

**Deuschmann**

*your ticket to all buses*

**Instruction Manual  
Universal Fieldbus-Gateway  
UNIGATE® IC2 - Fast Ethernet**



**Deuschmann Automation GmbH & Co. KG**  
[www.deuschmann.com](http://www.deuschmann.com) | [wiki.deuschmann.de](http://wiki.deuschmann.de)





<b>1</b>	<b>General introduction</b>	<b>10</b>
<b>2</b>	<b>UNIGATE® IC</b>	<b>11</b>
2.1	The UNIGATE® IC2 model series	11
2.2	Software	11
2.3	Asynchronous Serial Interface (UART)	11
2.4	Synchronous serial interface (SSI/SPI)	11
2.5	The Debug-interface	11
2.6	UNIGATE® IC2 hardware survey	12
<b>3</b>	<b>Hardware design.</b>	<b>13</b>
3.1	Ports	13
3.2	Pinout	13
3.2.1	-Boot enable	14
3.2.2	Load out (SPI-Master: SS0-)	14
3.2.3	Data out (SPI-Master: SS1-)	14
3.2.4	Data In (SPI: MISO)	14
3.2.5	Load In (SPI: MOSI)	14
3.2.6	Clock (SPI: SCK)	14
3.2.7	-Reset In	14
3.2.8	LED-LINKACT	15
3.2.9	Error-LED	15
3.2.10	LED-FE	15
3.2.11	-Config Mode	15
3.2.12	DbgTX, DbgRx.	15
3.2.13	TE	15
3.2.14	TX, RX.	15
3.3	Basic line of proceeding	16
3.4	Connection examples	17
3.5	Layout examples	19
3.6	Handling (mounting the UNIGATE® IC2 on the carrier board)	21
3.7	Requirement for Ethernet-Interface	22
<b>4</b>	<b>The serial interface</b>	<b>23</b>
4.1	Overview	23
4.2	Initialization of the serial interface	23
4.3	Use of the serial interface	23
4.4	Further operation modes	23
4.5	UART	23
<b>5</b>	<b>Synchronous serial interface</b>	<b>24</b>
5.1	Shift register operation	24
5.1.1	Example-Script	24
5.2	SPI mode	25

---

5.2.1	Example-Script . . . . .	25
<b>6</b>	<b>The Debug-interface . . . . .</b>	<b>26</b>
6.1	Overview of the Debug-interface . . . . .	26
6.2	Starting in the Debug-mode . . . . .	26
6.3	Communication parameter for the Debug-interface . . . . .	26
6.4	Possibilities with the Debug-interface . . . . .	26
6.5	Commands of the Debug-interface . . . . .	26
<b>7</b>	<b>Operating modes of the IC2. . . . .</b>	<b>27</b>
7.1	Standard operating mode . . . . .	27
7.2	Configuration mode . . . . .	27
7.3	Firmware update mode . . . . .	27
<b>8</b>	<b>Script and configuration . . . . .</b>	<b>28</b>
8.1	Overview . . . . .	28
8.2	The configuration mode . . . . .	28
8.3	Update the script . . . . .	28
8.4	Configuration of the UNIGATE® IC2 . . . . .	31
8.4.1	Ethernet . . . . .	31
8.4.2	RS232/RS485/RS422 . . . . .	31
<b>9</b>	<b>Generating a Script . . . . .</b>	<b>32</b>
9.1	What is a Script? . . . . .	32
9.2	Memory efficiency of the programs . . . . .	32
9.3	What can you do with a Script device? . . . . .	32
9.4	Independence of buses . . . . .	32
9.5	Further settings at the gateway . . . . .	33
9.6	The use of the Protocol Developer . . . . .	33
9.7	Accuracies of the baud rates at UNIGATE® IC2 . . . . .	33
9.8	Script processing times . . . . .	33
<b>10</b>	<b>Ethernet interface . . . . .</b>	<b>35</b>
10.1	Communication protocols . . . . .	35
10.1.1	TCP or UDP? . . . . .	35
10.1.2	The UNIGATE® as TCP server . . . . .	35
10.1.3	TCP timeout . . . . .	35
10.1.4	MODBUS/TCP . . . . .	35
10.2	Data exchange mode . . . . .	35
10.3	Web server . . . . .	36
10.4	FTP-server . . . . .	36
10.4.1	Script update via FTP . . . . .	36
10.4.2	System configuration update via FTP . . . . .	36
10.4.3	File system. . . . .	36
10.5	Setting the IP address . . . . .	36

10.6	Zero Configuration	37
10.7	Comparison with former UNIGATE® FE	37
10.7.1	UDP channel mode	37
10.7.2	Raw-channel mode	37
10.7.3	BusStatus	37
<b>11</b>	<b>Error handling at UNIGATE® IC2</b>	<b>39</b>
<b>12</b>	<b>Firmware-update</b>	<b>40</b>
12.1	Overview	40
12.2	Adjusting the firmware-update-mode	40
12.2.1	Adjustment by hardware	40
12.2.2	Adjustment by software	40
12.3	Execution of the firmware-update	40
12.4	Safety note	40
<b>13</b>	<b>Technical data</b>	<b>41</b>
13.1	Mechanics of the UNIGATE® IC2	41
13.1.1	General dimensions of UNIGATE® IC2	41
13.2	Technical data UNIGATE® IC2 - Fast Ethernet	43
<b>14</b>	<b>Accessory</b>	<b>44</b>
14.1	FirmwareDownloadTool (FDT)	44
14.2	Protocol Developer	44
14.3	Developer Kit UNIGATE® IC-AB IC	44
14.3.1	Developer Board UNIGATE® IC-AB	44
14.3.2	Quick start	45
<b>15</b>	<b>Appendix</b>	<b>46</b>
15.1	Explanations of the abbreviations	46
<b>16</b>	<b>Servicing</b>	<b>48</b>
16.1	Returning a device	48
16.2	Downloading PC software	48



#### Disclaimer of liability

We have checked the contents of the document for conformity with the hardware and software described. Nevertheless, we are unable to preclude the possibility of deviations so that we are unable to assume warranty for full compliance. The information given in the publication is, however, reviewed regularly. Necessary amendments are incorporated in the following editions. We would be pleased to receive any improvement proposals which you may have.

#### Copyright

Copyright (C) Deutschmann Automation GmbH & Co. KG 1997 – 2022. All rights reserved. This document may not be passed on nor duplicated, nor may its contents be used or disclosed unless expressly permitted. Violations of this clause will necessarily lead to compensation in damages. All rights reserved, in particular rights of granting of patents or registration of utility-model patents.



## 1 General introduction

In the past the integration of a fieldbus connection required an enormous effort from the progress engineers. On account of the large variety of communication systems it is not enough to compile the right combination of communication hardware; due to their standards and fundamentals different busses also require the corresponding skills of the engineers.

This does not apply in case of the UNIGATE<sup>®</sup> IC2 by Deutschmann Automation any more. All digital functions, software, stack and driver as well as optocoupler are integrated on a UNIGATE<sup>®</sup> IC2 in correspondence with the standard. In addition to the reduction of the required size, also different fieldbuses can easily be integrated.

**Through the flexible firmware of UNIGATE<sup>®</sup> IC2 no software-changes are required on the side of the customer!**

Since 1997 Deutschmann Automation has experience in the field of fieldbus gateways; this enormous experience results in the UNIGATE<sup>®</sup> IC as a consistent sequel of this successful product line.

### Terminology

In the entire document and in all parts of the software that is to be used, the terms Input and Output are used. Input and Output are ambiguous, always depending on the viewpoint. We see the fieldbus as central interface and as integral component of your device; therefore in all places it is always referred to data from the viewpoint of the Slave, that is Input data, as data from the Master to the Slave - regardless of the used bus.

### Representation of numbers

Numbers in decimal format are always represented without prefix and without suffix as well. Hexadecimal numbers are always marked with the prefix 0x.

The UNIGATE<sup>®</sup> IC2 is a further development of the 1st series of the UNIGATE<sup>®</sup> IC. Equipped with a Cortex-M4 controller, the UNIGATE<sup>®</sup> IC2 achieves much higher transfer rates when communicating via SPI or UART. In addition, the scripting processing time (depending on the commands used) is reduced by a factor of about 50 - 80. Thus, script execution times in the microsecond range are possible. Execution time of a script line is about 10  $\mu$ s.

The UNIGATE<sup>®</sup> IC2 is also PIN-compatible with the 1st series of the UNIGATE<sup>®</sup> IC.

## 2 UNIGATE® IC

The UNIGATE® IC2 integrates all components necessary for the communication (hardware drivers, protocol stacks, etc.). Besides the UNIGATE® IC2 only connectors have to be integrated. The communication is defined flexibly and application-specific with a Script.

### 2.1 The UNIGATE® IC2 model series

UNIGATE® IC2 is based on the pinout of the successful UNIGATE® IC series and thus allows an easy migration. The continuity of the pinout on the application side allows the easy extension of the own product range by further industrial interfaces. A current list of available fieldbus and Industrial Ethernet versions can be found at <http://www.deutschmann.com>.

