



Deuschmann

your ticket to all buses

**Bedienerhandbuch
Universelles Feldbus-Gateway
UNIGATE® IC - LONWorks (62)**



Deuschmann Automation GmbH & Co. KG
www.deuschmann.de | wiki.deuschmann.de

1	Allgemeine Einführung	10
2	Das UNIGATE® IC	11
2.1	Technische Einführung	11
2.2	Verfügbarkeit	11
2.3	Firmware	11
2.4	Die serielle Standardschnittstelle	11
2.5	Die synchrone serielle Schnittstelle	11
2.6	Die Debug Schnittstelle	11
2.7	UNIGATE® IC Hardwareübersicht	12
3	Hardware Design	13
3.1	Anschlüsse	13
3.2	Pinout	13
3.2.1	-Boot enable	14
3.2.2	Load out	14
3.2.3	Data out	14
3.2.4	Data In	14
3.2.5	Load In	14
3.2.6	Clock	14
3.2.7	-Reset In	14
3.2.8	Service	14
3.2.9	LED - LN	15
3.2.10	-Config Mode	15
3.2.11	DbgTX, DbgRx	15
3.2.12	TE	15
3.2.13	TX, RX	15
3.3	Software	15
3.4	Prinzipielle Vorgehensweise	16
3.5	Anschlussbeispiele	16
3.6	Layout-Beispiele	19
3.7	Handling (Befestigung des UNIGATE® IC auf der Trägerplatine)	21
4	Die serielle Schnittstelle	22
4.1	Übersicht	22
4.2	Initialisierung der seriellen Schnittstelle	22
4.3	Benutzung der seriellen Schnittstelle	22
4.4	Weitere Betriebsmodi	22
5	Synchrone Schnittstelle	23
5.1	Übersicht über die synchrone serielle Schnittstelle	23
5.2	Scriptbeispiel	23
6	Die Debug Schnittstelle	25
6.1	Übersicht über die Debug Schnittstelle	25

6.2	Start im Debugmode	25
6.3	Kommunikationsparameter für die Debugschnittstelle	25
6.4	Möglichkeiten mit der Debug Schnittstelle	25
6.5	Befehle der Debug Schnittstelle	25
7	Script und Konfiguration	26
7.1	Übersicht	26
7.2	Der Konfigurationsmodus	26
7.3	Script updaten	26
7.4	Konfiguration des UNIGATE® IC	28
7.4.1	LONWorks	28
7.4.2	RS232/RS485/RS422	28
8	Erstellung eines Scripts	30
8.1	Was ist ein Script	30
8.2	Speichereffizienz der Programme	30
8.3	Was kann man mit einem Script Gerät machen	30
8.4	Unabhängigkeit von Bussen	30
8.5	Weitere Einstellungen am IC Gateway	31
8.6	Die Benutzung des Protocol Developer	31
8.7	Genauigkeiten der Baudraten bei UNIGATE® IC	31
8.8	Scriptarbeitungszeiten	32
9	LONWorks	34
9.1	Allgemeines über UNIGATE® LONWorks – RS232/RS485	34
9.2	Knoten	34
9.3	Netzwerkvariablen (NVTs)	34
9.3.1	Implementierung der Netzwerkvariablen in das UNIGATE®-LONWorks	35
9.3.2	Installation	35
9.3.3	Technische Daten (Merkmale) für LONWorks.	35
9.4	Scriptbeispiel für die Initialisierung des LONWorks	36
9.5	LON-spezifische Parameter	38
10	Fehlerbehandlung beim UNIGATE® IC	40
11	Firmware Update.	41
11.1	Übersicht	41
11.2	Einstellen des Firmware-Update-Modus	41
11.2.1	Einstellung per Hardware	41
11.2.2	Einstellung per Software.	41
11.3	Durchführung des Firmware-Updates	41
11.4	Sicherheitshinweis	41
11.5	Betriebsmodi des IC	41

12 Technische Daten	42
12.1 Mechanik des UNIGATE® IC	42
12.2 Technische Daten UNIGATE® IC-LONWorks	44
13 Zubehör	45
13.1 FirmwareDownloadTool (FDT)	45
13.2 Protocol Developer	45
13.3 Developer Kit UNIGATE® IC-AB	45
13.3.1 Developer Board UNIGATE® IC-AB	45
13.3.2 Schnellstart	46
14 Anhang	48
14.1 Erläuterungen der Abkürzungen	48
15 Service	51
15.1 Einsendung eines Gerätes	51
15.2 Download von PC-Software	51

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 mit einer FTT-10A Schnittstelle 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® 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 22.

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 und Softwarebeispiele finden Sie im Kapitel 5.2, auf Seite 23. 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.

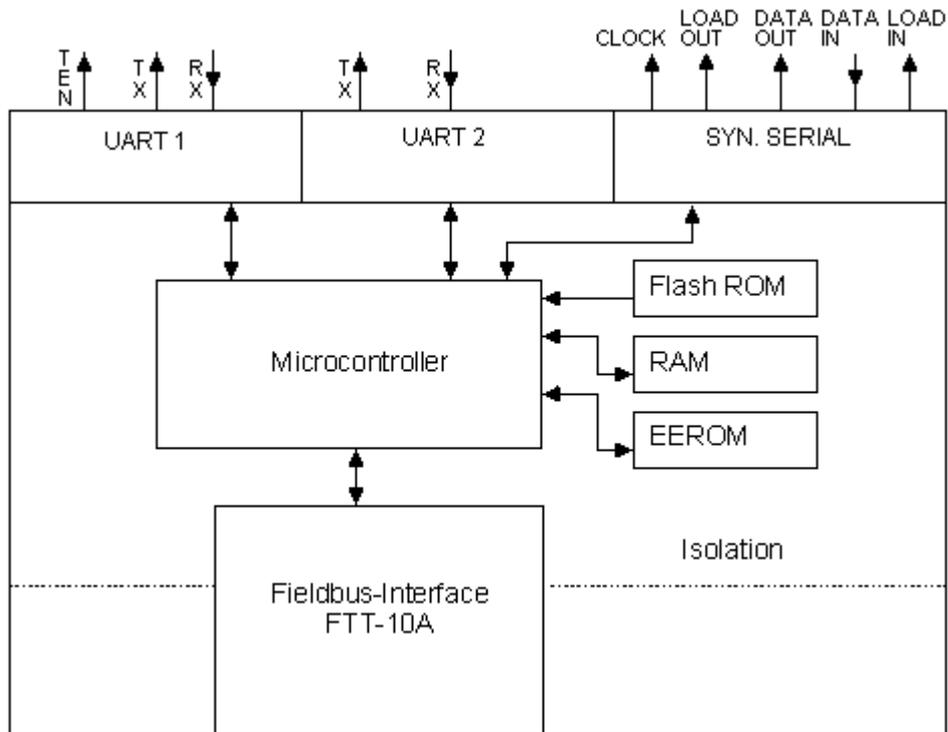
2.6 Die Debug Schnittstelle

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.

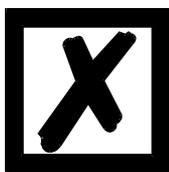


3 Hardware Design

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 entnommen werden.



