per Software	38
11.3	Durchführung des Firmware-Updates	38

11.4 Sicherheitshinweis . . . . . 38

11.5 Betriebsmodi des IC . . . . . 38

**12 Technische Daten . . . . . 39**

12.1 Mechanik des UNIGATE® IC . . . . . 39

12.1.1 Abmessungen UNIGATE® IC . . . . . 39

12.2 Technische Daten UNIGATE® IC-PROFIBUS . . . . . 41

12.2.1 Leistungsmerkmale der verschiedenen Versionen . . . . . 41

**13 Zubehör . . . . . 42**

13.1 FirmwareDownloadTool (FDT) . . . . . 42

13.2 Protocol Developer . . . . . 42

13.3 Developer Kit UNIGATE® IC-AB . . . . . 42

13.3.1 Developer Board UNIGATE® IC-AB . . . . . 42

13.3.2 Schnellstart . . . . . 43

**14 Anhang . . . . . 44**

14.1 Erläuterungen der Abkürzungen . . . . . 44

**15 Darstellung der Daten in PROFIBUS DP . . . . . 46**

15.1 Konfiguriertelegramm . . . . . 46

15.2 Konfiguriertelegramm . . . . . 46

15.3 Diagnose . . . . . 47

15.3.1 Diagnose in DPV1 . . . . . 49

15.4 Datenaustausch . . . . . 49

**16 Service . . . . . 50**

16.1 Einsendung eines Gerätes . . . . . 50

16.2 Download von PC-Software . . . . . 50



### 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 Allgemeine Einführung

In der Vergangenheit erforderte die Integration eines Feldbusanschlusses einen enormen Aufwand der Entwicklungsingenieure. Durch die vielfältigen Kommunikationstechniken ist es nicht damit getan, die richtige Kombination von Kommunikationshardware zusammenzustellen; die verschiedenen Busse erfordern durch ihre Normen und Grundlagen auch entsprechendes Können der Ingenieure.

Im Falle des Deutschmann UNIGATE® IC trifft das nicht mehr zu. Alle digitalen Funktionen, Software Stack und Treiber sowie Optokoppler entsprechend der Norm sind alle auf einem IC integriert. Zusätzlich zur Verringerung der benötigten Größe können auch verschiedene Feldbusse einfach integriert werden.

**Durch die flexible Firmware des UNIGATE® IC sind auf der Kundenseite keinerlei Änderungen der Software notwendig!**

Deutschmann Automation ist seit 1997 im Bereich der Feldbusgateways erfahren; aus dieser enormen Erfahrung heraus resultiert das UNIGATE® IC als eine konsequente Fortführung dieser erfolgreichen Produktlinie.

### *Terminologie*

Im gesamten Dokument und in allen Teilen der zu verwendenden Software werden die Begriffe Input und Output benutzt. Input und Output sind mehrdeutig; jeweils abhängig von der Sichtweise. Wir sehen den Feldbus als zentrale Schnittstelle und als integralen Bestandteil Ihres Gerätes; daher bezeichnen wir an allen Stellen - unabhängig vom verwendeten Bus - die Daten aus der Sicht des Slave, also Input Daten als Daten vom Master zum Slave.

### *Darstellung von Zahlen*

Zahlen im dezimalen Format werden immer sowohl ohne Prefix als auch ohne Suffix dargestellt. Hexadezimale Zahlen werden immer durch das Prefix 0x gekennzeichnet.

## 2 Das UNIGATE® IC

### 2.1 Technische Einführung

Das Deutschmann Automation UNIGATE® IC beinhaltet alle für die Kommunikation in einem Feldbus nötigen Komponenten in einer einzelnen Baugruppe, das bedeutet, dass ein Entwickler sich nicht mehr um diese Details zu kümmern braucht; es ist lediglich ein Hardware Redesign notwendig, um das UNIGATE® IC und die nötigen Steckverbinder zu integrieren.

### 2.2 Verfügbarkeit

Das Modul ist als PROFIBUS DP verfügbar. Weitere Feldbusse sind in Arbeit bzw. Planung; diese werden sich nur in den Anschlüssen des Busses unterscheiden. Die allgemeinen Pins 1 - 9 sowie 24 und 26 - 32 werden ihre Bedeutung auch für weitere Feldbusimplementierungen unverändert behalten. Eine aktuelle Liste für alle UNIGATE® ICs finden Sie unter:  
<http://www.deutschmann.de>.

### 2.3 Firmware

UNIGATE® IC wird über Scripte programmiert. Im Prinzip ist jedes Script, das für ein UNIGATE® SC entwickelt wurde, auch auf dem UNIGATE® IC lauffähig.

### 2.4 Die serielle Standardschnittstelle

Intelligente Geräte, die bereits einen Microcontroller oder Microprozessor haben, verfügen im allgemeinen auch über eine serielle, asynchrone Schnittstelle mit TTL Pegeln. Diese wird direkt mit der TTL Schnittstelle des UNIGATE® ICs verbunden. Weitere Informationen zu dieser seriellen Schnittstelle finden Sie im Kapitel 4, auf Seite 23.

### 2.5 Die synchrone serielle Schnittstelle

Zusätzlich zu der Standardschnittstelle gibt es auch die Möglichkeit der synchronen Datenein- und Ausgabe. So können zum Beispiel digitale IOs über Schieberegisterbausteine oder auch analoge IOs über DA-Wandler mit seriellem Ein-/Ausgang angeschlossen werden. Für synchrone IOs können maximal 256 Signale benutzt werden (256 Bit). Schaltungsbeispiele finden Sie im Kapitel 3.5, auf Seite 15 und Softwarebeispiele finden Sie im Kapitel 5. Diese Schnittstelle kann ebenso dazu genutzt werden Bausteine bzw. Geräte mit SPI-Schnittstelle anzubinden. Es ist auch möglich z. B. digitale oder analoge E/A-Module zu bauen, ohne dass das Kundengerät über einen eigenen Controller verfügt; das Feldbus IC ist auch ohne diesen autonom lauffähig.

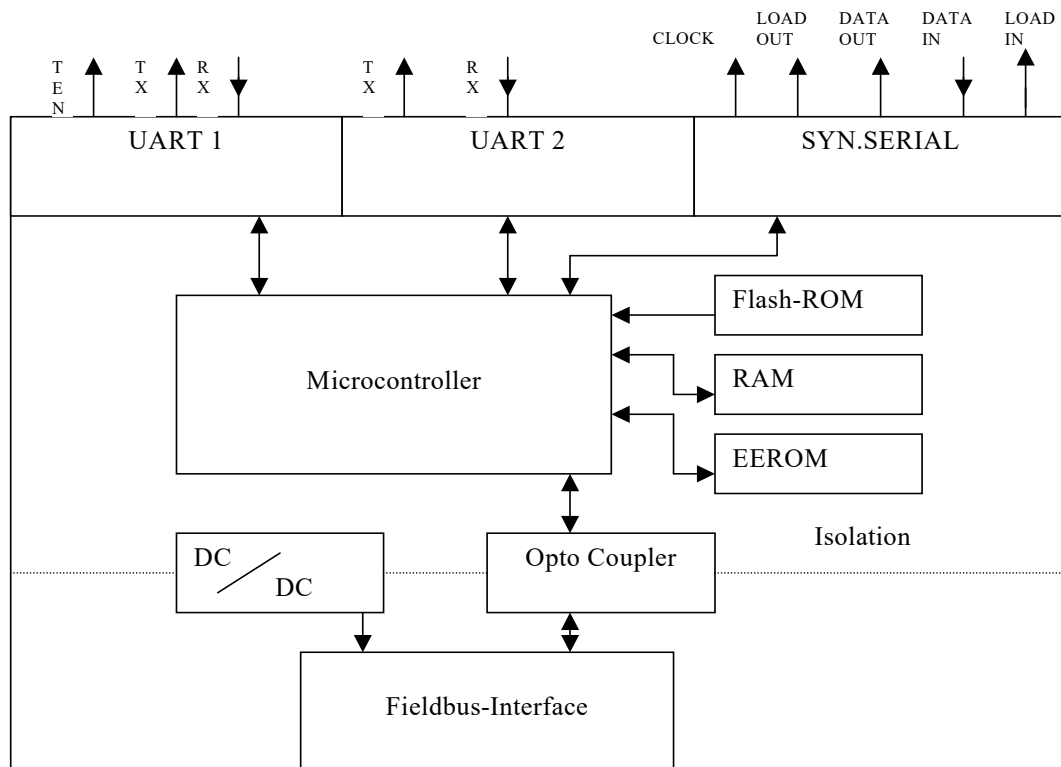
### 2.6 Die Debug Schnittstelle

Das UNIGATE® IC verfügt über eine Debug Schnittstelle, die es erlaubt, ein Script schrittweise abzuarbeiten und auch Daten zu beobachten oder zu manipulieren. Dies ist für die Entwicklung eines Scripts unerlässlich. Normalerweise wird ein Script mit der Software Protocol Developer entwickelt. Details hierzu finden Sie im Handbuch Protocol Developer.

*Alle Schnittstellen können unabhängig voneinander zur selben Zeit benutzt werden.*

## 2.7 UNIGATE® IC Hardwareübersicht

Die Hardware des UNIGATE® IC besteht aus einigen wenigen Standardbauteilen. Die folgende Grafik zeigt den funktionellen Aufbau des IC.



### 3 Hardware Design

In diesem Kapitel werden die grundlegenden Hinweise gegeben, die nötig sind, um das UNIGATE® IC in eigene Hardware Designs zu übernehmen. Es werden im folgenden alle Anschlüsse des UNIGATE® ICs ausführlich beschrieben.

#### 3.1 Anschlüsse

Das UNIGATE® IC hat 32 Pins im Layout wie ein DIL 32 Baustein. Pins 10 - 12 sowie 21 – 23 sind wegen der elektrischen Isolation nicht belegt. Die genauen mechanischen Abmessungen können Kapitel 12, auf Seite 39 entnommen werden.



Für ALLE 32 Pins sind im Layout Bohrungen vorzusehen.

#### 3.2 Pinout

Pin	Technische Spezifikationen	Name	Beschreibung	Anmerkung
1	$5V \pm 5\% < I_{typ} 170mA$ $3,3V \pm 5\% < I_{typ} 170mA$	Vcc	+ 5V Spannungsversorgung + 3,3V Spannungsversorgung (optional bei PBL)	siehe Kapitel Kapitel 12.2.1
2	IN <sub>Logic</sub>	-BE	Boot enable	
3	OUT <sub>Driver</sub>	Load out (SS0-)	Übernahmesignal der Ausgangsdaten der synchronen seriellen Schnittstelle	
4	OUT <sub>Driver</sub>	Data out (SS1-)	Ausgangsdaten für synchrone, serielle Schnittstelle	
5	IN <sub>Logic</sub>	Data in (MISO)	Eingangsdaten der synchronen, seriellen Schnittstelle	Interner Pull-Up
6	OUT <sub>Logic</sub>	Load in (MOSI)	Übernahmesignal der Eingangsdaten der synchronen seriellen Schnittstelle	
7	OUT <sub>Driver</sub>	Clock (SCK)	Taktsignal für synchrone, serielle Schnittstelle	
8	IN <sub>Reset</sub>	-Reset In	Reset-Eingang des IC	Interner Pull-Up
9	connected to pin 1	Vcc	+ 5 V Spannungsversorgung	
10-12	nc	nc	kein Pin vorhanden	
13*	according to norm	PB-A TxPB <sup>1)</sup>	PROFIBUS-Signal nach Norm	galvanisch getrennt Isolationsspannung 1000 Vrms
14*	according to norm	PB-B RxPB <sup>1)</sup>	PROFIBUS-Signal nach Norm	galvanisch getrennt Isolationsspannung 1000 Vrms
15*	according to norm	PB-RTS	PROFIBUS-Signal nach Norm	galvanisch getrennt Isolationsspannung 1000 Vrms
16-18	according to norm	nc	nicht belegt	
19*	according to norm	PB-GND	PROFIBUS-Signal nach Norm	galvanisch getrennt Isolationsspannung 1000 Vrms
20*	according to normX	PB-5V	PROFIBUS-Signal nach Norm	galvanisch getrennt Isolationsspannung 1000 Vrms
21-23	nc	nc	kein Pin vorhanden	
24	connected to pin 32	GND*	Ground Spannungsversorgung des IC	
25	OUT <sub>Tristate</sub>	LED-PB	Bus Error LED des PROFIBUS	8 mA, interner Vorwiderstand $5V = 1,5 k\Omega$ , $3,3V = 470R$
26	IN <sub>Logic</sub>	-Config Mode	Signal zum Starten im Konfigurationsmodus	Interner Pull-Up
27	OUT <sub>Logic</sub>	DbgTX	serial Debug TX	
28	IN <sub>Logic</sub>	DbgRX	serial Debug RX	Interner Pull-Up
29	IN <sub>Logic</sub>	RX	serielle Daten RX	Interner Pull-Up
30	OUT <sub>Logic</sub>	TX	serielle Daten TX	

31	OUT <sub>Logic</sub>	TE	Transmit enable	
32	GND	GND	Ground Spannungsversorgung des IC	

Die PROFIBUS Signale sind galvanisch getrennt. Die Isolationsspannung beträgt 1000 Vrms.