### 2.2 Software

The device firmware encapsulates the required protocols, communication stacks and interface drivers. There is no need for time-consuming software development. Device behavior and application-side communication are defined using a simple script language.

The script is created with the freely available software Protocol Developer. Deutschmann also offers script development as a service. The Protocol Developer software contains the documentation of the Script commands and many Script templates and examples.

Scripts developed for UNIGATE® IC can be reused with UNIGATE® IC2.

### 2.3 Asynchronous Serial Interface (UART)

Microcontroller-based systems generally have an asynchronous serial interface (UART) with TTL levels. This can be directly connected to the UNIGATE® IC2. Further information on this serial interface can be found in chapter 4.

### 2.4 Synchronous serial interface (SSI/SPI)

In addition to the asynchronous serial interface there is also the possibility of synchronous data input and output. Digital and analog signals can be integrated via shift register modules, IO expanders and DA/AD converters either in shift register or SPI mode.

For synchronous IOs a maximum of 256 signals can be used (256 bit). Circuit examples can be taken from chapter 3.5 and software examples from chapter 5.

Simple IO devices can be realized without any additional microcontroller.

### 2.5 The Debug-interface

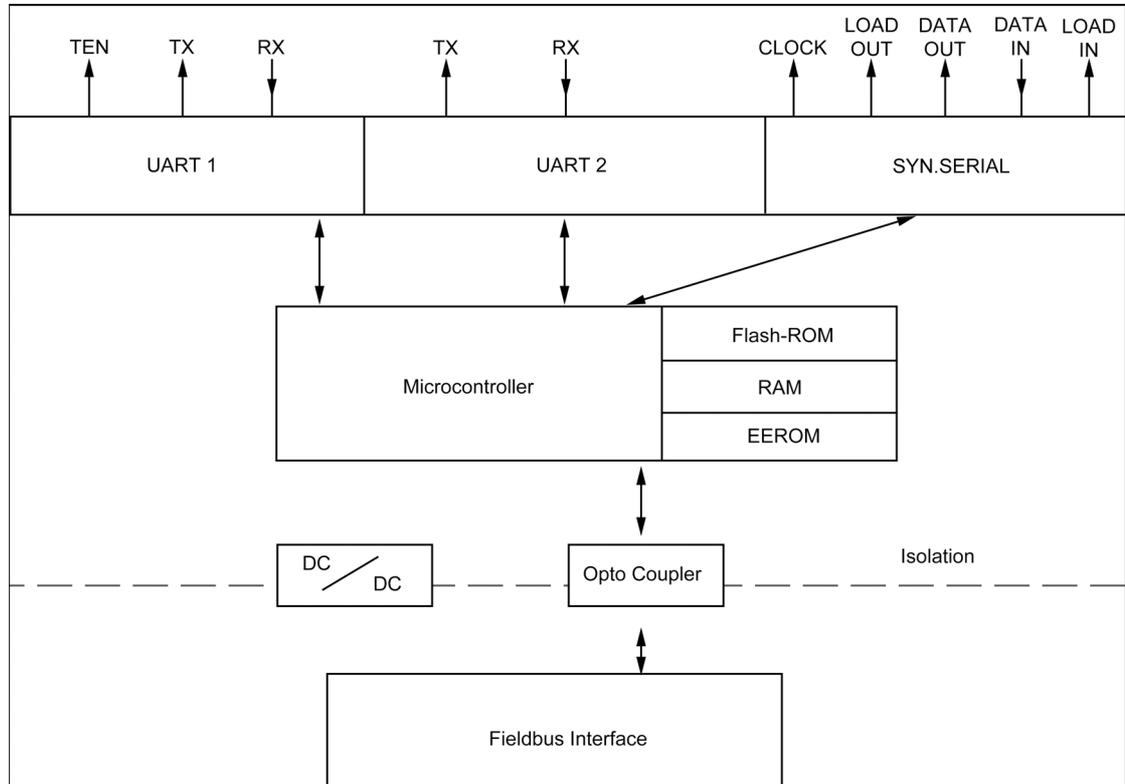
The UNIGATE® IC2 has a debug interface, which allows to execute a Script step by step and also to observe or manipulate data. This is essential for the development of a Script. Normally a script is developed with the Protocol Developer software. Details can be found in the Protocol Developer manual.

All interfaces can be used independently at the same time.

**All interfaces can be used independently at the same time.**

## 2.6 UNIGATE® IC2 hardware survey

The hardware of the UNIGATE® IC2 consists of some few standard components. The picture below shows the functional structure of the IC2.



### 3 Hardware design

This chapter gives basic advise, that is required in order to load UNIGATE® IC2 into your own hardware designs. In the following all ports of UNIGATE® IC2 are described in detail.

#### 3.1 Ports

UNIGATE® IC2 features 32 pins in its layout as a DIL 32 component. Pin 10 - 12 and 21 - 23 as well are not wired due to the electrical isolation. The exact mechanical dimensions can be taken from chapter 13.



In the layout boreholes for ALL 32 pins have to be planned.

#### 3.2 Pinout

Pin	Technical specifications	Name	Description	Remark
1	3,3V ± 5% < 210mA	Vcc	+ 3,3 V voltage supply	
2	IN <sub>Logic</sub>	-BE	boot enable	
3	OUT <sub>Driver</sub>	Load out (SS0-)	strobe signal for synchronous, serial interface	
4	OUT <sub>Driver</sub>	Data out (SS1-)	output data for synchronous, serial interface	
5	IN <sub>Logic</sub>	Data in (MISO)	input data of the synchronous, serial interface	internal pull-up
6	OUT <sub>Logic</sub>	Load in (MOSI)	input data of the synchronous, serial interface; alternatively strobe signal of the output data	internal pull-up
7	OUT <sub>Driver</sub>	Clock (SCK)	clock pulse signal for synchronous, serial interface	
8	IN <sub>Reset</sub>	-Reset in	reset-input of the IC2	
9	connected to pin 1	Vcc	+ 3,3 V voltage supply	
10			connected to Pin 18	
11			connected to Pin 17	
12	nc	nc	not connected	
13	according to norm	RJ45_6	Ethernet-signal RD-	galvanically isolated insulation voltage 1000 Vrms
14	according to norm	RJ45_3	Ethernet-signal RD+	galvanically isolated insulation voltage 1000 Vrms
15	according to norm	RJ45_2	Ethernet-signal TD-	galvanically isolated insulation voltage 1000 Vrms
16	according to norm	RJ45_1	Ethernet-signal TD+	galvanically isolated insulation voltage 1000 Vrms
17	according to norm	LED-LINKACT	Ethernet linkpulses found / network data traffic	from hardware rev. B on connection see chapter 3.6
18	according to norm	Error-LED	anode of the red LED (currently not supported)	(not supported yet) connection see chapter 3.6
19	according to norm	nc	not connected	
20	according to norm	Ground (PE)	ground	from hardware rev. B on
21	nc	nc	not connected	
22	nc	nc	not connected	
23			connected to Pin 20	
24	connected to pin 32	GND	ground supply voltage of the IC2	
25		LED-FE	bus status LED (anode of the green LED)	
26	IN <sub>Logic</sub>	-Config Mode	signal to start the configuration mode	internal pull-up
27	OUT <sub>Logic</sub>	DbgTX	serial Debug TX	
28	IN <sub>Logic</sub>	DbgRX	serial Debug RX	internal pull-up

29	IN <sub>Logic</sub>	RX	serial data RX	internal pull-up
30	OUT <sub>Logic</sub>	TX	serial data TX	
31	OUT <sub>Logic</sub>	TE	transmit enable	
32*	GND	GND	ground supply voltage of the IC2	

The Ethernet signals are galvanically isolated. The insulation voltage is 1000 Vrms.

	V <sub>IL</sub>	V <sub>IH</sub>
IN <sub>Reset</sub>	< 0,7V	>2,4V
IN <sub>Logic</sub>	< 1V / 5μA	>2,3V / 5μA
	V <sub>OL</sub>	V <sub>OH</sub>
Out	< 0,45V / 4mA	> 2,8V / 4mA

### 3.2.1 -Boot enable

The IC2 is started in the firmware update mode with the level GND during the power up process. See also chapter 12.

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

Strobe signal for the synchronous serial interface. With the positive edge at this output data is taken from the connected shift registers to the physical outputs. In SPI mode this Pin serves as a low-active Slave-Select-Signal.

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

On this line data is output on the synchronous serial interface. The most significant bit of the data is output first. In SPI mode this Pin serves as a low-active Slave-Select-Signal.

### 3.2.4 Data In (SPI: MISO)

Data is read in on the synchronous serial interface via this signal. The most significant bit of the data is expected first. In SPI mode this Pin serves as data transfer from Slave to Master.

### 3.2.5 Load In (SPI: MOSI)

This pin is the strobe signal for the input data of the synchronous serial interface. In SPI mode this Pin serves as data transfer Master to Slave.

### 3.2.6 Clock (SPI: SCK)

This signal is the clock line for the synchronous serial interface. That signal is equally valid for data input and data output.