Für ALLE 32 Pins sind im Layout Bohrungen vorzusehen.

3.2 Pinout

Pin	Technische Spezifikationen	Name	Beschreibung	Anmerkung
1	5V ± 5%, max. 75 mA DC	Vcc	+ 5 V Spannungsversorgung	
2	IN _{Logic}	-BE	Boot enable	
3	OUT _{Driver}	Load out	Übernahmesignal der Ausgangsdaten der synchronen seriellen Schnittstelle	
4	OUT _{Driver}	Data out	Ausgangsdaten für synchrone, serielle Schnittstelle	
5	IN _{Logic}	Data in	Eingangsdaten der synchronen, seriellen Schnittstelle	Internal pull-up
6	OUT _{Logic}	Load in	Übernahmesignal der Eingangsdaten der synchronen seriellen Schnittstelle	
7	OUT _{Driver}	Clock	Taktsignal für synchrone, serielle Schnittstelle	
8	IN _{Reset}	-Reset In*)	Reset-Eingang des IC	Internal pull-up
9	connected to pin 1	Vcc	+ 5 V Spannungsversorgung	
10-12	nc	nc	kein Pin vorhanden	
13	nc	nc		
14	Standard	LON_A	LONWorks	
15	Standard	LON_B	LONWorks	
16		Erde		
17	IN/OUT	Service	Service Pin Taste	
19-20	nc	nc		
21-23	nc	nc	kein Pin vorhanden	
24	connected to pin 32	GND	Ground Spannungsversorgung des IC	
25	*) OUT	LED-LN	Bus State	
26	IN _{Logic}	-Config Mode	Signal zum Starten im Konfigurationsmodus	Internal pull-up
27	OUT _{Logic}	DbgTX	serial Debug TX	
28	IN _{Logic}	DbgRX	serial Debug RX	Internal pull-up
29	IN _{Logic}	RX	serielle Daten RX	Internal pull-up
30	OUT _{Logic}	TX	serielle Daten TX	
31	OUT _{Logic}	TE	Transmit enable	
32	GND	GND	Ground Spannungsversorgung des IC	

*) A-gn = Anode der grünen LED

	V_{IL}	V_{IH}
IN_{Reset}	< 0,7V	> 3,85V
IN_{Logic}	< 0,8V / 0,5mA	> 1,95V / 10 μ A

	V_{OL}	V_{OH}
OUT_{Logic}	< 0,6V / 1mA	> 3,8V / 0,1mA
OUT_{Driver}	< 0,33V / 4mA	> 3,8V / 4mA

3.2.1 -Boot enable

Mit dem Pegel GND während dem Power Up Vorgang wird das IC im Firmware Update Mode gestartet.

3.2.2 Load out

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

3.2.3 Data out

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

3.2.4 Data In

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

3.2.5 Load In

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

3.2.6 Clock

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

3.2.7 -Reset In

A reset of the UNIGATE[®] can be triggered with this input. There is an internal pull-up resistor so that the input can remain unconnected. The pulse duration should be at least 10 ms.

3.2.8 Service

An diese Leitung muss eine Taste nach GND angeschlossen werden. Beim Betätigen der Taste sendet das IC eine Service Pin Nachricht auf das LON Netz. Damit kann das IC am Netz angemeldet werden. An diesem Pin 17 kann auch eine LED angeschlossen werden. Pin 17 an Kathode, Anode über 1,5 kOhm an +5V. (siehe Kapitel 3.6)

- LED an: Der Knoten hat keinen gültigen Applikationscode.
- LED blinkt: Blinkt mit der Frequenz von 1/2 Hz. Der Knoten hat einen Applikationscode aber keine Netzapplikationsinformation.
- LED aus: Der Knoten hat beides, sowohl Applikation und Konfiguration.

3.2.9 LED - LN

Sie blinkt beim Betätigen der Service Taste oder einem Wink Kommando und leuchtet wenn einmal Daten über das LON Netz an das IC ausgetauscht wurden.

-> Bei RS485-Betrieb kann diese LED nicht genutzt werden!

3.2.10 -Config Mode

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

3.2.11 DbgTX, DbgRx

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

3.2.12 TE

Das Transmit Enable Signal ermöglicht den Anschluss von RS485 Treibern an der seriellen Schnittstelle des IC. Das Signal wird immer dann auf 5V geschaltet, wenn das IC über die Leitung TX sendet.

3.2.13 TX, RX

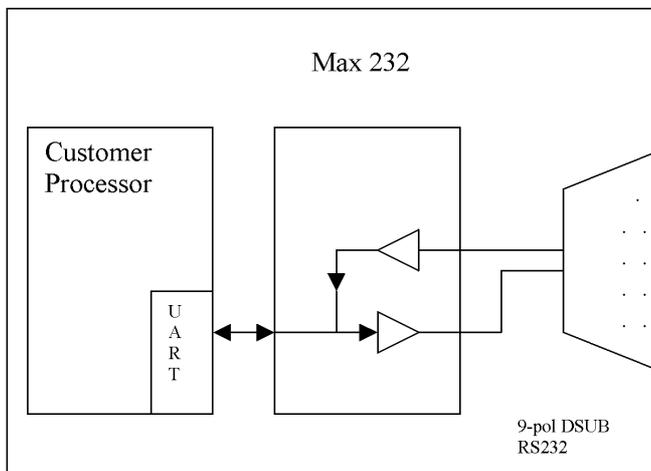
Sende und Empfangsleitungen der seriellen Schnittstelle. diese Schnittstelle ist entsprechend der Beschreibung in Kapitel 4 frei programmierbar.

3.3 Software

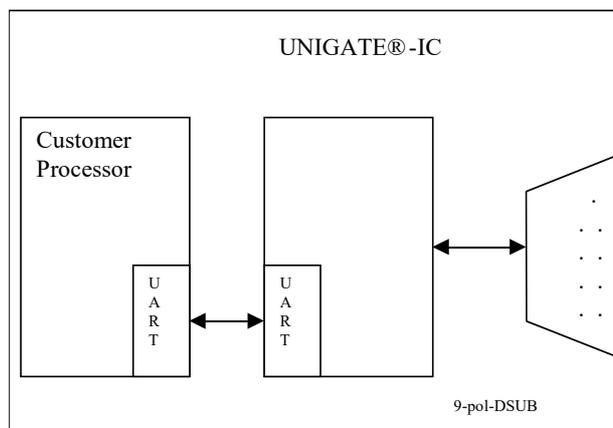
Die Software führt Scriptbefehle aus, diese wiederum steuern die Hardware des IC und wickeln per Software ihr komplettes Protokoll ab. Das Script selbst kann durch die Firma Deutschmann oder mit der Software Protocol Developer von Ihnen erstellt werden. Für eine detaillierte Beschreibung der Scriptbefehle 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 LON Bus zur Verfügung steht. Um die Belegung normgerecht zu halten, ist allerdings ein Hardware-Redesign nötig.