<sup>1)</sup> Ausführung ohne 485 (PROFIBUS) - Treiber (siehe Kapitel 3.7 und Kapitel 3.10.2).

\*) In der Ausführung "LWL" (ohne galvanische Trennung auf der PROFIBUS-Seite) liegen an den Pins 13, 14, 15 direkt die TTL-Signale. Die Pins 19 und 20 sind offen, da bei LWL keine galvanisch getrennte PROFIBUS-Spannung benötigt wird bzw. diese dann selbst erzeugt wird (siehe Kapitel 3.10.3).

<b>5V±5%, Typ &lt; 170mA</b>	V <sub>IL</sub>	V <sub>IH</sub>
IN <sub>Reset</sub>	< 0,7V	>3,85V
IN <sub>Logic</sub>	< 0,8V / 0,5mA	>1,95V / 10µA
	V <sub>OL</sub>	V <sub>OH</sub>
OUT <sub>Logic</sub>	< 0,6V / 1mA	>3,8V / 0,1mA
OUT <sub>Driver</sub>	< 0,33V / 4mA	>3,8V / 4mA
<b>3,3V±5%, Typ &lt; 170mA</b>	V <sub>IL</sub>	V <sub>IH</sub>
IN <sub>Reset</sub>	< 0,7V	>2,4V
IN <sub>Logic</sub>	< 0,5V / 0,4mA	>1,5V / 10µA
	V <sub>OL</sub>	V <sub>OH</sub>
OUT <sub>Logic</sub>	< 0,6V / 1mA	>2,3V / 0,1mA
OUT <sub>Driver</sub>	< 0,5V / 4mA	>2,5V / 4mA

### 3.2.1 -Boot enable

Mit dem Pegel GND während dem Power Up Vorgang wird das IC im Firmware Update Mode gestartet. Siehe auch Kapitel 11, auf Seite 38.

### 3.2.2 Load out (SPI-Master: SS0-)

Übernahmesignal für die synchrone serielle Schnittstelle. Mit der positiven Flanke an diesem Ausgang werden Daten von den angeschlossenen Schieberegistern an die physikalischen Ausgänge übernommen.

Im SPI-Modus dient dieser Pin als Low-aktives Slave-Select-Signal.

### 3.2.3 Data out (SPI-Master: SS1-)

Auf dieser Leitung werden die Daten auf der synchronen seriellen Schnittstelle ausgegeben. Das höchstwertige Bit der Daten wird zuerst ausgegeben.

Im SPI-Modus dient dieser Pin als Low-aktives Slave-Select-Signal.

### 3.2.4 Data In (SPI: MISO)

Über dieses Signal werden die Daten auf der synchronen seriellen Schnittstelle eingelesen. Das höchstwertige Bit der Daten wird zuerst erwartet.

Im SPI-Modus dient dieser Pin zur Datenübertragung vom Slave zum Master.

### 3.2.5 Load In (SPI: MOSI)

Dieser Pin ist das Übernahmesignal für die Eingangsdaten der synchronen seriellen Schnittstelle.

Im SPI-Modus dient dieser PIN zur Datenübertragung vom Master zum Slave.

### 3.2.6 Clock (SPI: SCK)

Dieses Signal ist die Taktleitung für die synchrone serielle Schnittstelle. Dies Signal gilt gleichermaßen für die Datenein- und Ausgabe.

### 3.2.7 -Reset In

Mit diesem Eingang kann ein Reset des UNIGATE® ausgelöst werden. Es ist ein interner Pull-Up-Widerstand vorhanden, sodass der Eingang unbeschaltet bleiben kann. Die Pulsdauer sollte mindestens 10 ms betragen.

### 3.2.8 LED-PB

An diese Leitung kann eine rote LED angeschlossen werden. Diese wird vom PROFIBUS ASIC gesteuert und erlischt im Zustand „Data Exchange“. (siehe Kapitel 3.6)

Pin	Zustände	Beschreibung
25	rot leuchtend	Keine Verbindung zum PROFIBUS Master
	AUS	Verbunden mit PROFIBUS Master

### 3.2.9 -Config Mode

Ist der Pin auf dem Pegel GND startet das IC im Konfigurationsmode.

### 3.2.10 DbgTX, DbgRx

Dies sind sowohl Sendeleitung als auch Empfangsleitung der Debug Schnittstelle des IC. Zur Funktionsbeschreibung der Debug Schnittstelle siehe Kapitel 6, auf Seite 26.

### 3.2.11 TE

Das Transmit Enable Signal ermöglicht den Anschluss von RS485 Treibern an der seriellen Schnittstelle des IC. Das Signal wird immer dann auf High geschaltet, wenn das IC über die Leitung TX sendet.

### 3.2.12 TX, RX

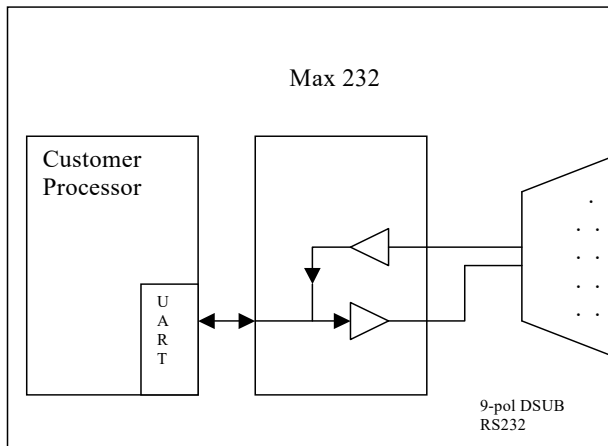
Sende und Empfangsleitungen der seriellen Schnittstelle. diese Schnittstelle ist entsprechend der Beschreibung in Kapitel 4, auf Seite 23 frei programmierbar.

## 3.3 Software

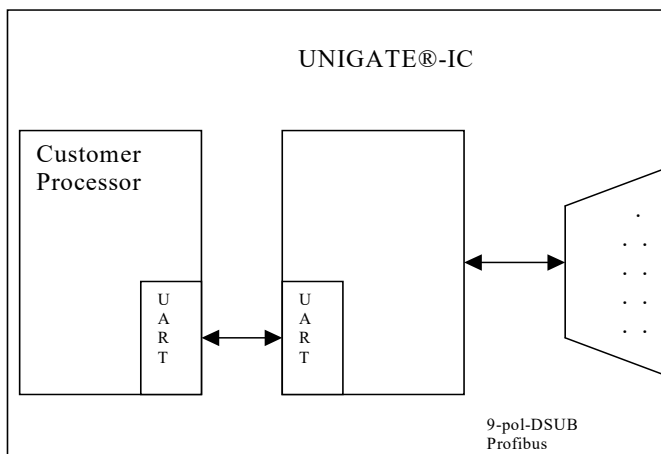
Die Software führt Scriptbefehle aus, diese wiederum steuern die Hardware des IC und wickeln per Software ihr komplettes Protokoll ab. Das Script selbst kann durch die Firma Deutschmann oder mit der Software Protocol Developer von Ihnen erstellt werden. Für eine detaillierte Beschreibung der Scriptbefehle des Protocol Developer siehe Benutzerhandbuch Protocol Developer und die Online-Dokumentation zu den Scriptbefehlen.

### 3.4 Prinzipielle Vorgehensweise

Theoretisch ist es ausreichend, den in Ihrer Anwendung enthaltenen RS232-Treiber durch das UNIGATE® IC zu ersetzen.



Ihr Gerät, das im wesentlichen wie oben aufgebaut sein dürfte wird nun modifiziert, so dass an der 9-pol Buchse der PROFIBUS zur Verfügung steht. Um die Belegung normgerecht zu halten, ist allerdings ein Hardware-Redesign nötig.



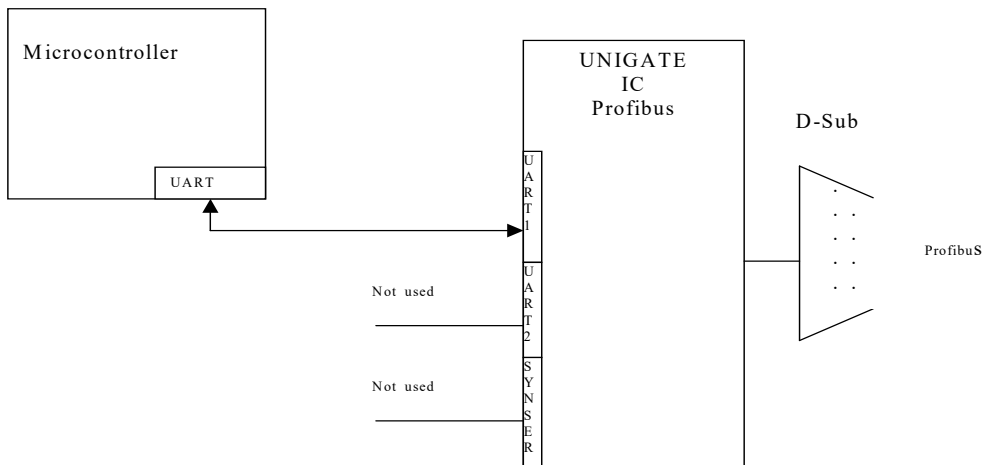
Nach dem Austausch des RS232 Treibers durch das UNIGATE® IC steht an der 9-pol D-SUB-Buchse der PROFIBUS zur Verfügung.

Deutschmann bietet auch eine entsprechende Adapterplatine an; mit ihr können bestehende Geräte ohne Redesign umgerüstet werden; siehe hierzu Kapitel 13, auf Seite 42.

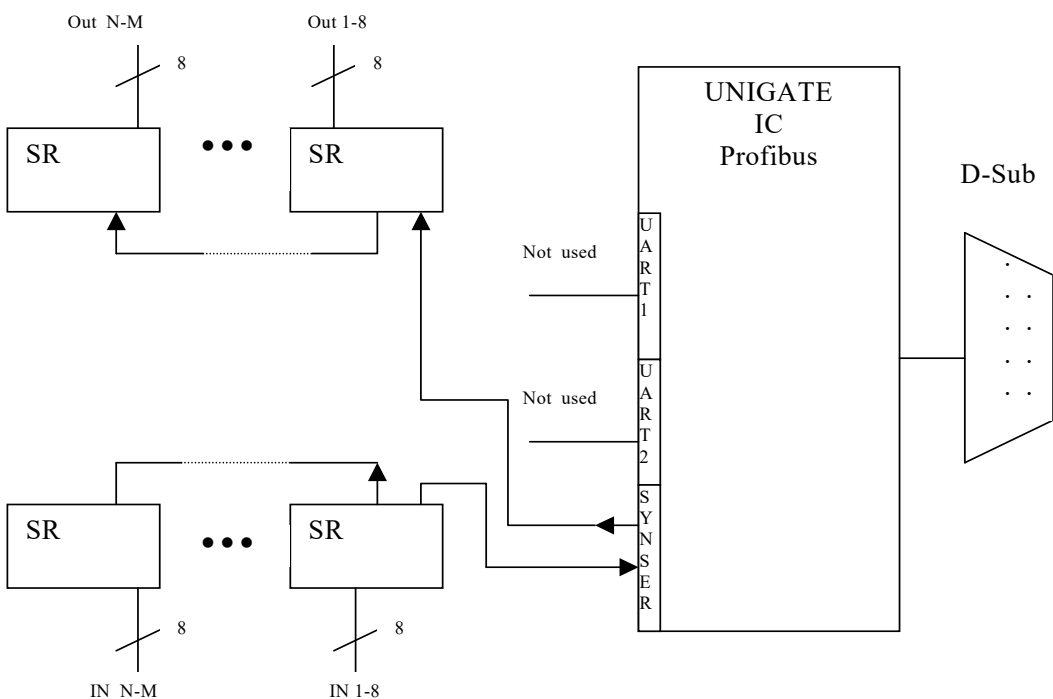
### 3.5 Anschlussbeispiele

Hier finden Sie einige Hinweise, die Ihnen für ein Redesign helfen. Im folgenden sind einige Varianten aufgeführt, die Ihnen die Entscheidung etwas leichter machen sollen.