### 3.2.7 -Reset In

- A reset generator is on board; with it in the normal case the reset input is not required. In this case the reset input has to be connected with VCC, in order to avoid interferences (see chapter 3.6).
- If the customer's application has to initiate a reset of the UNIGATE® IC2, then the reset input can also be connected with a reset output of the customer's application instead of connecting it with VCC. Here all specifications of the reset signal, mentioned in chapter 3.2 have to be kept. The reset-impulse is supposed to last at least 10 μs.

### 3.2.8 LED-LINKACT

A green LED can be connected to that line (see chapter 3.5). The LED indicates whether there is an Ethernet connection. (Link pulses are received). It flashes when there is data traffic in the network (activity) No difference between 10 and 100 MBit/s is made.

Off	No Ethernet connection
On	Established Ethernet connection

### 3.2.9 Error-LED

This line is not supported yet. (applies to Hardware revision C and <)

### 3.2.10 LED-FE

For example, a green LED can be connected to the line. (see chapter 3.6).

Off	Command Bus-start not executed yet
Flash 1 Hz	Ethernet - blink test (via Software UNIGATE-Scan)
Flash 2 Hz	UNIGATE started in configuration mode.
On	Command Bus-start carried out

### 3.2.11 -Config Mode

If the pin has the level GND, then the IC2 starts in the configuration mode.

### 3.2.12 DbgTX, DbgRx

They are transmission line and receive line as well of the IC2's Debug-interface. For the function description of the Debug-interface see chapter 6.

### 3.2.13 TE

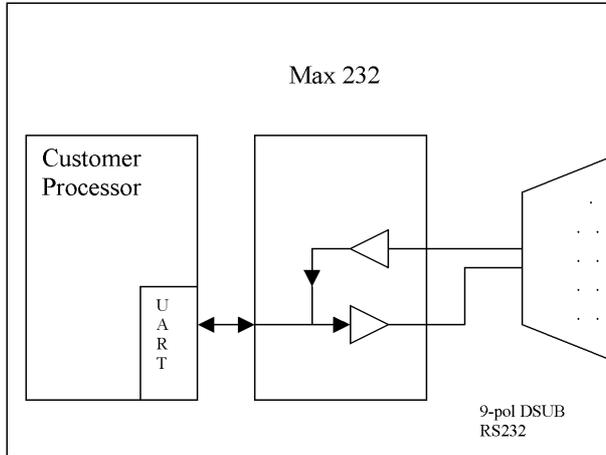
The transmit enable signal allows the connection of RS485 drivers to the IC2's serial interface. The signal is set to High whenever the IC2 sends via the line TX.

### 3.2.14 TX, RX

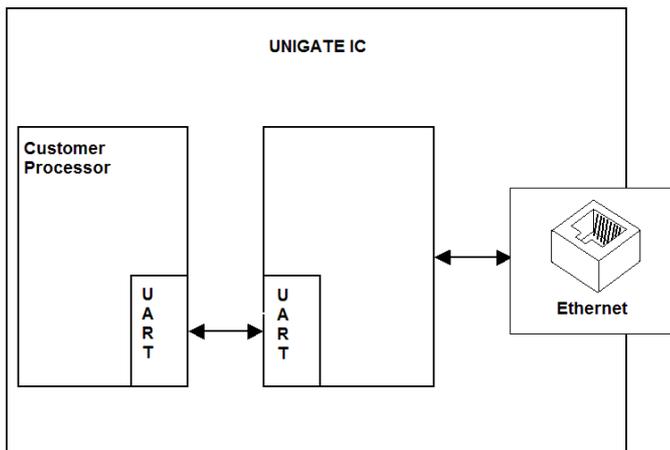
Transmission and receive line of the serial interface. This interface is programmable in accordance with the description in chapter 4.

### 3.3 Basic line of proceeding

In theory it is enough to replace the RS232-driver that is included in your application by the UNIGATE<sup>®</sup> IC2.



Your device, which on the whole is supposed to be assembled as shown above, will now be modified in a way that the Ethernet is available at the 9-pol. socket. However, a hardware redesign is necessary in order to keep the assignment in standard form.



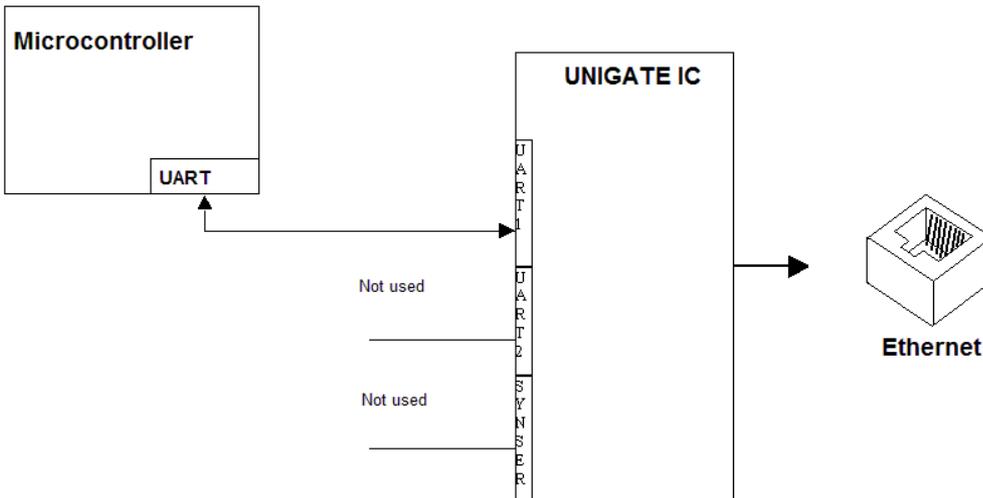
After the RS232-driver has been replaced by the UNIGATE<sup>®</sup> IC2, Ethernet is available at an RJ45-socket.

Deutschmann Automation is also offering an appropriate adapter board. With it existing devices can be adapted without re-design; see chapter 14.

### 3.4 Connection examples

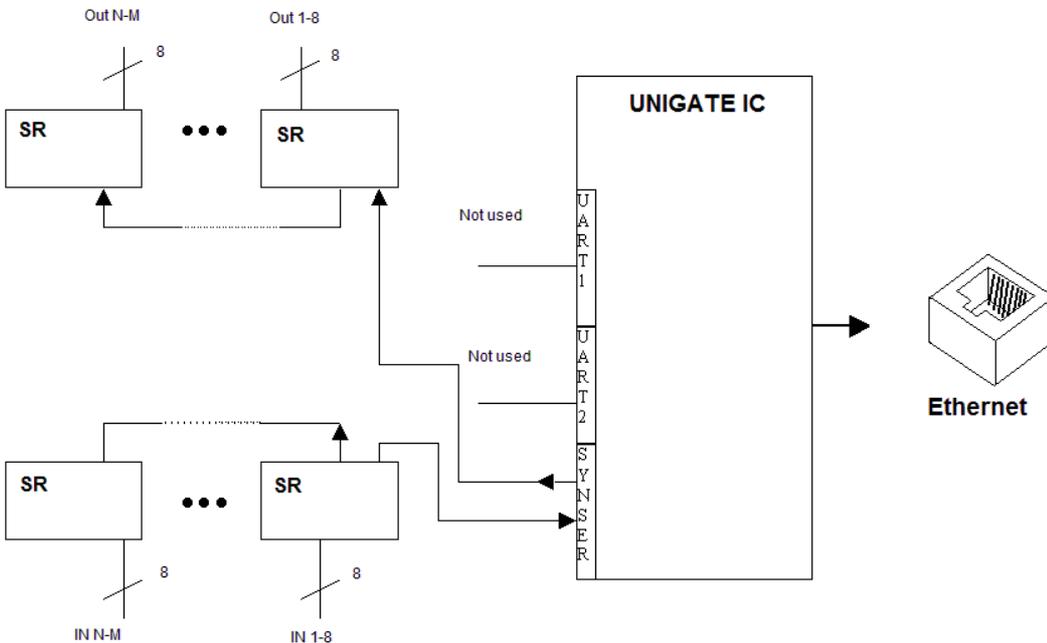
Here you will find some advice that offers help for a re-design. In the following several versions are listed, that should make it easier for you to decide.

Version 1: Use as a pure link module for the bus



The UNIGATE® IC2 independently processes the communication with the customer's device via the TTL-interface.