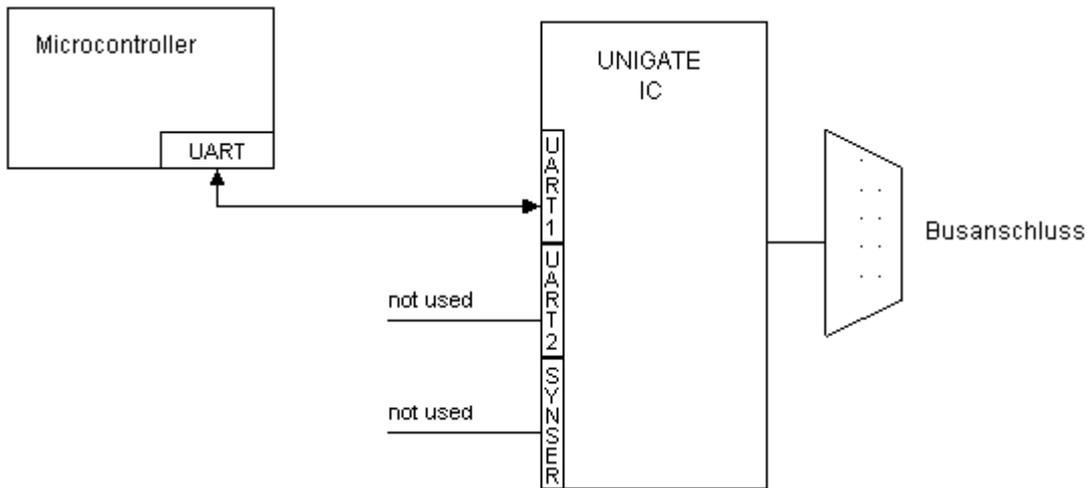
Nach dem Austausch des RS232 Treibers durch das UNIGATE® IC steht an der 9-pol. D-SUB-Buchse der LON Bus zur Verfügung.

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

3.5 Anschlussbeispiele

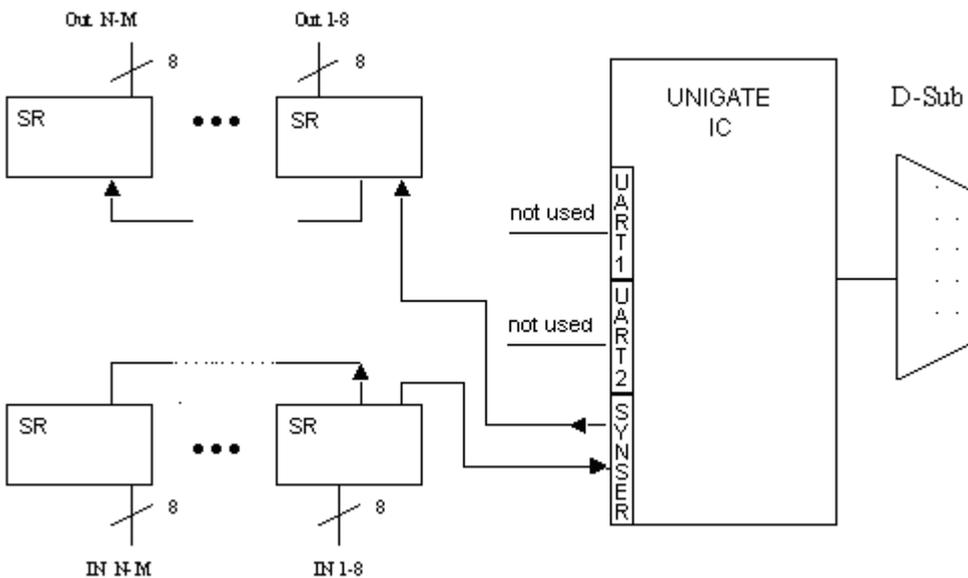
Hier finden Sie einige Hinweise, die Ihnen für ein Redesign helfen. Im folgenden sind einige Varianten aufgeführt, die Ihnen die Entscheidung etwas leichter machen sollen.

Variante 1: Einsatz als reines Koppelmodul für den Bus



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

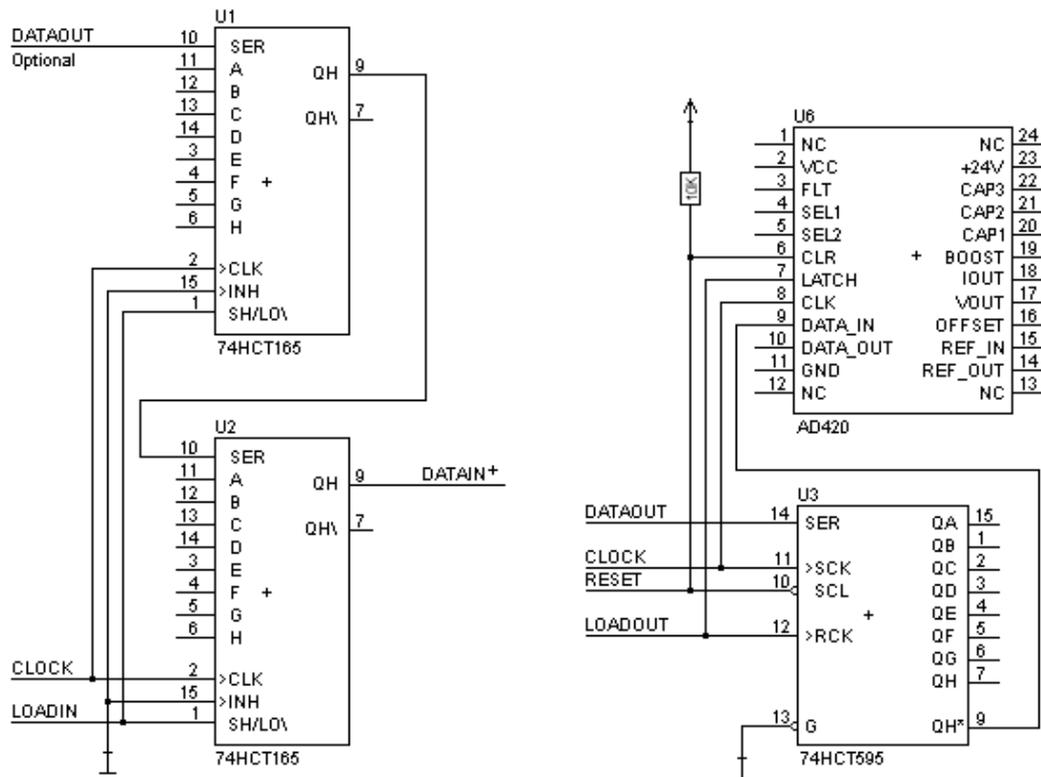
Variante 2: Einsatz des UNIGATE® IC für digitale oder analoge E/A-Module



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.