## Variante 1: Einsatz als reines Koppelmodul für den Bus



Das UNIGATE<sup>®</sup> IC wickelt selbständig die Kommunikation mit dem Kundengerät über die TTL-Schnittstelle ab.

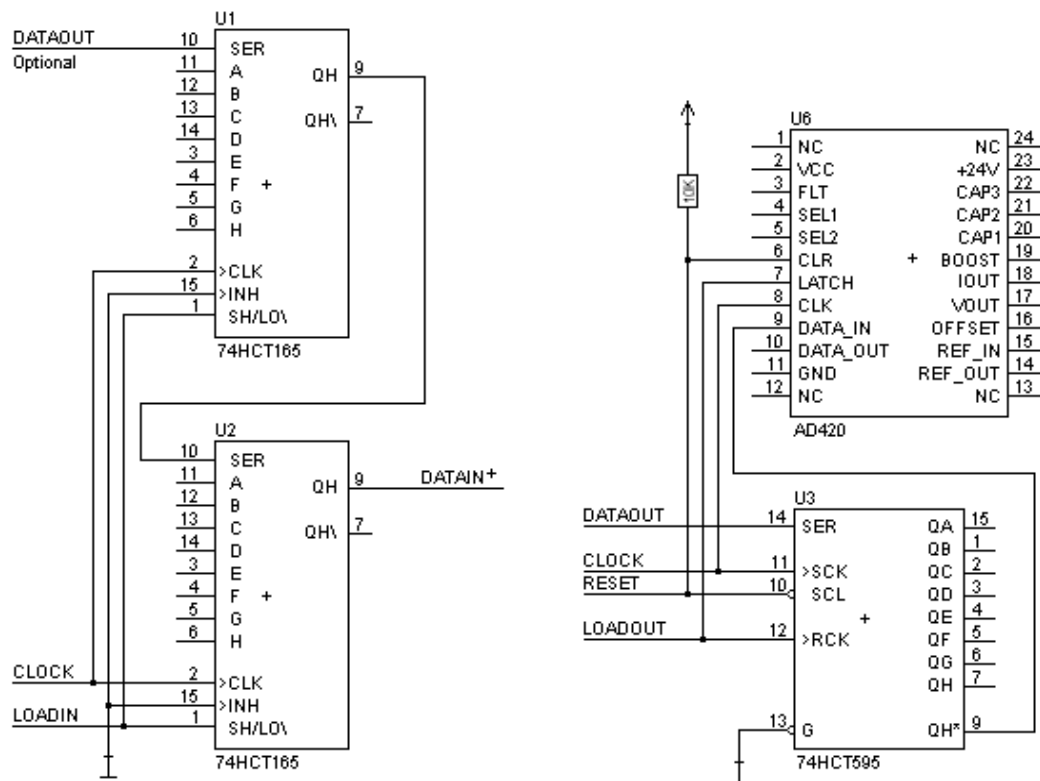
Variante 2: Einsatz des UNIGATE<sup>®</sup> IC für digitale oder analoge E/A-Module

Hier wird lediglich die synchrone serielle Schnittstelle benutzt, die asynchrone serielle Schnittstelle ist hier im Prinzip ohne Bedeutung. Falls sie das Script jedoch in Ihrer fertigen Anwendung programmieren möchten, empfiehlt sich der Einsatz einer Pfostenleiste für die asynchrone Schnittstelle, Sie können damit die ISP Programmierung vornehmen.

**Es ist für diese Betriebsart kein zusätzlicher Controller auf Ihrer Applikation notwendig!**



Das folgende Schaltbild dient als Beispiel wie Schieberegister Bausteine an das IC angekoppelt werden können.

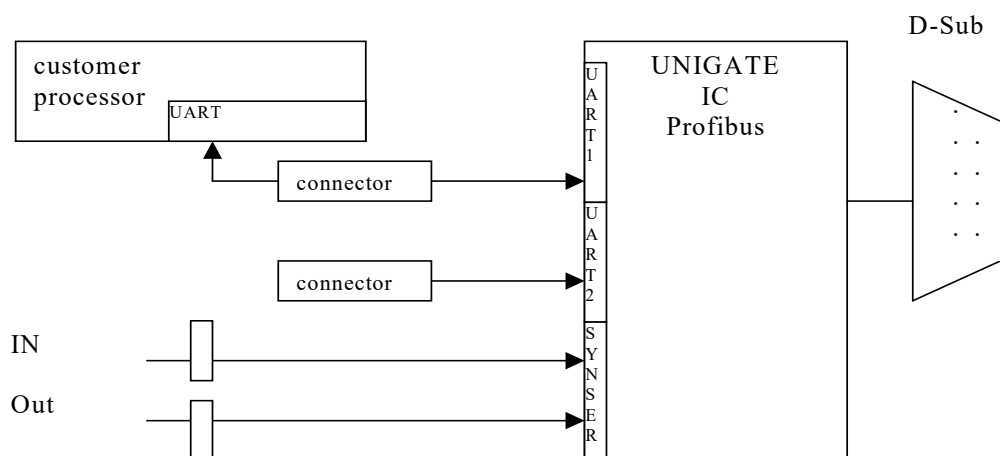


### Variante 3: Beispiel für digitale I/Os

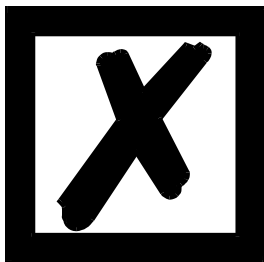
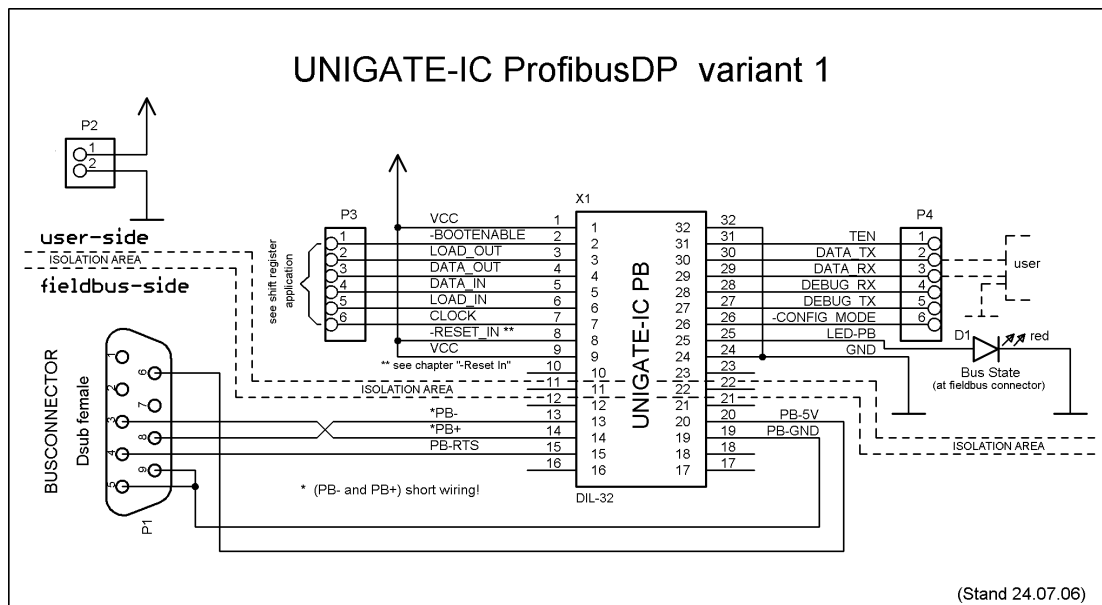
Die serielle synchrone als auch die asynchrone Schnittstelle kann vom UNIGATE<sup>®</sup> IC zur selben Zeit bedient werden. Hieraus ergibt sich die Möglichkeit, eine bestehende Applikation um zusätzliche digitale oder analoge E/As zu erweitern.

Im Kapitel 5.1.1 finden Sie ein Beispiel für ein Script, das diese E/As bedient.

Für alle Varianten gilt: Eine vorgesehene Steckverbindung der seriellen Schnittstelle in der Anwendung schafft die Möglichkeit, über eine externe Verbindung ein Update der Firmware oder der Software durchzuführen.



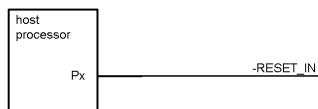
### 3.6 Layout-Beispiele



**Die Verbindung der beiden PROFIBUS-Leitungen (PB+ und PB-) sind so kurz wie möglich auszuführen, um Reflexionen und damit Störungen auf der PROFIBUS-Leitung zu vermeiden!**

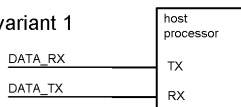
**Sollten diese Stichleitungen (PB+ und PB-) zu lang sein, kann es bei höheren Busgeschwindigkeiten ( $\geq 1,5$  Mbaud) zu Kommunikationsproblemen kommen. Gemäß der PROFIBUS Prüfvorschrift beträgt die maximal zulässige Gesamtlänge aller Stichleitungen 80cm bei 12Mbaud. Über ein komplettes Bussegment von 32 Knoten entspricht dies 25mm pro Knoten.**

#### description for Reset-Input

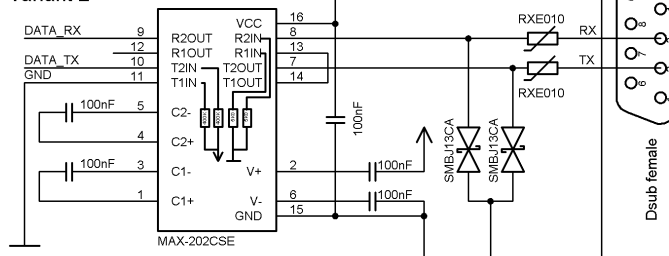


#### user application

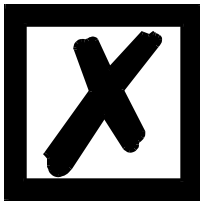
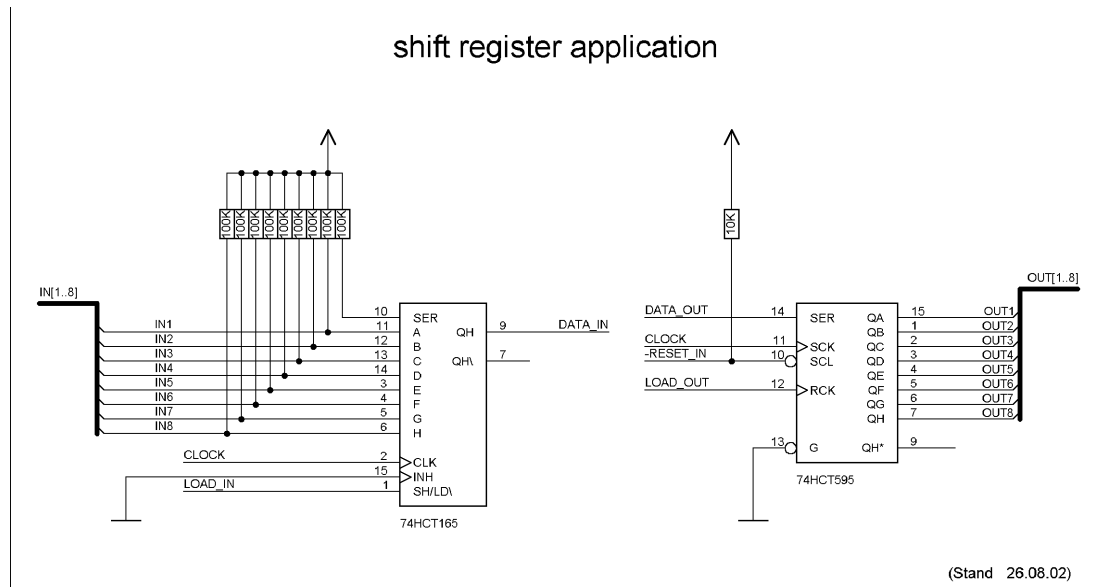
##### variant 1