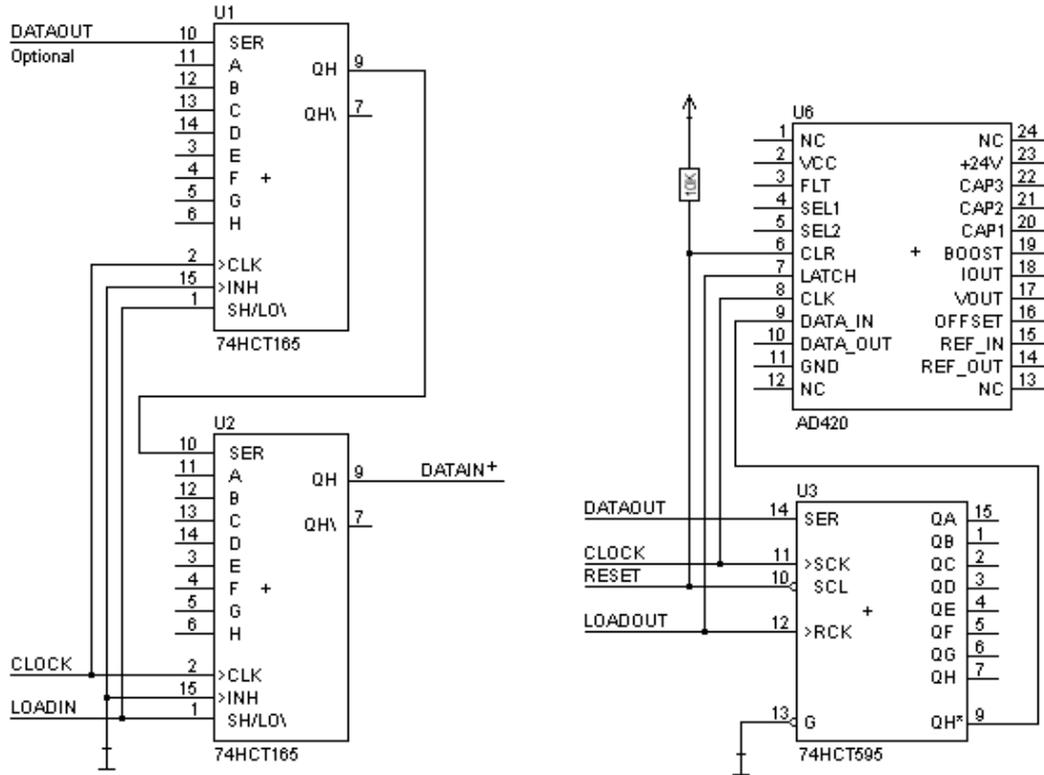
Version 2: Use of UNIGATE® IC2 for digital or analog I/O-modules



Here only the synchronous serial interface is used, the asynchronous serial interface is basically of no account. If you want to program the Script in your completed application, then the use of a connector for the asynchronous interface is recommended. With it you can carry out the ISP-programming.

**For this operating mode no additional controller is required on your application!**

The following circuit diagram is an example for how shift register components can be connected to the IC2.

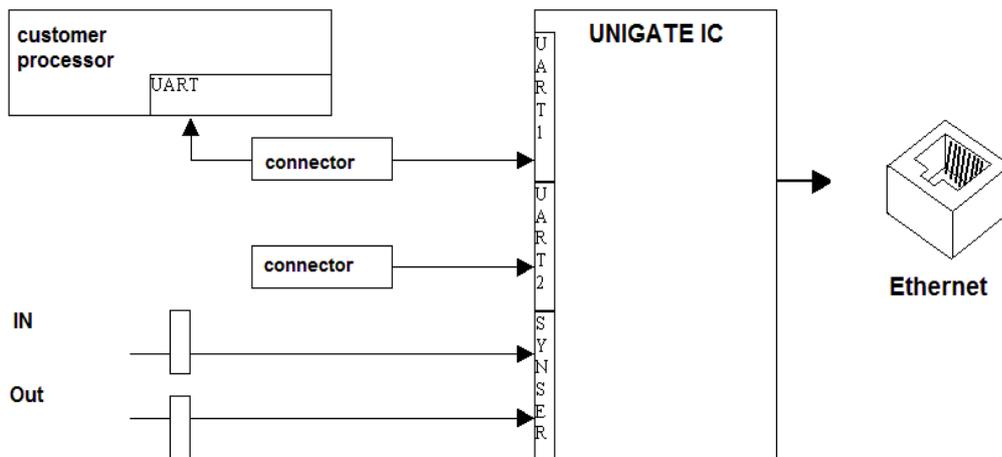


Version 3: Example for digital I/Os

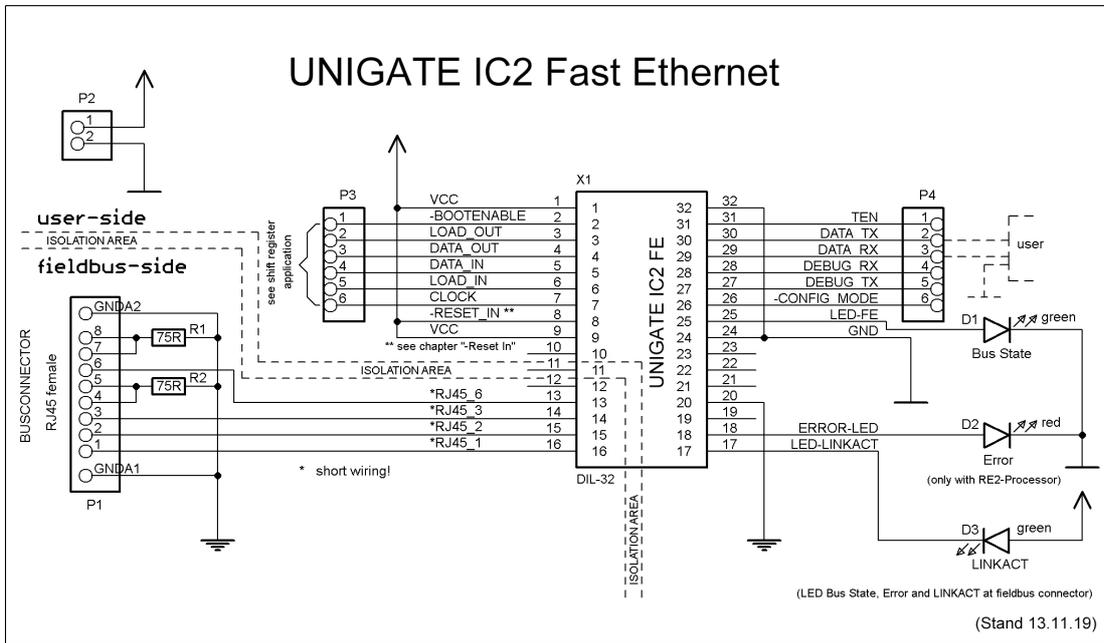
The serial synchronous and the asynchronous interface as well can be operated by UNIGATE® IC2 at the same time. Here the possibility results that an existing application can be extended by additional digital or analog I/Os.

In chapter 5.2 you find an example for a Script, that operates these I/Os.

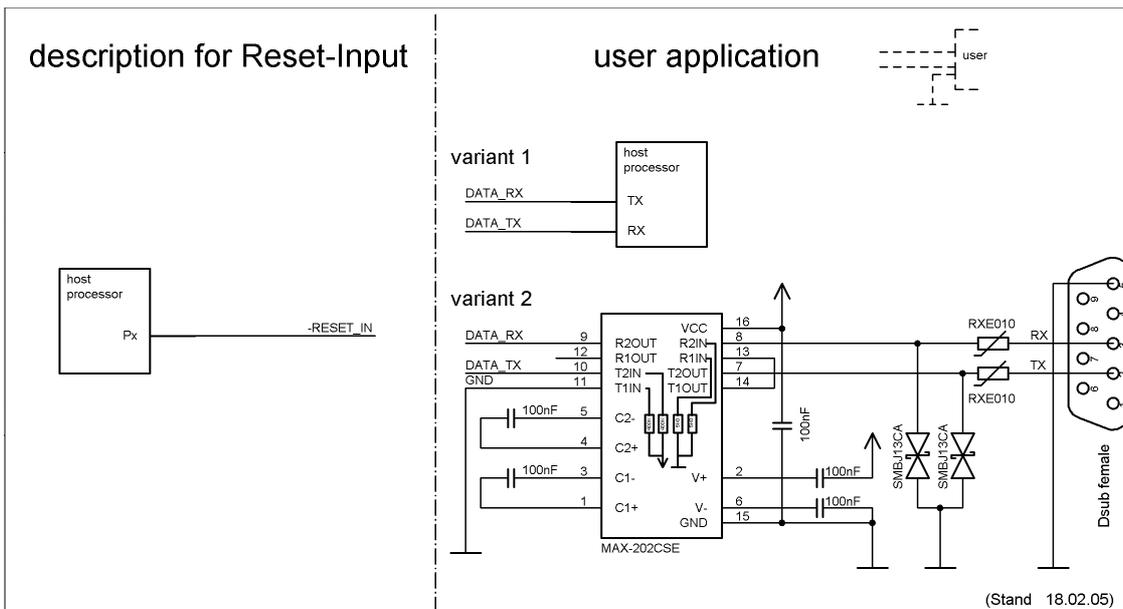
Valid for all versions: A planned plug connection of the serial interface in the application offers the possibility of an update of the firmware or the software via an external connection.