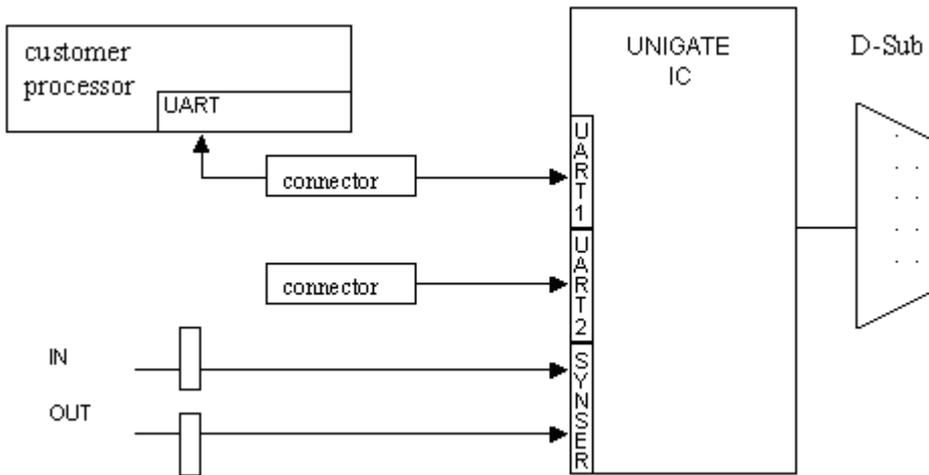


Variante 3: Beispiel für digitale I/Os

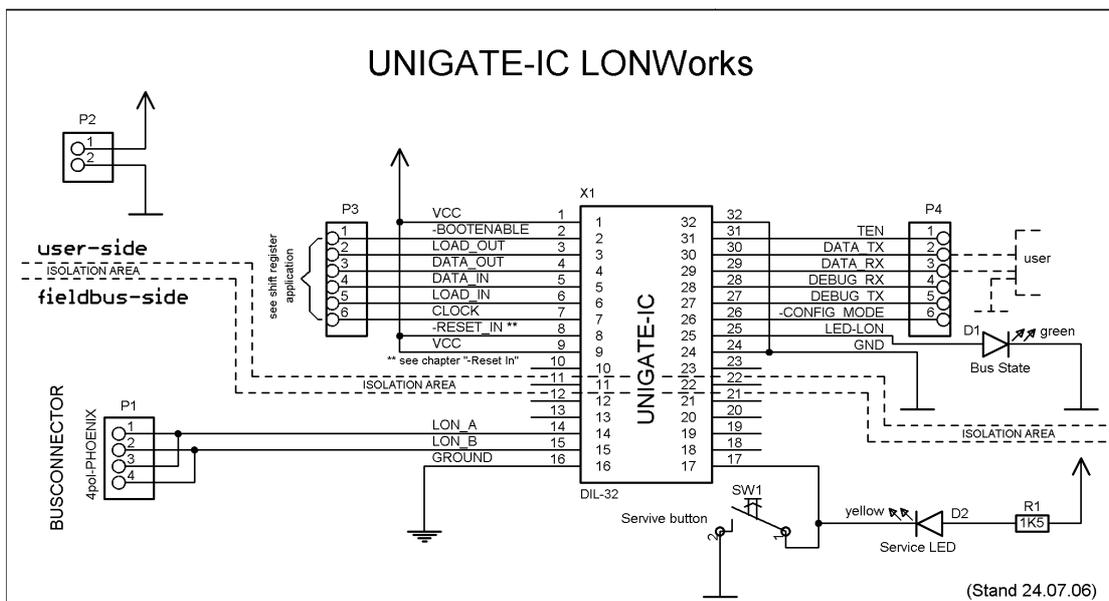
Die serielle synchrone als auch die asynchrone Schnittstelle kann vom UNIGATE® IC zur selben Zeit bedient werden. Hieraus ergibt sich die Möglichkeit, eine bestehende Applikation um zusätzliche digitale oder analoge E/As zu erweitern.

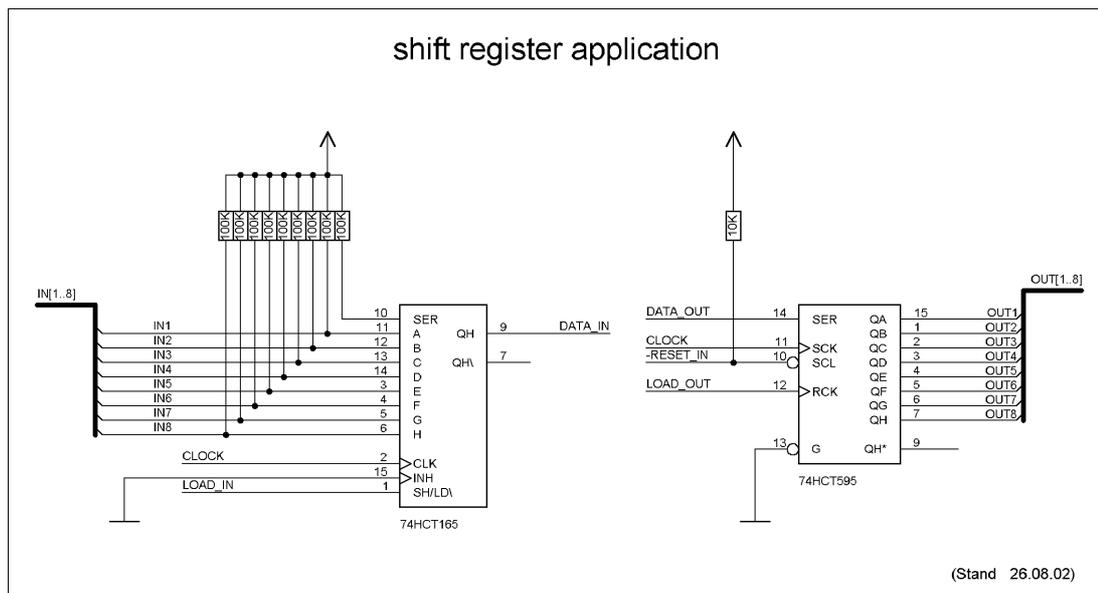
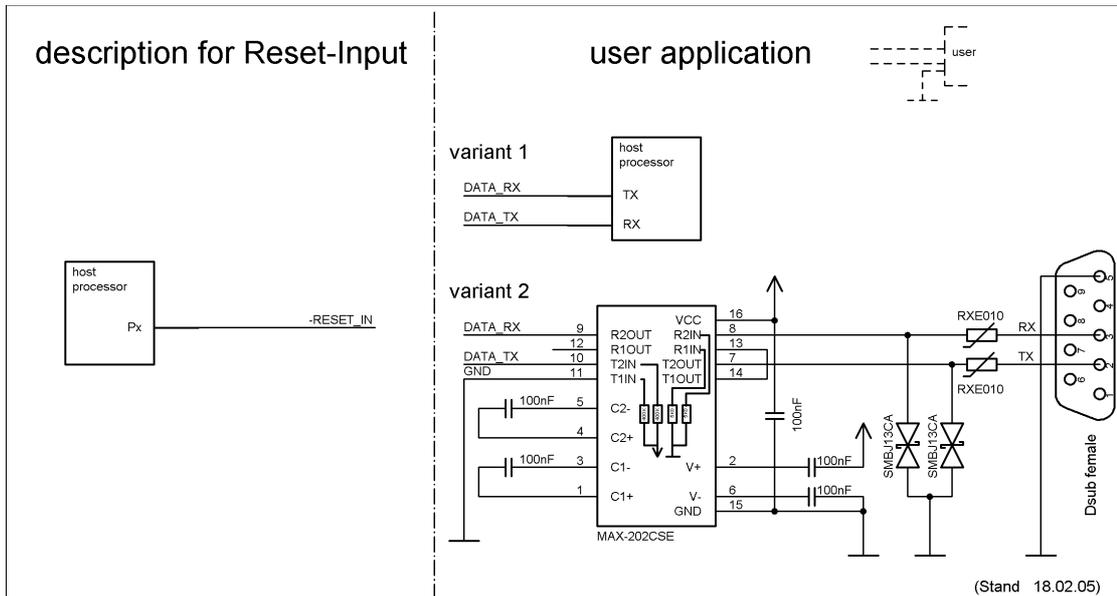
Im Kapitel 5.2 finden Sie ein Beispiel für ein Script, das diese E/As bedient.