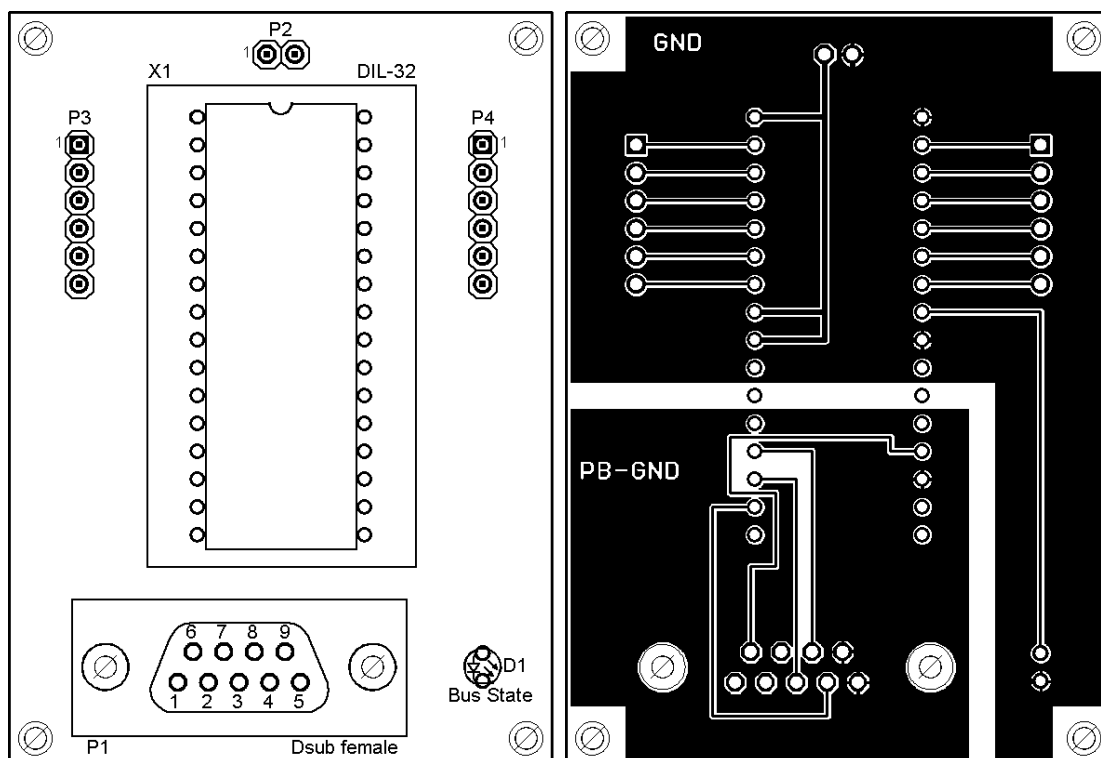
##### variant 2



## shift register application



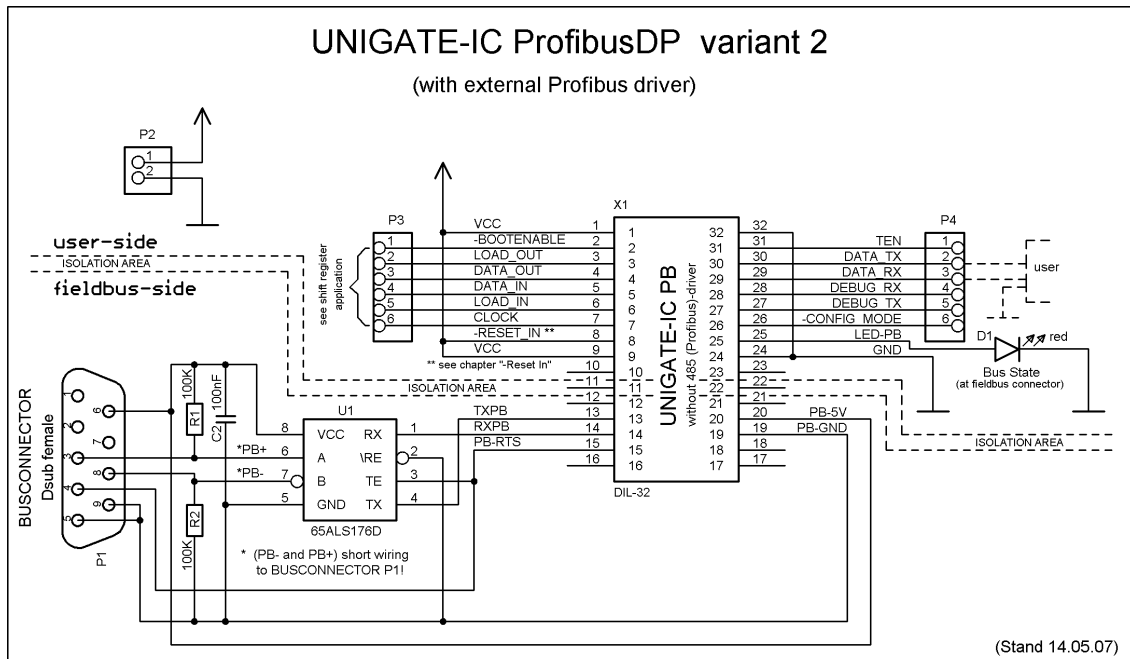
Beim in diesem Beispiel eingesetzten 74HCT595 ist der Einschaltzustand undefiniert, dafür können aber über den OutEnable-Pin 13 die Ausgänge in den Tri-State Zustand versetzt werden. Ist es in einer Applikation wichtiger einen definierten Einschaltzustand zu bekommen, und der OutEnable-Pin nicht notwendig, kann hier z.B. der 74HCT594 verwendet werden.



### 3.7 Betrieb mit externem PROFIBUS-Treiber



**Hinweis:** Nur in Verbindung mit IC-Ausführung "ohne 485 (PROFIBUS) - Treiber".



### 3.8 Termination (PROFIBUS)

Es ist generell zu empfehlen, PROFIBUS-Stecker mit integriertem Busabschluss zu verwenden. Ist dies nicht möglich, kann ein schaltbarer Abschlusswiderstand (220 R) auf dem Trägerboard integriert werden.

In diesem Fall muss außerdem ein Pull-Down-Widerstand (390 R) nach PB-GND und ein Pull-Up-Widerstand (390 R) nach PB-5V integriert werden!

### 3.9 Handling (Befestigung des UNIGATE<sup>®</sup> IC auf der Trägerplatine)

Generell gibt es zur Montage des UNIGATE<sup>®</sup> IC's auf der Trägerplatine folgende Möglichkeiten, die in Abhängigkeit von der Applikation und der zu erwartenden Schock- und Vibrationsverhältnisse ausgewählt werden können:

- Aufstecken auf einen Sockel im Trägerboard. Eventuell das UNIGATE<sup>®</sup> IC im Sockel an 2 oder 4 Pins festlöten. Das IC lässt sich in der Regel dann nach Entfernen der Lötunkte problemlos wieder herausziehen.
- Im Layout neben dem Sockel zwei Löcher vorsehen. Nach dem Aufstecken des UNIGATE<sup>®</sup> IC's auf den Sockel einen isolierten Draht über das IC ziehen und in den vorgesehenen Löchern der Trägerplatine festlöten.
- Mit einem Draht oder Kabelbinder das UNIGATE<sup>®</sup> IC auf der Fassung fixieren.
- Manuelles Einlöten direkt auf dem Trägerboard.

- Automatisches Einlöten direkt auf dem Trägerboard, wobei dann "selectiv" gelötet werden muss (kein Wellenlöten)

Die gesockelte Variante hat den Vorteil, dass Script- oder Firmware-Updates leicht eingespielt werden können, wenn das nicht auf dem Trägerboard mit vorgesehen ist. Ausserdem kann so durch Wechsel des UNIGATE® IC's leicht der Feldbus gewechselt werden, wenn die entsprechenden Steckverbinder auf dem Trägerboard vorgesehen sind. Ein weiterer Vorteil ist, dass - in der Regel - die Trägerplatine dann nur Reflow gelötet werden muss.

Die gelötete Variante hat den Vorteil, dass die Bauhöhe geringer ist, und eine höhere Schock- und Vibrationssicherheit gegeben ist.

### **3.10 Unterschiede zwischen den verschiedenen Ausführungen**

#### **3.10.1 Standard Ausführung**

(z.B. V3473 UNIGATE® IC-PBDPL-5V)

In der normalen Ausführung, die in der Regel angewendet wird, ist das IC "vollbestückt"; d.h. sie müssen nur noch den D-Sub eindesignen, und fertig! (siehe im 3.6, auf Seite 18 das Layout-Beispiel "UNIGATE® IC PROFIBUS DP variant1")

#### **3.10.2 Ausführung "ohne 485 (PROFIBUS)-Treiber"**

(z.B. V3626 UNIGATE® IC-PBDPL-5V-without RS485-driver) Für Anwendungen, bei denen das UG-IC nicht in unmittelbarer Nähe des PROFIBUS-Steckers angeordnet werden kann, sollte die Lösung "ohne Treiber" verwendet werden. Der technische Hintergrund ist Folgender: Im PROFIBUS können in einem Segment bis zu 32 Teilnehmer betrieben werden. Je höher die verwendete Baudrate ist, um so kritischer werden Stichleitungen auf der PROFIBUS-Leitung. Da jede Verbindung vom PROFIBUS-Stecker bis zum Treiber als Stichleitung zu sehen ist, muss hier also aufgepasst werden. In der Praxis kann man davon ausgehen, dass Leitungen bis 25mm Länge auch bei der höchsten Baudrate (12 Mbaud) keine Probleme darstellen; d.h. wenn die Leitung zwischen UG-IC und PROFIBUS-Stecker kleiner als 25mm ist, kann mit dem Standard UG-IC gearbeitet werden, bei größerem Abstand sollte die Ausführung "ohne 485 (PROFIBUS) Treiber" eingesetzt werden. (siehe 3.7, auf Seite 21).

#### **3.10.3 Ausführung "LWL"**

(z.B. V3743 UNIGATE® IC-LWL-PBDPL-5V)

Bei der LWL-Ausführung fehlt ausser dem Treiber (s.o.) auch noch die gesamte galvanische Trennung (Optokoppler und DC/DC-Wandler). Diese Ausführung kann - wie der Name schon sagt - für eine LWL-Anbindung verwendet werden.

Sie findet aber auch Einsatz bei Kunden, bei denen unsere 1kV galvanische Trennung nicht ausreicht. Diese Kunden können dann extern in ihrem Layout eine "bessere" galvanische Trennung eindesignen.

Verwendung kann diese Variante auch finden, wenn unsere Kunden mit der Höhe des UG-IC ein Problem haben (bei DPL und DPX). Hier kann dann der DC/DC-Wandler, der in diesem Fall die Höhe vorgibt, auf der Kundenplatine montiert werden.

## 4 Die serielle Schnittstelle

### 4.1 Übersicht

Die serielle Schnittstelle ist die wichtigste Verbindung zwischen dem UNIGATE® IC und dem Microcontroller Ihrer Anwendung. Die Schnittstelle ist so ausgelegt, dass Ihre Anwendung zumindest auf der Softwareseite nicht geändert werden muss; der große Leistungsumfang der seriellen Schnittstelle des UNIGATE® IC bietet hierfür die Grundlage. Das UNIGATE® IC ermöglicht den Anschluss von Controllern mit einer Baudrate von 110 Baud – 625 kBaud. Die Baudrate für die Kommunikation selbst ist im Modul permanent gespeichert. Die maximale Größe für EA-Daten können Sie in der Tabelle in Kapitel 12.2.1 unter „RS-Puffergröße“ nachlesen. Des Weiteren kann die maximale Größe für EA-Daten mit dem Scriptbefehl `Get RSOutBufFree16` ausgelesen werden.

Je nach eingespieltem Script des UNIGATE® IC führt das Modul selbständig Aktionen aus, um vom angeschlossenen Gerät Daten zu ermitteln. Für Kunden, die bereits eine Software-Adaption für ein UNIGATE® bei der Fa. Deutschmann besitzen, kann auch dieses Protokoll bzw. Script nach einer Anpassung vom IC abgearbeitet werden.

In jedem Fall wird das IC die Kommunikation mit dem Feldbus eigenständig übernehmen.

### 4.2 Initialisierung der seriellen Schnittstelle

Die Initialisierung der Schnittstelle erfolgt durch Scriptbefehle, wie z.B. `"Set Baudrate"`, `"Set Databits"`, `"Set Parity"`. Für eine detaillierte Beschreibung dieser Befehle siehe die Online-Dokumentation zum Protocol Developer bzw. das Handbuch zum Protocol Developer.

### 4.3 Benutzung der seriellen Schnittstelle

Die serielle Schnittstelle kann frei durch den Benutzer programmiert werden. Es stehen leistungsfähige Script Befehle zum Senden und Empfangen der Daten zur Verfügung; beispielhaft seien nur einige Möglichkeiten genannt, z.B. auf ein Zeichen mit Timeout zu warten, auf eine feste Anzahl Zeichen zu warten oder auch Daten im Modbus RTU Modus zu Senden und zu Empfangen. Eine Referenz zu diesen Befehlen bietet die Online-Dokumentation zum Protocol Developer sowie das Handbuch zum Protocol Developer.