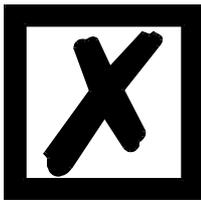
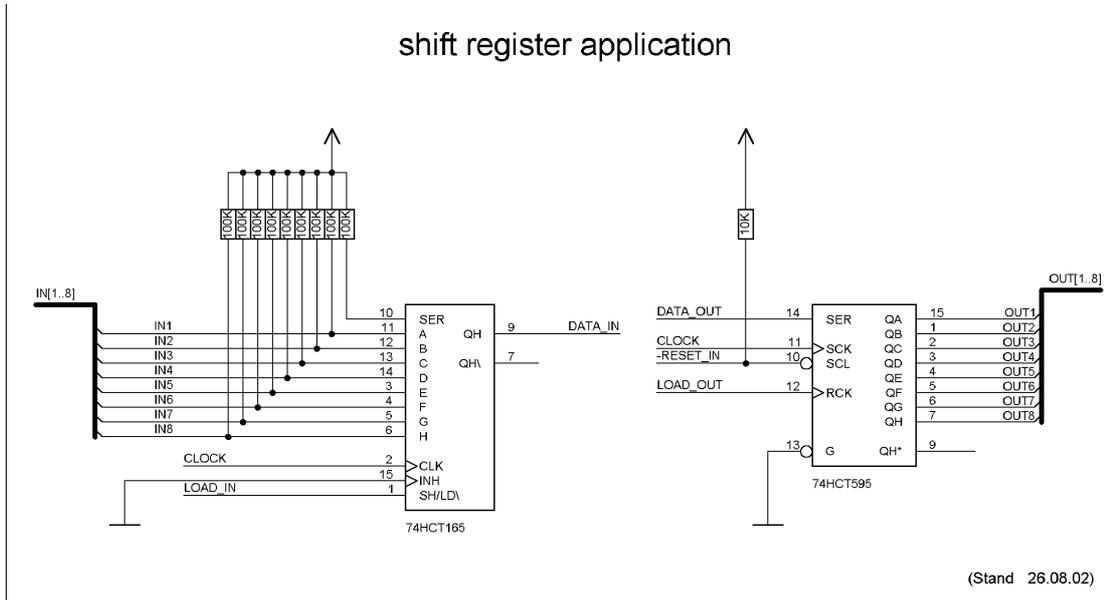


### 3.5 Layout examples



When connecting the UG-IC2 and the RJ45-connector it is important to make sure the signal lines RD-/RD+ resp. TD-/TD+ run as short as possible, parallel and equal in length. Because this is for analog signals, it is also advisable to not place digital lines in close proximity.





The 74HCT595 used in this example has an undefined on-position, but therefor can set the outputs to the tri-state condition via the OutEnable-Pin 13. If it is more important to have a defined on-position in an application, and the OutEnable-pin is not necessary, the 74HCT594 can be used here.

**Note:**

The following recommendation results from the data sheet of the Wiznet W1000 used in the UNIGATE IC-FE:

**Transformer Characteristics**

Parameter	Transmit End	Receive End
Turn Ratio	1:1	1:1
Inductance	350 uH	350 uH
Symmetrical TX & RX channels for auto MDI/MDIX capability		

### 3.6 Handling (mounting the UNIGATE® IC2 on the carrier board)

Depending on the application and the expected shock- and vibration-conditions you can choose from the following possibilities for the UNIGATE® IC2's installation on the carrier board:

- Mounting on a socket in the carrier board. If necessary solder the UNIGATE® IC2 to 2 or 4 pins in the socket. Normally the IC2 can easily be pulled out after the soldering points have been removed.
- Make arrangements for two holes next to the socket in the layout. After the UNIGATE® IC2 was plugged in the socket pull an isolated wire over the IC2 and solder it on the carrier board at the specified holes.
- Fasten the UNIGATE® IC2 With a wire or a tie wrap on the socket.
- Manual soldering directly on the carrier board.
- Automatic soldering directly on the carrier board, whereas „selective“ soldering is essential (no wave soldering)

The advantage of the socketed variant is the easy download of Script- and Firmware-updates, if the carrier board is not designed for it. Besides, that way the Fieldbus can be changed easily by changing the UNIGATE® IC2 if the corresponding plug connectors are provided on the carrier board. Another advantage is, that - normally - only a reflow soldering of the carrier board is necessary.

The advantage of the soldered variant is, that the installation height is lower and a higher shock- and vibration-safety is provided.

### 3.7 Requirement for Ethernet-Interface

There is an extremely large number of suppliers for RJ45 connectors (with and without integrated Ethernet transformer), or single Ethernet transformers on the market, therefore it is not possible for us to test all possibilities. If you consider that there are various qualities (SMD / THT, with / without LED, vertical / horizontal, with / without EMC springs tab at the top / bottom, single / multiple times, etc.) with RJ45 connectors alone, it is probably understandable.

In General, RJ45 connectors and the external or built-in Ethernet transformer must meet the following properties:

#### **Required parameters for Ethernet Transformer:**

Turns ratio 1 CT: 1 CT  $\pm$  5%

Open Circuit Inductance 350  $\mu$ H (min) @ 10 mV, 10 kHz

Insertion Loss 1.1 dB (max)

1500 Vrms dielectric, 1 minute

#### **Required parameters for RJ45 connector:**

CAT5, shielded if possible

To enable a quick start in the selection of appropriate components, we have compiled a list of order numbers of Würth Elektronik below, where we source our components. Of course, other manufacturers are possible, so the following list is to be considered exemplary.

#### **Ethernet transformer 10/100 Base-T 1:1**

We mainly use: Order-No. 749 010 014

#### **RJ45 jack with transformer**

We mainly use: Order-No. 615 008 145 521

#### **RJ45 jack (single or multiple) with integrated transformer**

To be cost-effective and more flexible we don't use RJ45 connectors with integrated transformers. Technically, however, nothing speaks against it.

We exemplary tested: Order-No. 749 905 100 1

For more information:

We had bad experiences with SMT connectors in the past, because in the field they got demolished every now and then due to mechanical loads.

## 4 The serial interface

### 4.1 Overview

The serial interface is the most important connection between the UNIGATE® IC2 and the micro controller of your application. The interface is designed in a way so that your application at least does not have to be changed on the software-side. The wide range of services of the UNIGATE® IC2's serial interface constitutes the basis for it. The UNIGATE® IC2 allows to connect of controllers with a baudrate of 110 baud to 6MBaud. The baudrate for the communication itself is permanently stored in the module. The maximum size for IO-data can be read-out with the Script command "Get RSOutBufFree16".

Depending on the read-in Script of the UNIGATE® IC2, the module carries out actions independently, in order to identify data from the connected device. For customers who already have a software-adaptation at the company Deutschmann Automation, this protocol as well or a Script after an adaptation can be processed by the IC2.

Anyway, the IC2 will take over the communication with the fieldbus independently.

### 4.2 Initialization of the serial interface

The initialization of the interface is carried out by Script-commands, such as "Set baudrate", "Set databits", "Set parity". For a detailed description of these commands see the online documentation for the Protocol Developer or the instruction manual for the Protocol Developer.

### 4.3 Use of the serial interface

The serial interface can freely be programmed by the user. Efficient Script-commands for sending and receiving data are available; just to mention some possibilities: such as waiting with timeout for a character, waiting for a fixed number of characters or also sending and receiving data in the Modbus RTU. A reference to these commands is offered in the online documentation for the Protocol Developer as well as in the instruction manual for the Protocol Developer.

### 4.4 Further operation modes

In the modes configuration mode and firmware-update mode the serial interface also serves to configure the standard software or to carry out a firmware-update. More details can be found in chapter 11.5.

### 4.5 UART

9-Bit mode not possible.

## 5 Synchronous serial interface

The synchronous serial interface of the UNIGATE® IC2 is used to connect clocked shift registers or components that have a Serial Peripheral Interface (SPI).

It allows

- the expansion of the IC2 for digital inputs and outputs (for example for driving LEDs or for reading switch positions)
- communicate with microcontrollers or
- the control of DA and AD converters.

Connection examples are give in chapter 3.

By using the synchronous serial interface can realize products that can work without another microcontroller (stand-alone mode). Examples are sensor products or digital IO modules

### 5.1 Shift register operation

Before the interface can be used, it has to be initialized by setting various Script parameters. (see chapter 5.1.1)

The parameters „ShiftRegisterInputType“ and „ShiftRegisterOutputType“ allow the use of different shift register types, which differ in the polarity of the shift register signals. To use the shift register types 74595 and 74165, for example, the values „RiseClk\_RiseLoad“ and „RiseClk\_LowLoad“ can be set.

The shift register width is set by the parameter „ShiftRegisterInputBitLength“ and „ShiftRegisterOutputBitLength“ The maximum width is 256 bits.

The data exchange with the connected shift registers ensues with the commands „WriteShiftRegister“, „ReadShiftRegister“ or bidirectional with the command „ShiftRegisterDataExchange“. The clock rate is between 280 kHz and 320 kHz.

Further information on the commands and parameter values can be found in the Help section of the Protocol Developer Software. On request, the Deutschmann Script language can be complemented by additional parameter values in order to support other types of shift registers.