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





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 Handling (Befestigung des UNIGATE® IC auf der Trägerplatine)

Generell gibt es zur Montage des UNIGATE® 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® 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® 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® 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.

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 maximale Größe für EA-Daten kann 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.

5 Synchrone Schnittstelle

5.1 Übersicht über die synchrone serielle Schnittstelle

Die synchrone serielle Schnittstelle ist eine Schnittstelle des IC zu getakteten Schieberegistern. Es ist hierüber nicht nur möglich digitale Signale ein oder auszugeben; auch die Ansteuerung von DA oder AD-Wandlern mit getakteter serieller Schnittstelle ist genauso möglich wie die Anbindung von LEDs oder das Einlesen von Drehschaltern. Anschlussbeispiele sind in Kapitel 3 gegeben.

Die synchrone serielle Schnittstelle kann auch in Produkten benutzt werden, die über keinen eigenen Microcontroller verfügen (Stand Alone Betrieb), um z.B. digitale EA-Module zu realisieren. Es können jeweils bis zu 256 Signale für Input und Output verarbeitet werden; für die unterschiedliche Anzahl von Input und Output Signalen zeichnet die Firmware des UNIGATE® IC verantwortlich und übernimmt hierfür die Kontrolle.

Der Datenaustausch mit dem Script erfolgt mit den Befehlen

```
"Set ShiftRegisterInputType / Set ShiftRegisterOutputType" und  
"Set ShiftRegisterInputBitLength / Set ShiftRegisterOutputBitLength"  
sowie "WriteShiftRegister" als auch "ReadShiftRegister".
```

Um den Einsatz der Synchron-Seriellen-Schnittstelle möglichst effizient zu gestalten, kann über das Script ein "Schieberegistertyp" und eine "Bit-Länge" eingestellt werden, wobei beide innerhalb des Scriptes durch Script Befehle dynamisch geändert werden können.

Dadurch wird erreicht, dass die Firmware den gesamten Datentransfer mit der Hardware steuern kann, und der Datenaustausch schnellstmöglich erfolgt.

Zur Zeit sind die Schieberegistertypen "RiseClk_RiseLoad" und "RiseClk_LowLoad" implementiert, die z. B. beim Einsatz des 74595 und 74165 benötigt werden (siehe auch Kapitel 5.2). Andere Typen können hier aber sehr schnell und einfach - durch Deutschmann Automation GmbH - ergänzt werden.

Der Schieberegistertyp bestimmt, mit welchen Flanken bzw. Pegeln Daten zur Hardware ausgegeben bzw. eingelesen werden.

Der Datenaustausch beschränkt sich in diesem Fall auf die Script-Befehle "WriteShiftRegister" und "ReadShiftRegister". Die Taktrate beträgt min. 150kHz, so dass beispielsweise ein Schieberegister mit 32 bit in max. 215µs eingelesen oder geschrieben wird.

5.2 Scriptbeispiel

```
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 an den Ausgängen des Analogwandlers  
// 0x21 an den Ausgängen des Schieberegisters
```

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) ausgegeben. 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, 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 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