### 4.4 Weitere Betriebsmodi

Die serielle Schnittstelle dient in den Modi Konfigurationsmodus sowie Firmware-Update-Modus auch zum Konfigurieren der Standardsoftware bzw. zum Durchführen eines Firmware Update. Details hierzu finden Sie im Kapitel 11.5, auf Seite 38.

## 5 Synchrone serielle Schnittstelle

Die synchrone serielle Schnittstelle des UNIGATE® IC dient der Anbindung von getakteten Schieberegistern oder Bauteilen, die über ein *Serial Peripheral Interface (SPI)* verfügen. Sie ermöglicht die Erweiterung des IC um digitale Ein- und Ausgänge (z. B. zur Ansteuerung von LEDs oder zum Einlesen von Schalterstellungen), die Kommunikation mit Mikrocontrollern oder die Ansteuerung von DA- oder AD-Wandlern. Anschlussbeispiele sind in Kapitel 3 gegeben. Durch die Verwendung der synchronen seriellen Schnittstelle lassen sich Produkte realisieren, die ohne einen weiteren Mikrocontroller auskommen (Stand-Alone-Betrieb). Beispiele hierfür sind Sensorprodukte oder digitale EA-Module.

### 5.1 Schieberegisterbetrieb

Bevor die Schnittstelle verwendet werden kann, muss sie zunächst durch das Setzen diverser Script-Parameter initialisiert werden (Kapitel 5.1.1).

Die Parameter `ShiftRegisterInputType` und `ShiftRegisterOutputType` erlauben die Verwendung verschiedener Schieberegistertypen, die sich in der Polarität der Schieberegistersignale unterscheiden. Zur Verwendung der Schieberegistertypen 74595 und 74165 können beispielsweise die Werte `RiseClk_RiseLoad` und `RiseClk_LowLoad` gesetzt werden. Die Schieberegisterbreite wird über die Parameter `ShiftRegisterInputBitLength` und `ShiftRegisterOutputBitLength` eingestellt. Die Maximalbreite beträgt je 256 Bit. Der Datenaustausch mit den angeschlossenen Schieberegistern erfolgt mit den Befehlen `WriteShiftRegister`, `ReadShiftRegister` oder bidirektional mit dem Befehl `ShiftRegisterDataExchange`. Die Taktrate bewegt sich zwischen 280 kHz und 320 kHz.

Weiterführende Informationen zu den Befehlen und Parameterwerten finden Sie in der Hilfe der Protocol Developer Software. Auf Wunsch kann die Deutschmann Script-Sprache um zusätzliche Parameterwerte ergänzt werden, um weitere Schieberegistertypen zu unterstützen.

#### 5.1.1 Script-Beispiel

Hinweis: das Script-Beispiel bezieht sich auf das Schaltungsbeispiel in Kapitel 3.5

```
var InBuffer: Buffer[2];
var OutBuffer: Buffer[2];
MoveConst( OutBuffer[0], #0x58#0x21 );

Set( ShiftRegisterInputType, RiseClock_FallLoad );
Set( ShiftRegisterOutputType, RiseClock_RiseLoad );

Set( ShiftRegisterInputBitLength, 16 );
Set( ShiftRegisterOutputBitLength, 16 );

WriteShiftRegister( OutBuffer[0] );
ReadShiftRegister( InBuffer[0] );

// Eingangsdaten befinden sich jetzt in InBuffer
// 0x58 liegt am Analogwandler an
// 0x21 an den Ausgängen des Schieberegisters
```



## 5.2 SPI-Betrieb

Zur Nutzung der Schnittstelle im SPI-Betrieb muss diese zunächst initialisiert werden. Mit dem Befehl `InitSPI` werden die Betriebsart, der Modus (Signalpolarität und phase) und die Taktfrequenz festgelegt.

Der Datenaustausch erfolgt mit dem Befehl `ExchangeSPI`. Die maximale Taktfrequenz beträgt je nach Hardware 1 bzw. 5 MHz. Details entnehmen Sie bitte der IC-Pinout-Liste im Download-Bereich unserer Web-Seite.

Bitte beachten Sie auch die Dokumentation der Script-Befehle in der Hilfe des Protocol Developers.

### 5.2.1 Script-Beispiel

```
var  L_Freq      : long;
var  b_Channel   : byte;
var  w_Len       : word;
var  a_BufOut    : buffer[100];
var  a_BufIn     : buffer[100];

moveconst(L_Freq, 1000000); // 1 MHz
InitSPI(1 , 0 , L_Freq);

moveconst(b_Channel, 0);
moveconst(w_Len, 11);
moveconst(a_BufOut[0], "Hello World");
ExchangeSPI(b_Channel, w_Len, a_BufOut[0], a_BufIn[0]);
```

## **6 Die Debug Schnittstelle**

### **6.1 Übersicht über die Debug Schnittstelle**

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

### **6.2 Start im Debugmode**

Mit dem Anlegen von Spannung an das UNIGATE® IC (Power up) wird die Firmware nach Ausführung eines Selbsttests auf dieser Schnittstelle das binäre Zeichen 0 (0x00) ausgeben. Wenn das IC 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.

### **6.3 Kommunikationsparameter für die Debugschnittstelle**

Die Debugschnittstelle arbeitet immer mit 9600 Baud, kein Paritätsbit, 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.

### **6.4 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.

### **6.5 Befehle der Debug Schnittstelle**

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

## 7 Script und Konfiguration

### 7.1 Übersicht

Das im UNIGATE® IC gespeicherte Script und die Konfiguration können über die serielle Schnittstelle (Applikation) im Konfigurationsmodus ersetzt oder upgedated werden.

### 7.2 Der Konfigurationsmodus

Wird der Pin "ConfigMode" während des PowerUp oder Reset auf GND gezogen, startet das UNIGATE® IC im Konfigurationsmodus. In diesem Modus besteht die Möglichkeit mit dem IC zu kommunizieren, ohne dass die normale Software ausgeführt wird. Man kann in diesem Modus Einstellungen der Standardsoftware des UNIGATE® IC verändern oder auch ein neues Script in das UNIGATE® IC schreiben. Das UNIGATE® IC zeigt seinen Start im Konfigurationsmodus durch die Ausgabe einer Statusmeldung an, die z. B. so aussehen kann:

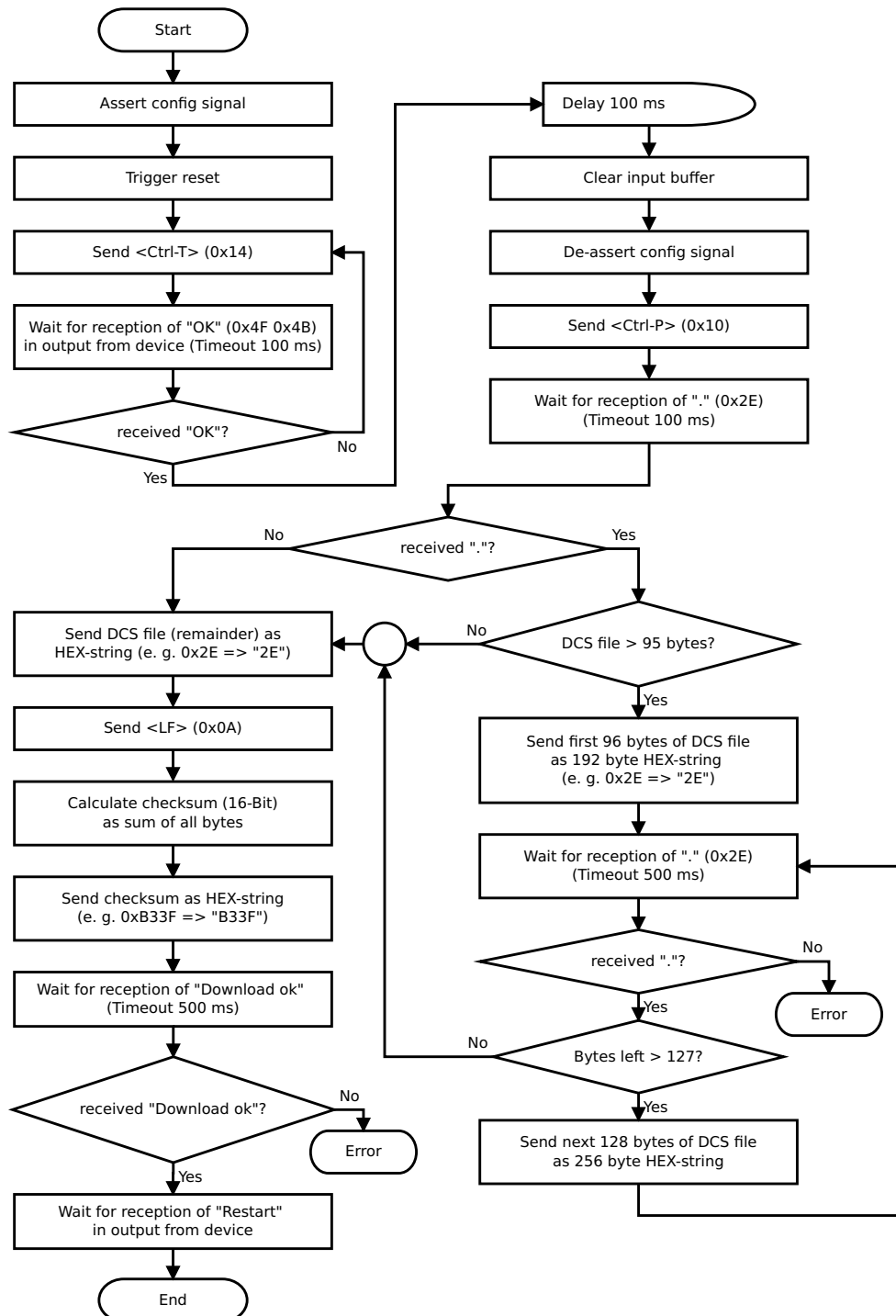
```
IC-PB-SC V5.9A[25] (c)dA Script(2k)="Leer" Author="Deutschmann Automation GmbH"  
Version="1.0" Date=21.08.2001 SN=47110001 ID=1
```

Beim UNIGATE® IC PROFIBUS beschränkt sich die Konfiguration auf das Einstellen der PROFIBUS-Adresse. Siehe auch Kapitel Kapitel 9.2 "Einstellen der PROFIBUS-Adresse".

### 7.3 Script updaten

- Die bevorzugte Variante ist die, dass das UNIGATE® IC in der von Deutschmann erhältlichen Basisplatine (Developerboard UNIGATE® IC-AB) eingesetzt ist und die Deutschmann Tools (Software WINGATE mit "Write Script" unter "File" oder mit der Software ScriptProgramTool) benutzt werden.
- Sie können das Script auch in Ihrer Applikation von Ihrem Host automatisch ersetzen (siehe Ablaufdiagramm)

## Script Download



**Der Ablauf ist wie folgt:**

Das Gateway muss sich im Config-Mode befinden.  
Der Script-Download wird eingeleitet mit "Ctrl-P (=0x10)".  
Danach folgen die Daten byteweise als ASCII-Hex-Zeichen.  
Der Download wird beendet mit einem "LF (=0x0A)".  
Danach folgt noch die Wort-Checksumme als ASCII-Hex-Zeichen.

Das Gateway antwortet mit einer Klartextantwort auf diesen Download und führt einen Warmstart durch.

Beispiel:

Es soll folgendes 4 Byte Script heruntergeladen werden: 0x01 0x12 0x5A 0x23

Die Summe der Bytes ergibt 0x0090 als Checksumme

Dann muss folgende Sequenz geschickt werden:

1. 0x10 Ctrl-P
2. 0x30 '0'
3. 0x31 '1'
4. 0x31 '1'
5. 0x32 '2'
6. 0x35 '5'
7. 0x41 'A'
8. 0x32 '2'
9. 0x33 '3'
10. 0x0A LF
11. 0x30 '0'
12. 0x30 '0'
13. 0x39 '9'
14. 0x30 '0'

Gateway-Antwort: "Download ok"

## **7.4 Konfiguration des UNIGATE® IC**

Das UNIGATE® IC wird mit einem Leer Script ausgeliefert.

Die Konfiguration beschränkt sich beim UNIGATE® IC - PROFIBUS DP auf die Einstellung der Feldbus-Adresse, die mit der Software WINGATE® geändert werden kann.

### **7.4.1 PROFIBUS**

- Konfigurationsdaten: gemäß GSD-Datei (DAGW2079 bzw. UGIC3218)
- Diagnosedaten: max. 8 Byte (s. Kapitel Fehlerbehandlung)
- Baudrate: automatische Erkennung bis 12 Mbaud
- Sync: unterstützt
- Freeze: unterstützt
- Ident-Nr.: 0x2079 bzw. 0x3218

#### 7.4.2 RS232/RS485/RS422

- RS-Type: RS232
- Startbit: 1
- Datenbits: 8
- Stopbit: 1
- Parity: Kein
- Baudrate: 9600 Baud
- Default-Einstellung Diese Konfiguration kann über das Script geändert werden.