#### 5.1.1 Example-Script

Note: The script example refers to the circuit example in chapter 3.4.

```
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] );

// Input data is now in the INBuffer
// 0x58 is applied to the outputs of the analog converter
// 0x21 at the shift register`s outputs
```

## 5.2 SPI mode

Before the interface can be used in SPI mode, this must be initialized. The command `InitSPI` sets the operating type, the mode (signal polarity and phase) and the clock frequency. The data exchange ensues with the command `ExchangeSPI`. The maximum clock frequency is 12 MHz. For details please see the IC2-Pinout list in the download area of our website. Please also refer to the script commands documentation in the online help of the Protocol Developer.

### 5.2.1 Example-Script

```
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 The Debug-interface**

### **6.1 Overview of the Debug-interface**

The UNIGATE® IC2 features a Debug-interface, that allows a step-by-step processing of a Script. Normally this interface is only required for the development of a Script.

### **6.2 Starting in the Debug-mode**

When applying power to the UNIGATE® IC2 (power up) the firmware will output the binary character 0 (0x00) after a self-test was carried out on this interface. If the IC2 receives an acknowledgement via this interface within 500 ms, it is in the Debug-mode. The acknowledgement is the ASCII-character O (0x4F).

With the start in the Debug-mode the further execution of Script-commands will be put to a stop.

### **6.3 Communication parameter for the Debug-interface**

The Debug-interface is always operating with 9600 baud, no parity, 8 data bit, 1 stop bit. It is not possible to change this parameter in the Protocol Developer. Please consider the fact that these settings have to be in accordance with those of the PC-COM-interface and that the flow control (protocol) has to be set on „none“ there.

### **6.4 Possibilities with the Debug-interface**

Usually the Protocol Developer is connected to the Debug-interface. With it a step-by-step processing of a Script, monitoring jumps and decisions and looking at memory areas is possible. Moreover breakpoints can be set. It basically possesses all characteristics a software-development tool is typically supposed to have. However, it is also possible to carry out a Scrip-update via this interface.

### **6.5 Commands of the Debug-interface**

The commands for the use of the Debug-interface are described in the instruction manual Protocol Developer.

## **7 Operating modes of the IC2**

### **7.1 Standard operating mode**

This mode is required for normal use of the IC2. In this mode the IC2 will process all script commands. It is also called data exchange mode.

### **7.2 Configuration mode**

In the configuration mode the UNIGATE® IC2 will execute a self-test after start-up (or after a reset) and wait for further instructions after a successful self-test. Here it is possible to load a translated Script into the device or also to initiate the Firmware Download Mode.

### **7.3 Firmware update mode**

In this mode a newer firmware can be loaded onto the UNIGATE® IC2 (see chapter 12).

## 8 Script and configuration

### 8.1 Overview

The script stored in the UNIGATE® IC2, as well as the configuration, can be replaced or updated via the serial interface (application) in the configuration mode.

### 8.2 The configuration mode

If the pin „ConfigMode“ pulled to GND during the PowerUp or Reset, then the UNIGATE® IC2 starts in the configuration mode. In this mode it is possible to communicate with the IC2 without processing the regular software. In this mode it is possible to change the UNIGATE® IC2's settings of the standard software or to write a new Script in the UNIGATE® IC2. It shows its start in the configuration mode by issuing a status message, which might look as follows:

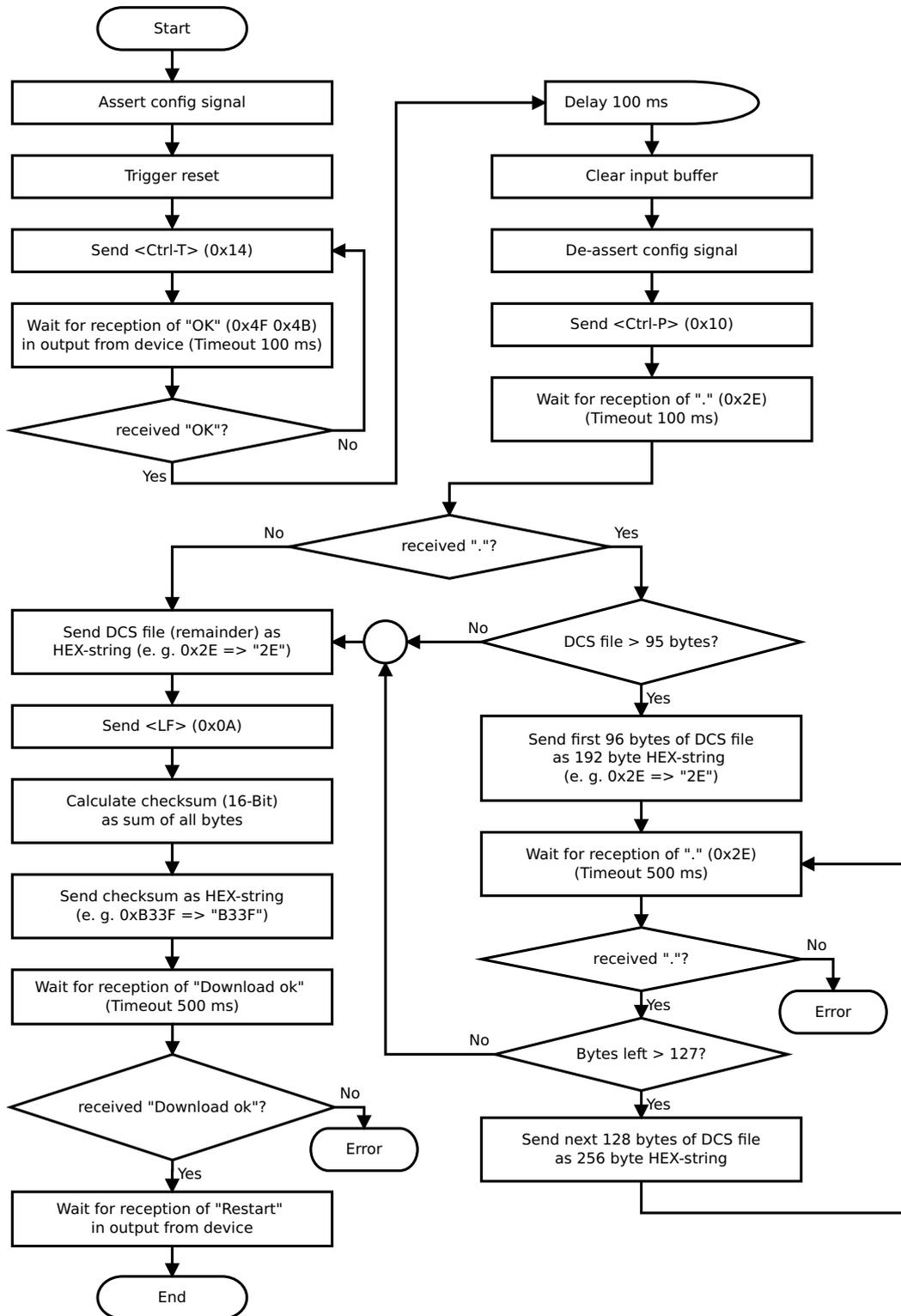
```
IC2FE V1.0.0[44] (c)dA[120MHz] Script(C:190/3968,V:118/16384)=""Leer" Author="Deutschmann Automation GmbH" Version="1.0.2" Date=06.11.2019 SN=47110001 IP=0.0.0.0 MAC=00-14-11-01-02-03 Data-Flash=1MByte
```

The configuration of a UNIGATE® IC2 Fast Ethernet is restricted to setting the Ethernet-address (see also chapter 10.5).

### 8.3 Update the script

- The preferred option is to insert the UNIGATE® IC into the base board from Deutschmann (Developer Board UNIGATE® IC-AB) and use the Deutschmann tools (software WINGATE with „Write Script“ under „File“ or with the software ScriptProgramTool).
- You can also automatically replace the script in your application from your host (see flowchart).

### Script Download



**The operational sequence is as follows:**

The Gateway has to be in the config-mode.

The script-download is initiated with "Ctrl-P (=0x10)".

After that the data follows byte by byte as ASCII-hex-characters.

The download is terminated with a "LF (=0x0A)".

Afterwards the word-checksum follows as ASCII-hex-characters.

The Gateway responds with a clear text reply to that download and carries out a warm start.

Example:

The following 4-bytes script is supposed to be downloaded: 0x01 0x12 0x5A 0x23

The sum of the bytes is 0x0090 as checksum.

Then the following sequence is to be sent:

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's reply: "Download ok"

## 8.4 Configuration of the UNIGATE® IC2

UNIGATE® IC2 is delivered with an empty Script.

The configuration of the UNIGATE® IC2 - Ethernet is limited to the setting of the IP configuration and the selection of DHCP.

### 8.4.1 Ethernet

- Protocol (UDP/IP or TCP/IP), up to four connections at once
- Baud rate: 10/100 MBit/s autodetect
- IP-address (Gateway)
- Ethernet (MAC-address)
- IP-destination host

### 8.4.2 RS232/RS485/RS422

- RS-type: RS232
  - Start bit: 1
  - Data bits: 8
  - Stop bit: 1
  - Parity: None
  - Baud rate: 9600 Baud
- Default setting. This configuration can be changed via the Script.

## 9 Generating a Script

### 9.1 What is a Script?

A Script is a sequence of commands, that are executed in that exact order. Because of the fact that also mechanisms are given that control the program flow in the Script it is also possible to assemble more complex processes from these simple commands.

The Script is memory-oriented. It means that all variables always refer to one memory area. While developing a Script you do not have to take care of the memory management though. The Protocol Developer takes on this responsibility for you.

### 9.2 Memory efficiency of the programs

A Script command can carry out e. g. a complex checksum like a CRC-16 calculation via data. For the coding of this command only 9 byte are required as memory space (for the command itself). This is only possible when these complex commands are contained in a library.

A further advantage of this library is, that the underlying functions have been in practical use for a couple of years and therefore can be described as 'void of errors'. As these commands are also present in the native code for the controller, at this point also the runtime performance of the Script is favorable.

### 9.3 What can you do with a Script device?

Our Script devices are in the position to process a lot of commands. In this case a command is always a small firmly outlined task. All commands can be put into classes or groups. A group of commands deals with the communication in general. This group's commands enable the gateway to send and receive data on the serial side as well as on the bus-side.

### 9.4 Independence of buses

Basically the Scripts do not depend on the bus, they are supposed to operate on. It means that a Script which was developed on a Profibus gateway can also be operated on an Interbus without changes, since the functioning of these buses is very similar. In order to also process this Script on an Ethernet gateway, perhaps further adjustments have to be made in the Script, so that the Script can be executed reasonably.

There are no fixed rules how which Scripts have to operate properly. When writing a Script you should take into account on which target hardware the Script is to be executed, so the necessary settings for the respective buses can be made.

The Ethernet gateway has 4 communication channels.

For the channels the following adjustments in the Script in each case have to be made:

- The port numbers (destination-port and source-port)
- At client operation TCP or at UDP the destination-IP-address
- The mode or the protocol of the channel:

- 0 = TCP-server,
- 1 = TCP-client,
- 2 = UDP,
- 3 = Channel not used (closed)

Without these settings a connection to the gateway cannot be set up via the Ethernet. Only ICMP-data traffic (PING) is possible.

## 9.5 Further settings at the gateway

Most devices require no further adjustments, except for those made in the Script itself. However, there are also exceptions to it. These settings are made by means of the software WINGATE. If you know our UNIGATE®-series, you are already familiar with the proceeding with it. An example is the adjustment of the IP-address and the net-mask of an Ethernet-gateway. These values have to be known as fixed values and are not available for the runtime. Another reason for the configuration of the values in WINGATE is the following: After an update of the Script these values remain untouched, i. e. the settings that were made once are still available after a change of the Script.

Only this way it is also possible that the same Script operates on different Ethernet-gateways, that feature different IP-addresses.