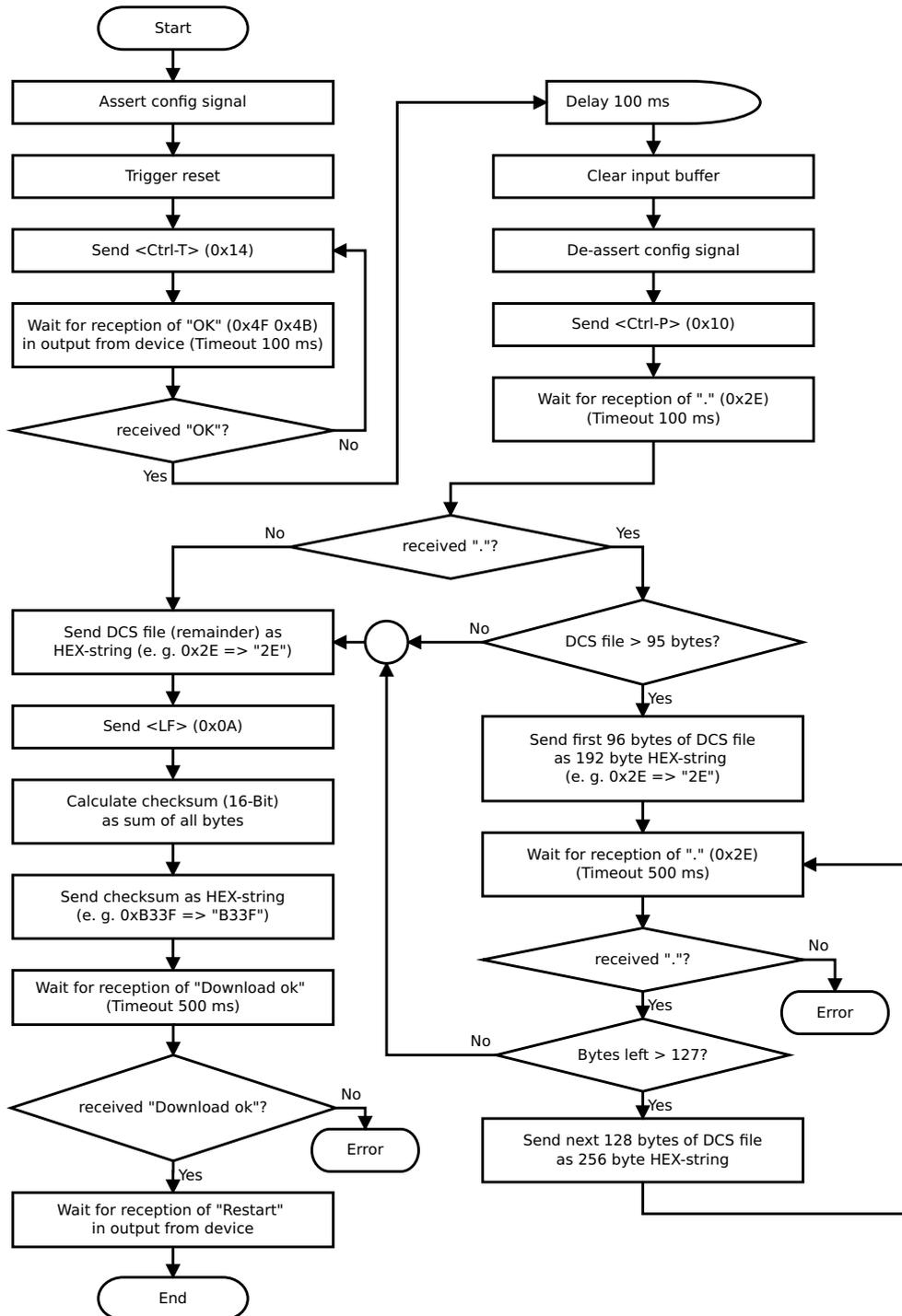
Wird der Pin "ConfigMode" während des PowerUp oder Reset auf GND gezogen, startet das UNIGATE® IC im Konfigurationsmodus. In diesem Modus besteht die Möglichkeit mit dem IC zu kommunizieren, ohne dass die normale Software ausgeführt wird. Man kann in diesem Modus Einstellungen der Standardsoftware des 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-LN(62)-SC V1.2a[26] (c)dA Script(8k)="Test" Author="gst" Version="V1.0t" Date=14.03.2005  
SN=47110001 Neuron-ID=04 2C 87 E9 01 00
```

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 eine 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.

7.4.1 LONWorks

- unterstützt SNVT Master List Revision 12 (ab Firmware V1.22).

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 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.

8.6 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.

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:

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

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

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

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%

8.8 Scriptbearbeitungszeiten

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 Scriptbearbeitung 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 LONWorks

9.1 Allgemeines über UNIGATE® LONWorks – RS232/RS485

Das UNIGATE® LONWorks-Modul kommuniziert nach dem LONTalk-Protokoll. Das bedeutet, dass es mit allen LONWorks-Knoten kommunizieren kann, die diesem Protokoll entsprechen. Weitere Informationen über LONWorks Technologie erfahren Sie unter www.echelon.de und www.lonmark.org

9.2 Knoten

Knoten sind Geräte, die mit physikalisch festen Bedienungsgeschichten interagieren und über ein Netz kommunizieren, das über das LONTalk-Protokoll verfügt. Jeder Knoten im Netz enthält eingelagerte Intelligenz, die das Protokoll realisiert und Steuerfunktionen ausführt (Handhabung der E/A Signale). Es wird keine zentrale Bedienung oder Master-Slave-Konstruktion benötigt und deshalb wird ein LONWorks-Netz oft ein "verteiltes E/A-Netz" genannt.

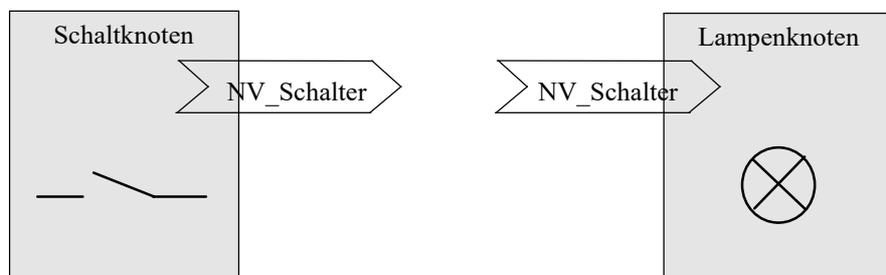
Jeder Knoten beinhaltet eine physikalische Schnittstelle, der Funksendeempfänger dieser Schnittstelle mit dem Kommunikationsmedium. Das UNIGATE®-LON-Modul benutzt das FTT-10A (Free Topology Twisted Pair) von der Echelon Gesellschaft. Dies ist das üblicherweise am meisten benutzte Medium in der Automation und diese Bauart unterstützt die Stern-, Bus- und Schleifenverdrahtung. Der FTT-10A-Funksendeempfänger ist mit dem verdrehten Buskabel verbunden und hat eine Baudrate von 78 kbit/s. Der FTT-10A hat eine hohe Impedanz wenn er nicht an Spannung liegt und unterbricht die Netzkommunikationen nicht, wenn die Betriebsspannung zusammenbricht

9.3 Netzwerkvariablen (NVTs)

Eine Netzwerkvariable ist ein Objekt auf einem Knoten, die mit einer oder mehreren Netzwerkvariablen an einen oder mehrere Knoten verbunden werden kann. Die Netzwerkvariable eines Knotens definiert seine „Eingabe“ und seine „Ausgabe“ von einem Netzwerkstandpunkt und erlaubt die Mitbenutzung der Daten in einer verteilten Anwendung. Wann immer eine Applikation in eine seiner Netzausgangsvariablen schreibt, wird der neue Wert der Netzvariablen über das Netz zu allen Knoten mit den Netzeingangsvariablen verbreitet, die mit dieser Netzausgangsvariablen verbunden wurden. Zum Beispiel: Um das Licht über ein LONWorks Netz anzuschalten, muss ein Schaltknoten seine Netzausgangsvariable mit der Netzeingangsvariable des Lampenknotens logisch verbunden sein. Wenn der Schalter betätigt wird, wird die Netzwerkvariable über das Netz verbreitet und im Lampenknoten empfangen, das Licht wird angeschaltet. Siehe das Bild unten.

Die LONMark Organisation hat mehrere Netzwerkvariablen anerkannt, die verschiedene Arten von standardmäßigen Datenabbildungen darstellen, z. B. Temperatur, Druckkraft, Prozent, Spannung. Sie werden Standard-Netzwerk-Variablen oder SNVT's genannt und sind in der "SNVT Master Liste und dem Programmierhandbuch" von der Echelon Gesellschaft aufgelistet.

Eine SNVT beinhaltet Informationen über Typ, Auflösung und Größe. Die LONMark Vereinigung definiert Objekte, die für eine spezielle Anwendung benutzt werden, als eine Gruppe von SNVTs. Bitte beachten Sie, dass Netzwerkvariablen vom Anwender auch als nicht Standard definiert werden können und gewöhnlich nur Netzwerkvariablen oder NVs genannt werden.



9.3.1 Implementierung der Netzwerkvariablen in das UNIGATE®-LONWorks

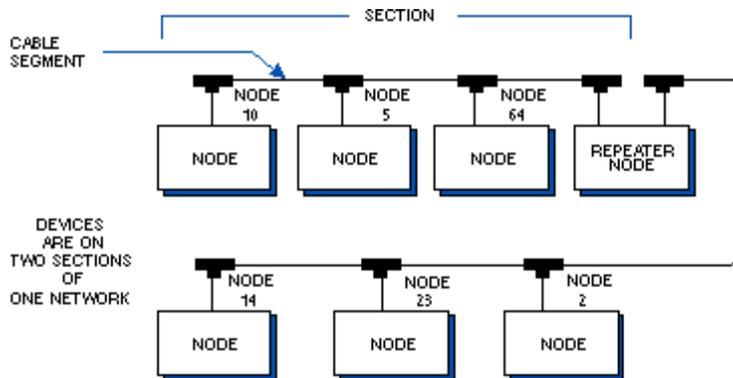
Ein LONWorks-Knoten ist applikationsspezifisch - was bedeutet, dass es unmöglich ist, eine allgemein gültige Gruppe von Netzwerkvariablen für alle Arten von Anwendungen zu haben. Daher muss das UNIGATE®-LON mit einer fixen Gruppe von Netzwerkvariablen vorkonfiguriert werden, die für eine bestimmte Art von Anwendung benötigt werden kann.

9.3.2 Installation

Die Netzvariablen können, mit Hilfe eines Installationswerkzeuges mit den Knoten verbunden werden. Dieses Binding verbindet die Netzwerkvariablen zwischen verschiedenen Knoten. Sobald das Binding fertig ist, kann das Installationswerkzeug aus dem Netz genommen werden. Während der Installationszeit erhält jeder Knoten eine einmalige Netzidentität. Für weitere Informationen über LONWorks, lesen Sie bitte die LONWorks-Technologie Gerätedaten von Motorola oder die LONWorks-Dokumentation von Echelon.

9.3.3 Technische Daten (Merkmale) für LONWorks

Technische Übertragung	Free topology (FTT-10A) twisted pair cable, Übertragungsrate 78 kbits/s
Max. Buslänge	Bus bis zu 2000 m (mit Busverstärkern)
Max. Knoten-zu-Knoten Distanz	500 m
Max. Anzahl von Knoten	32.385 Knoten eingeteilt in 127 Knoten/Subnetz 255-Subnetze/Domäne
Arten von Daten	Peer to peer Kommunikation



Das UNIGATE® IC LONWorks unterstützt max. 62 SNVTs.

Unterstützte SNVTs:	SNVT Master List
Bis Firmware V1.21 -> No. 1 - 166	Version 11
Ab Firmware V1.22 -> No. 1 - 176	Version 12

9.4 Scriptbeispiel für die Initialisierung des LONWorks