## **8 Erstellung eines Scripts**

### **8.1 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.

### **8.2 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.

### **8.3 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 umrissenen 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.

### **8.4 Unabhängigkeit von Bussen**

Prinzipiell sind die Scripte 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 Scripte 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.

### **8.5 Weitere Einstellungen am IC 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 unangeändert, 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.

## 8.6 Die Benutzung des Protocol Developer

Das Softwaretool Protocol Developer kann von unserer Internetseite [www.deutschmann.de](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.

## 8.7 Genauigkeiten der Baudraten bei UNIGATE® IC

Die Baudrate der seriellen Schnittstelle wird aus der Quarzfrequenz des Prozessors abgeleitet. Zwischenzeitlich arbeiten alle Script-Gateways 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:

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

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

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

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:

$\text{F32} = 40000000 / 32 = 1250000$

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

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

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%

## 8.8 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 das Problem, sondern die Anzahl Zeichen, die pro Sekunde übertragen werden.
- Datenlast auch auf der Feldbusseite nicht unnötig gross machen. Insbesondere bei azyklischen 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.

## 9 PROFIBUS DP

Das UNIGATE<sup>®</sup> IC PROFIBUS DP unterstützt zur Zeit PROFIBUS DP in der Varianten DPV0 - DPV2. Jede erdenkliche Kombination von Input- und Outputgrößen ist möglich. Sollte eine von Ihnen gewünschte Kombination nicht in den GSD Dateien (DAGW2079.GSD, UGIC3218.GSD) enthalten sein, kann sie durch die Angabe von Konfigurationsoctets beschrieben werden (siehe Kapitel 15).

Das UNIGATE<sup>®</sup> IC unterstützt automatisch die PROFIBUS-Funktionen Sync und Freeze. User Parameter werden ab V. 7.0 (IC-PB-DPL) unterstützt.

### 9.1 Beschreibung der DPV1- / DPV2-Funktionen

#### 9.1.1 DPV1

Die DPV1-Erweiterung besteht aus folgenden Funktionen:

##### 1. *Azyklischer Datenaustausch mit Klasse1-Master (z. B: SPS)*

Diese Funktion ist für einen DPV1-Slave optional. Unsere Gateways unterstützen diese Funktion standardmäßig. Der Klasse1-Master kann mittels dieser Funktion azyklisch Daten vom Slave lesen und schreiben. Diese Daten werden vom Script im Gateway verarbeitet. Der Kanal für den azyklischen Datenaustausch wird fest während der Parametrierung eingerichtet.

##### 2. *Azyklischer Datenaustausch mit Klasse2-Master (z. B: Bediengerät)*

Diese Funktion ist für einen DPV1-Slave ebenfalls optional. Unsere Gateways unterstützen diese Funktion standardmäßig. Der Klasse2-Master kann mittels dieser Funktion azyklisch Daten vom Slave lesen und schreiben. Diese Daten werden vom Script im Gateway verarbeitet. Der Kanal für den azyklischen Datenaustausch wird vor jedem Datenaustausch aufgebaut, und danach wieder geschlossen.

##### 3. *Alarmbehandlung*

Auch die Alarmer sind optional. Wenn sie aktiviert sind, ersetzen sie die gerätespezifische Diagnose. Unser Gateway unterstützt zur Zeit keine Alarmer.

Jeder DPV1-Slave muss die erweiterte Parametrierung unterstützen, da im Octet 8 des Parametrieretelegramms festgelegt wird, ob es sich um einen DPV0 oder DPV1-Slave handelt.

Ein DPV1-Slave kann auch an einem DPV0-Master betrieben werden, wenn die DPV1-Funktionen abgeschaltet bleiben.

#### 9.1.2 DPV2

Die DPV2-Erweiterung besteht aus folgenden Funktionen:

##### 1. *Isochron Mode (IsoM)*

Hierunter versteht man das taktsynchrone Verhalten eines Bussystems. Diese Funktion ist optional für einen DPV2-Slave, und wird über die GSD-Datei aktiviert. Unser Gateway unterstützt zur Zeit diesen Mode nicht.

##### 2. *Data Exchange Broadcast (DxB)*

Hierunter versteht man die Kommunikation zwischen Slaves (Querverkehr). Diese Funktion ist optional für einen DPV2-Slave, und wird über die GSD-Datei aktiviert. Unser Gateway unterstützt zur Zeit nur die Funktion des "Publisher" (Daten an andere Slaves senden). Die Funktionalität "Subscriber" (Daten von anderem Slave bekommen) wird zur Zeit nicht unterstützt.

### 3. Up- und Download

Diese Funktion ist ebenfalls optional für einen DPV2-Slave und wird zur Zeit von unserem Gateway nicht unterstützt.

### 4. Uhrzeitsynchronisation (Time stamp)

Diese Funktion ist ebenfalls optional für einen DPV2-Slave und wird zur Zeit von unserem Gateway nicht unterstützt.

### 5. Redundanzkonzept

Diese Funktion ist ebenfalls optional für einen DPV2-Slave und wird zur Zeit von unserem Gateway nicht unterstützt.

## 9.2 Einstellen der PROFIBUS-Adresse

Es gibt verschiedene Möglichkeiten die PROFIBUS-Adresse des IC einzustellen.

#### 1. Festes Einstellen der Adresse über die Konfiguration

Das UNIGATE® IC muss sich im Konfigurationsmodus befinden (siehe Kapitel 7.2 "Der Konfigurationsmodus"). Jetzt ist es mit WINGATE möglich, die Adresse einzustellen. Diese Adresse bleibt so lange erhalten, bis sie erneut geändert wird.

#### 2. Festes Einstellen der Adresse über das Script

Die Adresse kann auch fest im Script hinterlegt werden. Diese Vorgehensweise dürfte allerdings nur für wenige Applikationen interessant sein, da es nötig ist, das Script zu ändern um auch die PROFIBUS-Adresse anzupassen (siehe auch nachfolgendes Scriptbeispiel).

#### 3. Festes Einstellen der Adresse über die serielle Schnittstelle

Es ist auch möglich, die Adresse über die serielle Schnittstelle an das IC zu übertragen. Danach kann die Adresse über das Scriptkommando "SetByVar" eingestellt werden. Diese Möglichkeit sollte dann benutzt werden, wenn ihr Gerät z. B. über eine Bedienfront verfügt und das Menü der Front um die Einstellung "PROFIBUS-Adresse" erweitert werden kann. Das Einstellen der PROFIBUS-Adresse über die serielle Schnittstelle ist für solche Applikationen die komfortabelste Möglichkeit.

#### 4. Festes Einstellen der Adresse über den Bus selbst

Die Adresse 126 ist reserviert für Geräte, die nicht über hardwaremässige Einstellmöglichkeiten für die Bus-Adresse verfügen (z. B. Geräte mit Schutzklasse IP 65). Die Adresse 126 muss über WINGATE konfiguriert werden (siehe Punkt 1.)

#### 5. Anschluss der Drehschalter an die Schieberegister

Es können Drehschalter an ein Schieberegister angeschlossen werden, wie auch auf unserer Basisplatine. Es ist jetzt für das Script möglich, diese Schalter zu lesen und als Feldbusadresse zu setzen. Hierfür kann prinzipiell auch das nachfolgend angegebene Script benutzt werden.

### Scriptbeispiel für die Initialisierung des PROFIBUS

```
var InSize: word;
var OutSize: word;

Set (FieldbusID, 4) ;
// dieser Parameter kann auch über den Befehl SetByVar
// eingestellt werden
// var PBAдресse: long;
// MoveConst( PBAдресse, 4) ;
// oder aus den Schieberegistern
// SetByVar();

BusStart;
// der Profibus ASIC ist jetzt bereit.
// ab jetzt KANN der Master den Teilnehmer konfigurieren.
// Das bedeutet allerdings nicht, dass der Master den
// Teilnehmer bereits konfiguriert hat.

wait (Bus_Active);
// der Profibus ist jetzt im Zustand Data Exchange
// Dieser Befehl kann sehr lange dauern und ist nicht unterbrechbar!
// Jetzt ist im Script auch bekannt, mit welcher Konfiguration
// der Master den Slave in Betrieb genommen hat.

Get ( BusInputSize, InSize);
Get ( BusOutputSize, OutSize);
// InSize und OutSize sind aus der Sicht des IC's !!

// Daten lesen aus dem Bus ist möglich
// es sollten so viele Bytes gelesen werden wie vorhanden sind.
var InBuffer: Buffer[100];
Readbus ( InBuffer[0], InSize) ;

// Daten schreiben ist jetzt möglich.
// Es dürfen nicht mehr bytes geschrieben werden wie vorhanden sind.
var OutBuffer: Buffer[100];
WriteBus ( OutBuffer[0], OutSize );
```

## 10 Fehlerbehandlung beim UNIGATE® IC

Es können zwei Kategorien der System-Errors unterschieden werden:

Schwere Fehler (1-4): 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.

Der System-Error kann über das Script ausgelesen werden.

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

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

Tabelle 1: Fehlerbehandlung beim UNIGATE® IC

## 11 Firmware Update

### 11.1 Übersicht

Das UNIGATE® IC verfügt intern über einen 64 kByte Flash Speicher für die Firmware. Im Firmware-Update-Modus kann über die serielle Schnittstelle des UNIGATE® IC die Firmware ersetzt werden.

### 11.2 Einstellen des Firmware-Update-Modus

#### 11.2.1 Einstellung per Hardware

Das UNIGATE® IC kann per Hardware in den Firmware-Update-Modus gebracht werden. Hierzu muss während des Power-Up-Vorgangs das Signal  $\overline{BE}$  (-Boot enable) auf das Potential GND gezogen werden.

#### 11.2.2 Einstellung per Software

Befindet sich das UNIGATE® IC im Konfigurationsmodus (siehe Kapitel 7.2, auf Seite 27) kann es über das Kommando CTRL-F (0x06) interaktiv in den Firmware-Update-Mode gebracht werden. Nach dem Senden des Kommandos erfolgt eine Sicherheitsabfrage, die mit J oder N (Ja oder Nein) beantwortet werden muss. Nach positiver Bestätigung erfolgt der Neustart des IC im Firmware-Update-Mode.

### 11.3 Durchführung des Firmware-Updates

Der sicherste Weg für den Firmware-Update ist die Benutzung der Basisplatine in Kombination mit der Software "FDT.EXE" (Firmware-Download-Tool). Diese Hilfsmittel sind von Deutschmann zu erhalten (siehe Kapitel 13, auf Seite 42).

Es besteht auch die Möglichkeit sowohl die Beschreibung als auch die Tools des Controllerherstellers (TEMIC, 89C51RD2) zu benutzen.

### 11.4 Sicherheitshinweis

Der Firmware-Update sollte nur dann durchgeführt werden, wenn keine andere Möglichkeit zur Verfügung steht. Ein begonnener Firmware-Update-Vorgang kann NICHT rückgängig gemacht werden. Die vorige Firmware ist damit endgültig unbrauchbar.

### 11.5 Betriebsmodi des IC

#### *Standard-Betriebsmodus*

Dieser Modus wird für die normale Nutzung des IC benötigt. In diesem Modi wird das IC alle Script Befehle abarbeiten und normal die entsprechenden Benutzerdaten austauschen; auch der Bus wird in diesem Modus durch das IC bedient.

#### *Konfigurationsmodus*

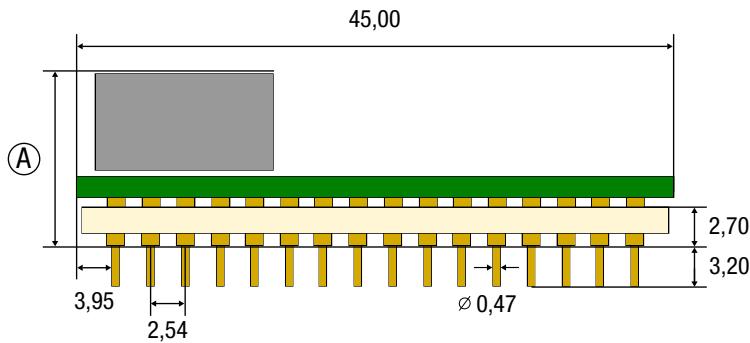
Im Konfigurationsmodus wird das UNIGATE® IC nach dem Start (oder nach einem Reset) einen Selbsttest ausführen und nach erfolgreichem Selbsttest auf weitere Anweisungen warten. Hier ist es möglich ein übersetztes Script in das Gerät zu laden oder auch den Firmware Download Mode zu initiieren.