## 9.6 The use of the Protocol Developer

The Protocol Developer is a tool for an easy generation of a Script for our Script gateways. Its operation is exactly aimed at this use. After starting the program the Script that was loaded the last time is loaded again, provided that it is not the first start.

Typical for Windows Script commands can be added by means of the mouse or the keyboard. As far as defined and required for the corresponding command, the dialog to the corresponding command is displayed, and after entering the values the right text is automatically added to the Script. The insertion of new commands by the Protocol Developer is carried out in a way that existing commands will not be overwritten. Generally a new command is inserted in front of the one where the cursor is positioned. Of course the commands can also be written by means of the keyboard or already written commands can also be modified.

## 9.7 Accuracies of the baud rates at UNIGATE® IC2

The baud rate of the serial interface is derived from the processor's crystal frequency. The deviation of the real baud rate from the set baud rate is always less than 1.1%.

## 9.8 Script processing times

The Script is translated by the Protocol Developer and the consequently generated code is loaded into the Gateway. Now the processor in the Gateway interprets this code. In this case, there are commands that can be processed very fast (e. g. "Set Parameter"). There are also commands, however, that take longer (e. g. copying 1000 bytes). Consequently, for one thing the processing time differs due to the kind of Script command. But the processing time of the Script commands is considerably more determined by the processor time that is available for this process. Since the processor has to carry out several tasks simultaneously (multitasking system) only a part of the processor's capacity is available for the Script processing. The following tasks - in the order of priority - are executed on the processor:

- Sending and receiving data at the Debug-interface (provided that the Protocol Developer has been started on the PC)
- Sending and receiving data at the RS-interface
- Sending and receiving data at the Fieldbus-interface
- Tasks controlled via internal clock (1 ms) (e. g. flashing of an LED)
- Processing of the Script

From experience approximately 10  $\mu$ s can be calculated for each Script line. This value confirmed itself again and again in many projects as a standard value. He is always quite right if the processor has enough time available for the Script processing.

By means of the tasks mentioned above, the following recommendation can be formulated in order to receive a rather fast Script processing:

- Deactivate the Debug-interface (it is the normal case in the serial use)
- Keep the data length at the RS-interface as small as possible. The baud rate is not the problem here, but the amount of characters which are transferred per second.
- Do not unnecessarily extend the data length at the Fieldbus side. Especially at acyclical bus data, if possible do only send them when changes were made. The data length at buses that are configured to a fixed length (e. g. PROFIBUS) should not be longer than absolutely necessary.

If the processing time should be too large in spite of these measures, there is the possibility to generate a customized Script command, that executes several tasks in one Script command. Please contact our support department for this purpose.

## 10 Ethernet interface

### 10.1 Communication protocols

Four channels are available, each of which can be initialized with a protocol. TCP, UDP or Modbus/TCP are available as protocols. In addition, a separate protocol can be implemented via the raw mode.

#### 10.1.1 TCP or UDP?

TCP is a connection-oriented protocol: during data transfer, there is a fixed logical connection between client and server here. TCP has all the mechanisms necessary to open a connection, ensure error-free data transfer over the network and close the connection again. For this purpose, the protocol software creates and manages separate buffers for each connection.

In contrast, UDP does not have mechanisms to repeat data packets or to ensure that data is received in the correct order. The data transmission with UDP is recommended, if a superior protocol between the terminal device at the UNIGATE® server and the application on the TCP/IP station ensures an error-free transmission anyway.

#### 10.1.2 The UNIGATE® as TCP server

In the typical TCP/IP application the UNIGATE® is initialized as TCP server. The initialized TCP/IP channel waits in the state "Passive Open" for incoming connections. A TCP client can open a TCP/IP connection with the UNIGATE® via the port number specified in the Script during the initialization of the server. After successful connection establishment, data can then be exchanged.

Starting from firmware version 1.1.0 up to three channels with the same port number can be set in the Script. So that e.g. two clients can communicate with the server at the same time, two channels must also be initialized for this. The ReadBusState parameter can be used to determine the state of the four available channels. For example, whether a connection exists. To communicate with several clients, the script should regularly change the channel and serve all existing connections.

If all initialized channels are occupied, further connection requests are queued. queue. No data can be exchanged on waiting connections. This is only possible when a channel becomes free and the waiting connection moves up.

#### 10.1.3 TCP timeout

A TCP connection, where the UNIGATE® IC2 Fast Ethernet is the server, is closed after 65635 seconds (18.2 hours), if no data has been transferred during this time.

#### 10.1.4 MODBUS/TCP

For the use of the Modbus/TCP protocol various script templates and examples are included in the Protocol Developer software.

### 10.2 Data exchange mode

The UNIGATE® must be in this mode, so that the installed Script can be executed and thus a data exchange can take place. It is also called standard operating mode.

### 10.3 Web server

If there are web pages, images or Java applets in the internal file system, they can be made accessible via the web server. (Files can be brought in via the FTP server).

The web server is off by default. By setting the parameter Web server after the bus start it can be activated in the script (value: 1) or deactivated again (value: 0).

Because of the necessary activation via the script, the web server works only in run mode. Otherwise it works independently of the script and does not occupy a channel.

### 10.4 FTP-server

The FTP server is automatically active in Config mode. In Run mode it is deactivated like the web server and can be activated and deactivated again in the script after the bus start via the parameter FTP server with the values values 1 and 0 in the script.

By default, the FTP server uses the user name "deutschmann" and the password "deutschmann" for login. (Anonymous login is not possible).

The login should be changed in the script by setting the parameters FTP\_UserName and FTP\_Password. (Only valid in run mode; in config mode the default login is still valid).

In run mode, the FTP server works independently of the script except for activation and does not occupy a channel.

#### 10.4.1 Script update via FTP

The DCS file generated by the Protocol Developer by compiling from the source code must be uploaded as "script.dcs" via FTP to the UNIGATE. The UNIGATE® recognizes this file at the start, takes over the Script and deletes the file afterwards.

#### 10.4.2 System configuration update via FTP

A GWC file generated by WINGATE must be uploaded as "system.gwc" via FTP to the UNIGATE. The UNIGATE® recognizes this file at the start, takes over the new configuration and deletes the file afterwards.

#### 10.4.3 File system

The UNIGATE® contains a memory module in which files are permanently stored. The file names may be up to 255 characters long. The maximum storage capacity is currently 1 MByte. On request up to 8 MByte can be equipped.

Subdirectories are not possible.

### 10.5 Setting the IP address

The UNIGATE® IC2 must be in the configuration mode (see also chapter 8.2 'The configuration mode'). Then it is possible to set the address with WINGATE or UNIGATE-SCAN. his address remains until it is changed again.

In the Script the IP address can also be set via the parameter IP\_volatile. With this method the set address is lost at a restart.

## 10.6 Zero Configuration

The UNIGATE<sup>®</sup> IC2-FE supports the function "Zero-Configuration". This means that the UNIGATE<sup>®</sup> automatically uses an IP address within the range of the Zero-Configuration at IP address 0.0.0.0. However, UNIGATE does not check whether the then used IP address is already in use.

## 10.7 Comparison with former UNIGATE<sup>®</sup> FE

### 10.7.1 UDP channel mode

With the FEX it was possible to initialize a channel with destination IP 0 and port 0 and thus to receive UDP packets from all IPs and all ports via this socket. This is no longer possible.

### 10.7.2 Raw-channel mode

In the IC2-FE not only channel 0 can be used, but any channel can be used in raw mode. However, it is then no longer possible to receive anything via UDP or TCP on other channels. There can only be one channel in Raw mode.

### 10.7.3 BusStatus

At the IC2-FE no channels are occupied for the functions UNIGATE-Scan, FTP- and Webserver. Thus ReadBusStatus after BusStart is 0x202222 as long as no further channels are initialized by the script.

**Script example for the initialization of the Ethernet**