```
//***** LON init
//*****
// Type No from SNVT Master-List
// http://www.tsd.org/ieee1473/SNVT.pdf [LON_SNVT_MasterList_-
// May2002]
// UG62 max count of SNVT's = 62 In and Out SNVT's together
// UG512 max count of SNVT's = 256 In and 246 Out-SNVT's

// 36=31byte, 37=31byte, 83=2byte, 88=29byte, 93=6byte =>99byte
// set In SNVT's
var InMapTable: buffer[5];          moveConst( InMapTable[0],
#36#37#83#88#93 );
var InMapsize : word;              moveConst( InMapsize, 5); // count of
In SNVT's

// set Out SNVT's
var OutMapTable: buffer[5];        moveConst( Out-
MapTable[0],#36#37#83#88#93 );
var OutMapsize : word;             moveConst( OutMapsize, 5); // count of
Out SNVT's

// init SNVT's
LonInMapping( InMapTable[0], InMapsize);
LonOutMapping(OutMapTable[0], OutMapsize);

// set SelfDocString
```

```
var SelfDocBuffer: buffer[20];   moveConst( SelfDocBuffer[0], "Vorlage
LON SC");
var SelfDocsize : word ;        moveConst( SelfDocsize, 14);
// init SelfDocString
LonSelfDocString( SelfDocBuffer[0], SelfDocsize);
// note: new init will activate only if ProgramID or SNVT types are
changed,
// like Configmode or Testmode. In this case also bindings will be lost!

// define ProgramID, default = SC000000 SD... for UG512 V1.82 with new
LON-module
Set ( LONProgramID , 123456 ) ; // SC123456, SC = normal mode
                                // CM006969 Configmode
                                // TM004711 Testmode

// define BusMethode for fieldbusIN
Set ( BusdataTriggerMethod , DataChange ) ; // newfieldbus IN data
only if changed
moveconst (OnChange, 1); // 1 = TRUE, 0 = FALSE
if OnChange equal TRUE then :LonInitEnd;

Set ( BusdataTriggerMethod , BusEvent ) ; // new fieldbus IN data on
BusEvent
// note: BusEvent - you receive the same data and you can read out from
wich index
:LonInitEnd;
//***** LON init end
*****
Set ( RS_State_LED , StaticRed ) ;
//***** start fieldbus
*****
BusStart ;
Wait ( Bus_Active ) ;

Get ( BusInputLen16 , wBusInSize ) ;
Get ( BusOutputLen16 , wBusOutSize ) ;

Set ( RS_State_LED , RedGreenFlashing ) ;
//*****
main*****
:start ;

Get ( BusDataChanged , bDestination ) ;// sind geänderte Busdaten
gekommen ?
```

```
if bDestination equal b0 then :CheckRS_In;
```

```
ReadBus ( aBusInBuf[0] , wBusInSize ) ;
```

Diese Routine ist ein Auszug aus dem Beispiel „LON_SC.dss“, das Sie im Protocol Developer unter dem Menü „New“ finden.

Weitere LON Script Beispiele befinden sich im Ordner Examples.

9.5 LON-spezifische Parameter

```
// Bus specific -
// LON
// Lon Input SNVT Mapping
    LonInMapping ( a_Source[0] , w_NumberChar ) ;
// Lon Output SNVT Mapping
    LonOutMapping ( a_Source[0] , w_NumberChar ) ;
// Lon Self Documentation String
    LonSelfDocString ( a_Source[0] , w_NumberChar ) ;

Set ( BusdataTriggerMethod , BusEvent ) ; // DataChange

// BusStart:
    BusStart ;
// Device Control -
// Wait State:
    Wait ( Bus_Active ) ;
//
*****
*****

Get ( AvailableBusData , w_word ) ;           // - 0..BusInputSize
Get ( BusDataChanged , b_byte ) ;

// Read Busdata:
    ReadBus ( a_Destination[0] , w_NumberChar ) ;
// Write Busdata:
    WriteBus ( a_Source[0] , w_NumberChar ) ;
// Wait for changing Busdata:
    WaitBusDataChanged ( w_Timeout , w_NumberChar ) ;

// Get Parameter:                               Set Parameter
// =====                                     =====
Get ( BusInputSize , b_byte ) ;                 // 0..255
Get ( BusOutputSize , b_byte ) ;               // 0..255
```

```
// supported from ScriptRev 26 and higher
Get ( BusInputLen16 , w_word ) ;           // 0..512
Get ( BusOutputLen16 , w_word ) ;         // 0..512

//
.....
.....
// LON only:
// -----
Get ( LONProgramID , l_long ) ;           Set ( LONProgramID , 0 ) ; // 0
.. 999999
Get ( CHANGED_SNVT_INDEX , wDestination ) ;
SetByVar ( CHANGED_SNVT_INDEX , wDestination ) ; // -1..255 (UG512)
                                                    // -1..61 (UG 62)