12 Technische Daten

Hier finden Sie alle nötigen technischen Daten zum UNIGATE® IC.  
Alle Maßangaben in mm.

12.1 Mechanik des UNIGATE® IC

12.1.1 Abmessungen UNIGATE® IC



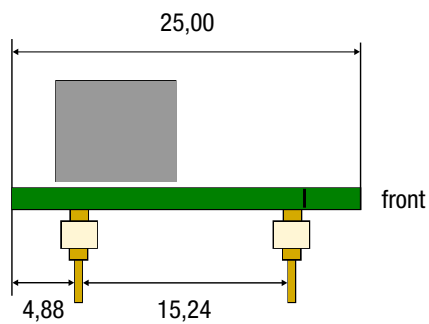
UNIGATE® IC

	CANopen	DeviceNet	EtherCAT	EtherCAT (wt*)	EtherNet/IP 1Port	EtherNet/IP 2Port	Ethernet TCP/IP + Modbus TCP	Ethernet TCP/IP (wt) + Modbus TCP (wt*)	LONWorks	MPI	PROFIBUS DP-DPV1	PROFIBUS DP (PBL/PBX/PBY)	PROFINET 1Port	PRROFINET 2Port (wt*)	RS
A= Gesamt- höhe (mm)	12	9,5	11	8	11	8	11	8	13	12	9,5	12	11	8	12

\*wt = without magnetics

**Hinweis:** Die Gesamthöhe aller UNIGATE® ICs können der "Pinout-Liste für UNIGATE® IC und UNIGATE IC2" entnommen werden. (Download PDF)

Die Stifte der UNIGATE<sup>®</sup> ICs sind im Rastermaß 2,54 mm angeordnet.



#### DIP-Spacing Code 6

Wenn Sie auch an den Einsatz von anderen UNIGATE<sup>®</sup> IC Varianten denken, müssen Sie eine max. Bauhöhe (inkl. Stifte) von  $\leq 20$  mm berücksichtigen.



## 12.2 Technische Daten UNIGATE® IC-PROFIBUS

Merkmale	Erläuterungen
Betriebsspannung	3,3 V / 5 V $\pm$ 5 %, 170 mA (typ.), 200 mA (max.)
Verpolungsschutz Spannungsversorgung	Nein
Schnittstelle	2 UART Schnittstellen, 1 synchron serielle Schnittstelle
Galvanische Trennung Feldbusseite	Standard
Feldbus-ID	Einstellbar über Script
Feldbus-Baudrate	bis 12 MBaud (Autodetect)
UART-Baudrate	bis 625 Kbaud (einstellbar über Script)
Feldbusdatenformat	Bis 96 bzw. bis 244 Byte I/O (je nach Version)
Technologie	ASIC
Sonstiges	Sync, Freeze, Extern können z. B. I/Os, analoge Signale, Schieberegister, LEDs, Schalter etc. angeschlossen werden
Abmessungen	45 x 25 x 9 mm (B x T x H) bzw. 45 x 25 x 12 mm (B x T x H)
Montage	32 DIL
Gewicht	ca. 9 g
Betriebstemperatur	-40°C ..+85°C
Lager-/Transporttemperatur	-40°C..+100°C
Einbaulage	Beliebig

### 12.2.1 Leistungsmerkmale der verschiedenen Versionen

Leistungsmerkmale	Version			
	UNIGATE IC PROFIBUS DPL	UNIGATE IC PROFIBUS DP	UNIGATE IC PROFIBUS DPV1	UNIGATE IC PROFIBUS DPX PROFIBUS DPY
Unterstützung	DPV1 / DPV2	DPV0	DPV1 / DPV2	DPV1 / DPV2
PB-Datenlänge	244 Byte I/O	96 Byte I/O	244 Byte I/O	244 Byte I/O
RS-Puffergröße	256 Byte	100 Byte	1 K	1 K
Bauhöhe in mm	12	9	9	12
Spannungsversorgung in Volt	5V / 3,3V	5V	5V	5V / 3,3V
Zertifiziert	Ja	Ja	Ja	Ja
GSD-Datei	ugic3218.gsd	dagw2079.gsd	ugic3218.gsd	ugic3218.gsd

## 13 Zubehör

Von Deutschmann Automation sind folgende Hilfsmittel zu erhalten.

### 13.1 FirmwareDownloadTool (FDT)

Das FirmwareDownloadTool steht auf der Homepage von Deutschmann Automation zum Download bereit. Es wird benötigt um ein Update der Firmware durchzuführen. Voraussetzung hierfür ist, dass die Software an die serielle Schnittstelle des IC herangeführt werden kann. Die Vorgehensweise für das Update selbst ist in der Software beschrieben.

### 13.2 Protocol Developer

Der Protocol Developer ist die Entwicklungsumgebung für Scripte, die auch den Debugger enthält. In diesem Software Paket ist auch die Dokumentation zu allen Scriptbefehlen enthalten. Auf der Homepage von Deutschmann Automation <http://www.deutschmann.de> liegt diese Software zum Download bereit. Das im PDF-Format verfügbare Handbuch Protocol Developer gibt weiterführende Hinweise zur Benutzung der Software.

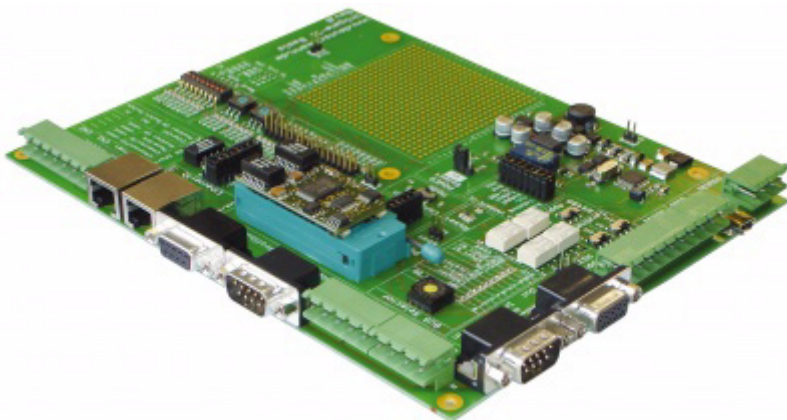
### 13.3 Developer Kit UNIGATE® IC-AB

Das Developer Kit beinhaltet

- ein Developer Board UNIGATE® IC-AB (siehe Kapitel 13.3.1)
- ein Steckernetzteil zur Versorgung des Developer Boards
- sowie Anschlusskabel für Appl. RS232, Debug RS232 und Appl. RS422/485
- USB-Kabel
- Software und Dokumentation ergänzen das Paket.

#### 13.3.1 Developer Board UNIGATE® IC-AB

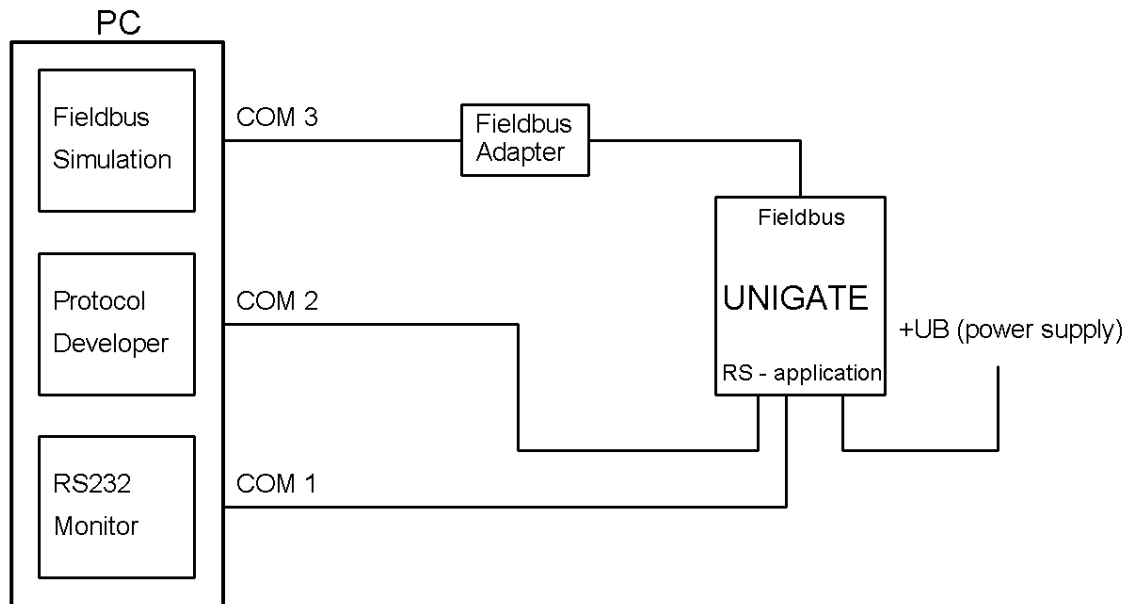
Das Developer Board wurde entwickelt, um die schnelle Implementierung des Deutschmann All-In-One Bus Node UNIGATE® IC in Ihre Elektronik zu gewährleisten. Das Board ist für alle von Deutschmann unterstützten Feldbusse und Industrial Ethernet Busse geeignet.



Das oder die benötigten UNIGATE® ICs werden separat bestellt. Die je nach Ausführung benötigte Spannung (5 Volt oder 3,3 Volt) ist einstellbar. Für die Anbindung an den PC (DEBUG-Schnittstelle) steht sowohl eine RS232-Schnittstelle wie auch ein USB-Anschluss zur Auswahl.

Die Applikationsanbindung erfolgt wahlweise über USB, RS232, RS485 oder die RS422. Zum Testen der jeweiligen Busseite stehen die Busanschlüsse gemäß Norm oder Marktstandard zur Verfügung. Die Deutschmann Add-On-Pakete (Busmaster-Simulation) sind optional erhältlich. Das Board enthält 32 Bit Input und 16 Bit Output, jeweils mit einer LED versehen. Verschiedene Abgriffe erlauben eine einfache Ankopplung an Ihren Prozessor. Ein Lochrasterfeld mit den wichtigsten Signalen (Spannung, IOs) erlaubt eine kundenspezifische Hardwareerweiterung (z. B. um einen D/A Wandler) anzubinden.

### 13.3.2 Schnellstart



Für einen transparenten Datenaustausch können Sie sich das im Protocol Developer hinterlegte File laden (File -> New -> „Profibus\_IC\_BasisBoard.dss“).

## 14 Anhang

### 14.1 Erläuterungen 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	=	PROFIBUS DP
PBDPL	=	PROFIBUS DP-Variante L

---

PBDPX	=	(siehe Vergleichstabelle UNIGATE® IC beim jeweiligen Produkt) PROFIBUS DP-Variante X
PBDPY	=	(siehe Vergleichstabelle UNIGATE® IC beim jeweiligen Produkt) PROFIBUS DP-Variante Y
PBDPV0	=	(siehe Vergleichstabelle UNIGATE® IC beim jeweiligen Produkt) PROFIBUS DPV0
PBDPV1	=	PROFIBUS DPV1
RS	=	Serial RS232/485/422

## 15 Darstellung der Daten in PROFIBUS DP

Jeder normkonforme PROFIBUS DP Master kann Daten mit dem Gateway austauschen. Wegen des Datenaufbaus können auch sehr "einfache" Masteranschlungen verwendet werden.

### 15.1 Konfiguriertelegramm

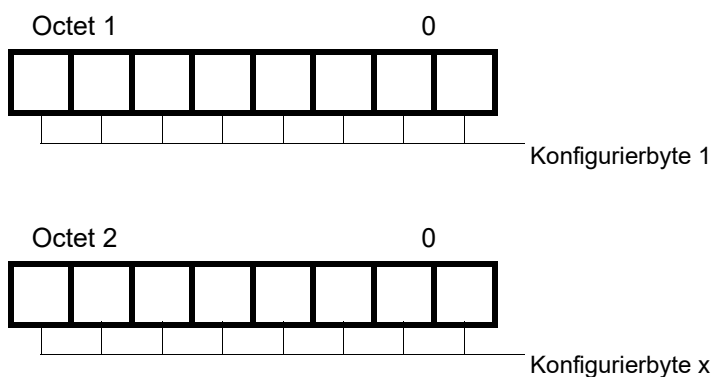
Nach dem Parametrieren hat der Master ein Konfiguriertelegramm an den entsprechenden Slave zu schicken. Über das Konfiguriertelegramm erhält der Slave die Informationen über die Länge der Ein/Ausgabe-Daten. Hat der Anwender beim UNIGATE® RS das Flag 'Längenbyte' gesetzt, sind das die maximalen Datenlängen, ansonsten die tatsächlichen Längen.

Das Konfigurier-Telegramm stellt der Anwender normalerweise auch im Projektierungswerkzeug zusammen, wo er evtl. auch den Adressbereich angeben kann, in dem die Nutz-Daten abgelegt sind.

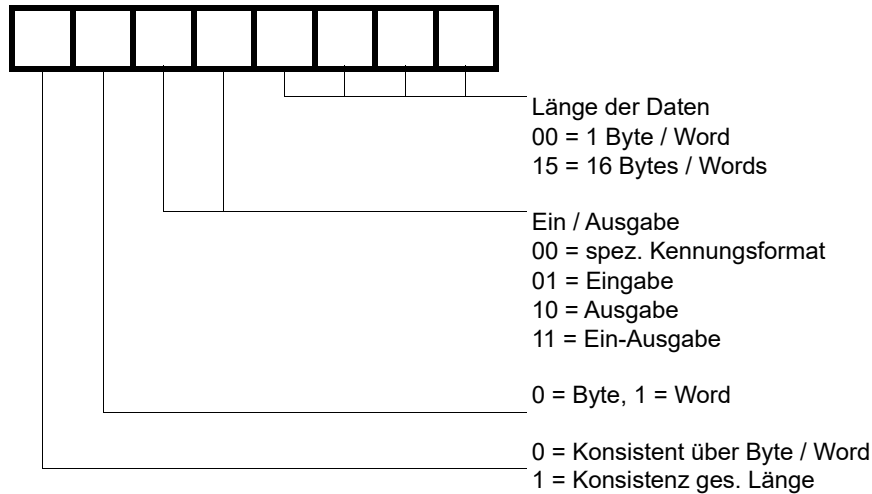
In einem Octet der DataUnit (DU) können Sie bis zu 16 Bytes oder Words beschreiben. Ein- und Ausgänge die gleiches Format haben, können Sie in einem Octet zusammenfassen. Ansonsten sind so viele Octets zu benutzen, wie viele unterschiedliche Bytes/Words Sie benutzen wollen, die sich nicht in einem Octet zusammenfassen lassen.

Entdeckt die Baugruppe bei der Überprüfung, dass die maximal zulässigen Ein/Ausgabedatenlängen überschritten wurden, meldet er bei späterer Diagnoseabfrage die falsche Konfigurierung an den Master. Sie ist dann nicht für den Nutzdatenverkehr bereit.

### 15.2 Konfiguriertelegramm



Aufbau eines Octets im Konfiguriertelegramm:



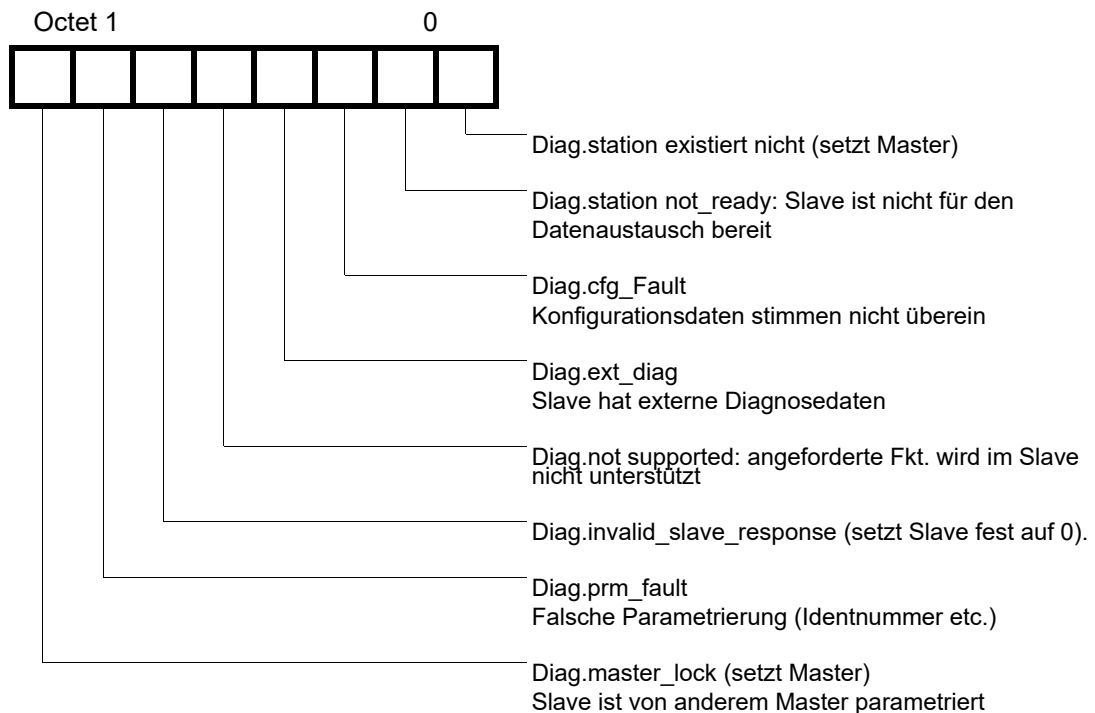
### 15.3 Diagnose

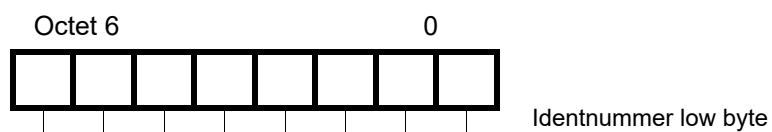
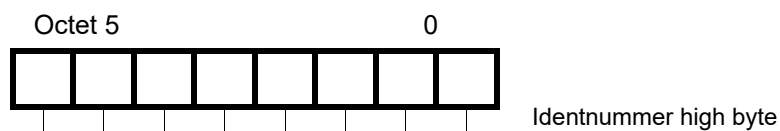
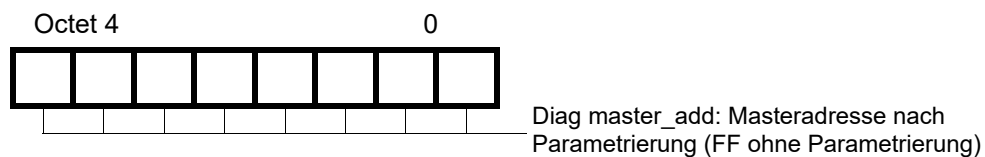
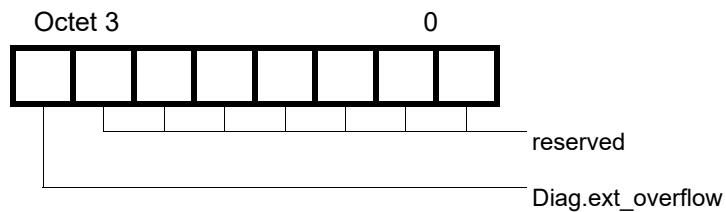
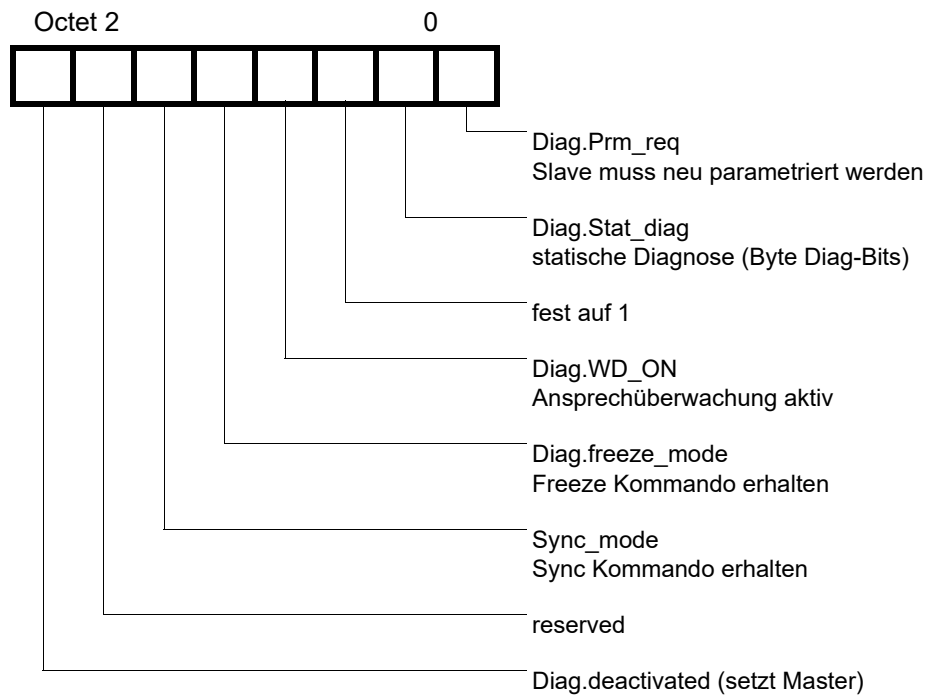
Diagnosedaten sind hochpriorie Daten. Das Gateway erzeugt eine externe Diagnose, wenn es einen internen Fehler erkennt.

Darstellung der Meldungen im externen Diagnosebyte:

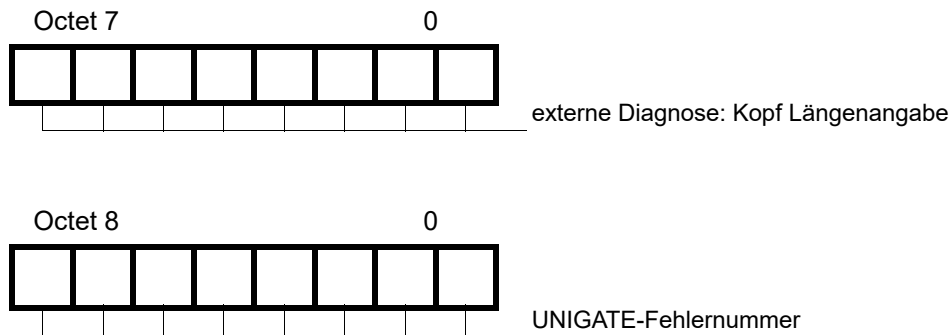
Die Diagnoseinformationen eines DP-Slaves bestehen aus Standarddiagnoseinformationen (6 Bytes) und einer anwenderspezifischen Diagnoseinformation. (Fehlernummer)

Telegramm zur Diagnoseanforderung:









### 15.3.1 Diagnose in DPV1

Die "externe Diagnose" von DPV0 (alter PB) wird bei DPV1 anders genutzt; und zwar werden bei DPV1 dort die Alarmer und Statusmeldungen übertragen. Da wir die Gateway-Fehlernummern in der externen Diagnose übertragen haben, war es nun notwendig, bei DPV1 eine Anpassung vorzunehmen. Um mit DPV1-Mastern kompatibel zu sein, sind nun noch die 3 Bytes "0x81, 0x00, 0x00" vor der eigentlichen Fehlermeldung eingefügt worden. Damit erkennt ein DPV1-Master nun unseren Gateway-Fehler als Statusmeldung.

Somit ergibt sich also eine unterschiedliche Darstellung unseres Gateway-Errors im PB: Bei DPV0 wird die Fehlernummer unverändert als 1 Byte externe Diagnose übertragen. Ist DPV1 über die GSD-Datei im Gateway aktiviert, erfolgt die Fehlernummer als 1 Byte Statusmeldung. In dem Fall, dass DPV1 aktiviert ist, und ein Master angeschlossen ist, der die Alarmer und Statusmeldungen nicht unterstützt, erscheint unsere Gatewayfehlnummer als "externe Diagnose" von 4 Byte (s.o.), wobei die Fehlernummer im 4. Byte enthalten ist, und die vorangehenden drei Byte (0x81, 0x00, 0x00) ignoriert werden können

## 15.4 Datenaustausch

Nachdem der Master in der Diagnose erkennt, dass der Slave für den Datenaustausch bereit ist, sendet er Datenaustauschtelegramme. Die Daten in Ein / Ausgangsrichtung legt der Master entweder in dem Adressbereich ab, der bei der Projektierung angegeben wurde, oder das Steuerungsprogramm muss die Daten über best. Funktionsbausteine holen bzw. bereitstellen.

## 16 Service

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

- FAQ/Wiki Bereich unserer Homepage [www.deutschmann.de](http://www.deutschmann.de) oder [www.wiki.deutschmann.de](http://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 1 Arbeitstag, selten länger als 3 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](mailto:support@deutschmann.de)

### 16.1 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.

### 16.2 Download von PC-Software

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

<http://www.deutschmann.de>