//
*****
*****
```

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
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 26) 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 45).

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

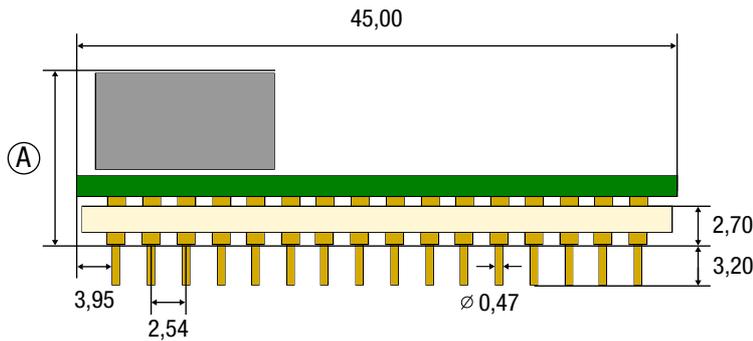
Im Konfigurationsmodus wird das UNIGATE® IC nach dem Start (oder nach einem Reset) einen Selbsttest ausführen und nach erfolgreichem Selbsttest auf weitere Anweisungen warten. Hier ist es möglich, ein übersetztes Script in das Gerät zu laden oder auch den Firmware Download Mode zu initiieren.

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



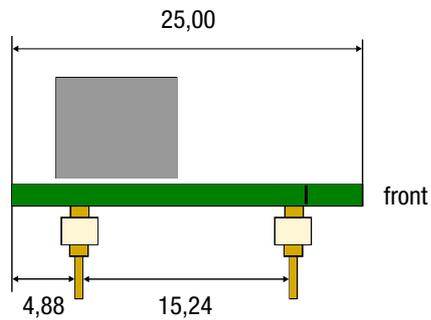
UNIGATE® IC

	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	PROFINET 2Port (wt*)	RS
A= Gesamthöhe (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[®] ICs sind im Rastermaß 2,54 mm angeordnet.



DIP-Spacing Code 6

Wenn Sie auch an den Einsatz von anderen UNIGATE[®] IC Varianten denken, müssen Sie eine max. Bauhöhe (inkl. Stifte) von ≤ 20 mm berücksichtigen.

12.2 Technische Daten UNIGATE® IC-LONWorks

Merkmale	Erläuterungen
Betriebsspannung	5 V ± 5 %, max. 75 mA DC
Verpolungsschutz Spannungsversorgung	Nein
Schnittstelle	2 UART Schnittstellen, 1 synchron serielle Schnittstelle
Feldbusseite	Standard FTT-10A
Programm-ID Feldbus-Baudrate	Einstellbar über Script Standard
UART-Baudrate	Bis 625 Kbaud (einstellbar über Script)
Feldbusdatenformat	Max. 62 SNVTs, max. 512 Byte
Technologie	FT 3120
Sonstiges	Extern können z. B. I/Os, analoge Signale, Schieberegister, LEDs, Schalter etc. angeschlossen werden
Abmessungen	45 x 25 x 13 mm (B x T x H), alte Hardware Rev.: 45 x 25 x 20 mm (B x T x H)
Montage	32 DIL
Gewicht	ca. 10 g
Betriebstemperatur	-40°C ..+85°C
Lager-/Transporttemperatur	-40°C..+100°C
Einbaulage	Beliebig

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.

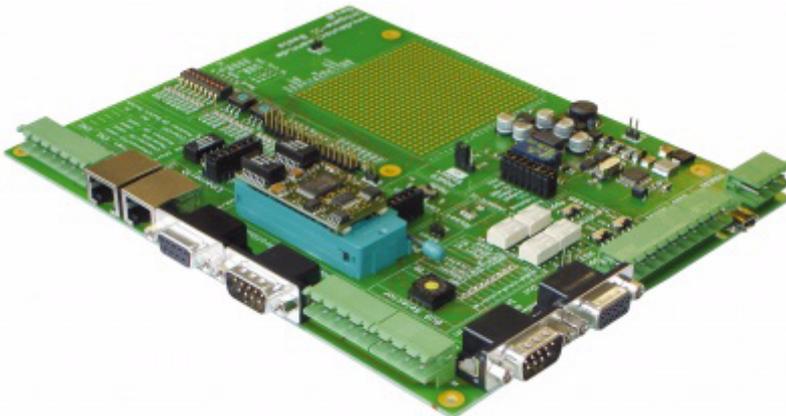
13.3 Developer Kit UNIGATE® IC-AB

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

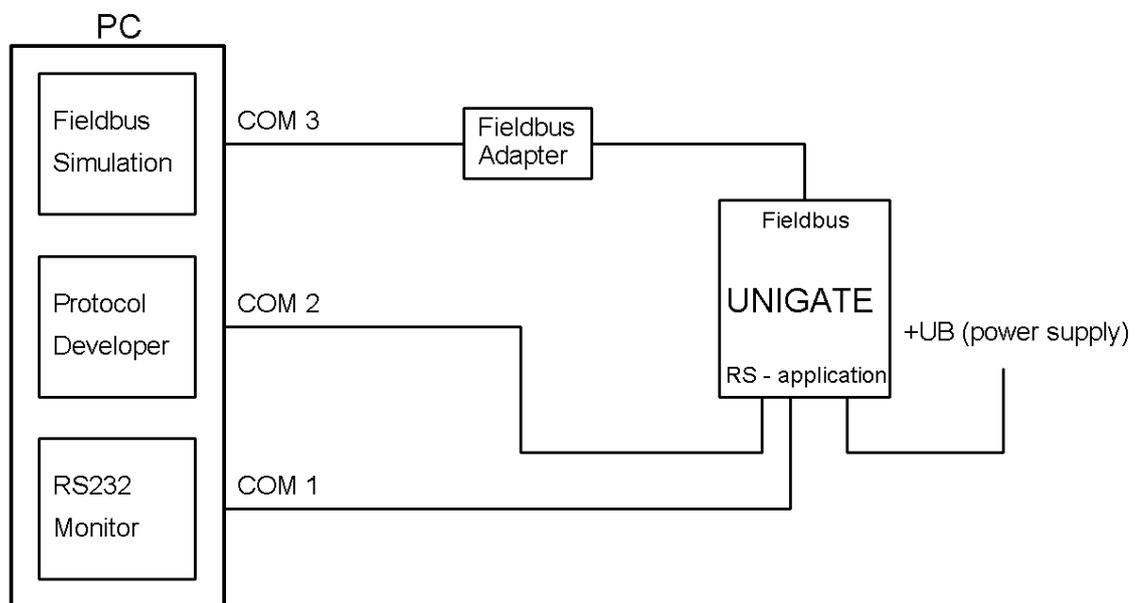
Das Developer Board wurde entwickelt, um die schnelle Implementierung des Deutschmann All-In-One Bus Node UNIGATE® IC in Ihre Elektronik zu gewährleisten. Das Board ist für alle von Deutschmann unterstützen Feldbusse und Industrial Ethernet Busse geeignet.



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 -> „LON_IC_Basisboard_UG62.dss“).

Schalterstellung: Nur für Bus-Adapter SLTA-10 von Echelon: 1 - 3 ↓
4 - 8 ↑

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	=	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

15 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 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

15.1 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.

15.2 Download von PC-Software

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

<http://www.deutschmann.de>