```
var wRemotePort: word; Moveconst(wRemotePort, 2001);
var wLocalPort: word; Moveconst(wLocalPort, 2000);
var lRemoteIP: long; Moveconst(lRemoteIP, 0xc0a82f0b); // 192.168.47.11

var lE3: long; Moveconst(lE3, 0xE3222222); // Channel 0 connected, rest
unused
var w0: word; Moveconst(w0, 0);

var bChannel: byte; Moveconst(bChannel, 0);
var lBusState: long;
var wInCount: word;
var wOutCount: word;
var aRecvData: buffer[100];
var aSendData: buffer[100];

// Initialize channel 0 as TCP-Client
InitFastEthernet(bChannel, TCPIP_Client, wRemotePort, wLocalPort, lRe-
moteIP);

BusStart; // activateEthernet

:waitForConnection; // Check TCP connection, wait if necessary
(unnecessary for UDP)
Get(ReadBusState, lBusState);
if lBusState equal lE3 then :connected;
jump :waitForConnection;

:connected; // Check for reception when connected
Get(AvailableBusData, wInCount);

if wInCount equal w0 then :send; // read and evaluate on receipt
ReadBus(aRecvData[0], wInCount);
{add code here to evaluate reception}

:send; // send something
Moveconst(wOutCount, 13);
Moveconst(aSendData[0], "Sende data...");
WriteBus(aSendData[0], wOutCount);

jump :waitForConnection; // typically scripts run as endless loop
```

## 11 Error handling at UNIGATE® IC2

A distinction can be made between two categories of system-errors:

Serious errors (1-4): In this case, the Gateway must be switched off and switched back on again. If the error occurs again, the Gateway must be exchanged and returned for repair.

Warnings (6-15): These warnings are displayed for one minute simply for information purposes and are then automatically reset. If such warnings occur frequently, please inform After-Sales Service.

The system-error can be read-out via the Script.

In the configuration mode these displays are not valid and only meant for internal use.

Error no.	Error description
0	Reserved
1	Hardware fault
2	EEROM error
3	Internal memory error
4	Fieldbus hardware error
5	Script error
6	Reserved
7	RS-transmit buffer overflow
8	RS-receive buffer overflow
9	RS timeout
10	General fieldbus error
11	Parity-or frame-check-error
12	Reserved
13	Fieldbus configuration error
14	Fieldbus data buffer overflow
15	Reserved

Table 1: Error handling at UNIGATE® IC2

## 12 Firmware-update

### 12.1 Overview

In the firmware-update-mode the firmware can be replaced via the serial interface of the UNIGATE® IC2.

### 12.2 Adjusting the firmware-update-mode

#### 12.2.1 Adjustment by hardware

UNIGATE® IC2 can be brought to the firmware-update-mode by the hardware. For it the signal -BE (-boot enable) must be pulled to the potential GND during the Power-up-process.

#### 12.2.2 Adjustment by software

If the UNIGATE® IC2 is in configuration mode (see chapter 8.2), it can be automatically switched to firmware update mode via "FDT. EXE" (firmware download tool) .

### 12.3 Execution of the firmware-update

The firmware update is carried out with the developer board in combination with the software "FDT.EXE" (firmware download tool). These tools are available from Deutschmann. (see chapter 14).

### 12.4 Safety note

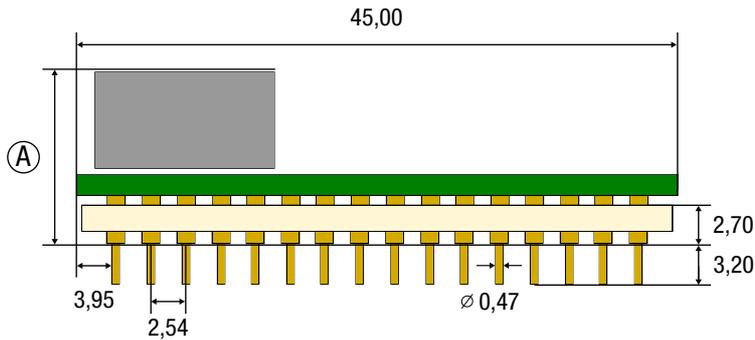
An update of the firmware is e.g. useful, if new functions are required, which are not yet supported in the existing firmware. Also if the UNIGATE® is e.g. in boot mode or in an undefined state, the reloading of the firmware resp. an update of the firmware can provide a remedy. A started firmware update process must not be interrupted.

### 13 Technical data

In this chapter you will find all necessary technical data on UNIGATE® IC2.  
All measurements in mm.

#### 13.1 Mechanics of the UNIGATE® IC2

##### 13.1.1 General dimensions of UNIGATE® IC2



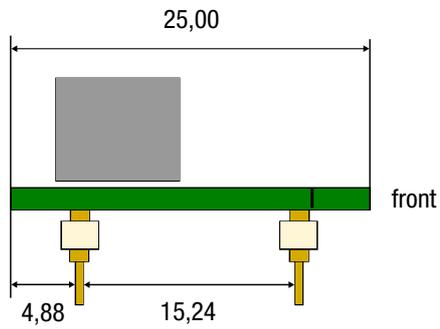
UNIGATE® IC2

	EtherCAT (wt*)	Ethernet TCP/IP + Modbus TCP	Ethernet TCP/IP (wt*) + Modbus TCP (wt*)	PROFIBUS DP	PRROFINET 2Port (wt*)
A= total height (mm)	8	11	8	13	8

\*wt = without magnetics

**Note:** The total height of all UNIGATE® ICs can be found in the "Pinout list for UNIGATE® IC and UNIGATE® IC2". (Download PDF)

The pins of the UNIGATE<sup>®</sup> ICs are arranged in a grid dimension of 2.54 mm



DIP-Spacing Code 6

In case you intend to use other fieldbus IC2s, the maximum overall height of  $\leq 20$  mm (including pins) has to be taken into consideration.

## 13.2 Technical data UNIGATE® IC2 - Fast Ethernet

Characteristics	Explanation
Supply voltage	3,3 V $\pm$ 5 %, 200 mA (typ.), 210 mA (max.)
Reverse polarity protection power supply	No
Interface	2 UART interfaces, 1 synchronous serial interface
Physical separation Ethernet-side	Standard
TCP/IP baud rate	Adjustable 10/100 MBit/s
UART-baud rate	Up to 6 MBaud (adjustable via Script)
Protocols	UDP, TCP, Modbus TCP, HTTP, FTP
Technology	Microprocessor
Others	I/Os, analogous signals, shift registers, LEDs, switches and so on can be connected externally
Dimensions	45 x 25 x 11 mm (WxDxH)
Installation	32 DIL
Weight	Approx. 9 g
Ambient temperature*	-40°C..+70°C
Storage / transport temperature	-40°C..+100°C
Built-in position	Any

\* When implementing a UNIGATE® IC in your electronics, not only the mechanical dimensions but also the thermal management must be taken into account. The ambient temperature specified in the manual does not take into account the specific installation situation (e.g. closed housing). External factors have a major influence on how well the heat is dissipated from the heat sources. When using the UNIGATE® IC in your electronics, adequate heat dissipation must therefore be ensured.

## 14 Accessory

The following tools are available from Deutschmann Automation.

### 14.1 FirmwareDownloadTool (FDT)

The FirmwareDownloadTool is available for download from our homepage: it is required for an update of the firmware. Condition for it is, that a PC can be connected to the serial of the IC2. The software describes the procedure of an update itself.

### 14.2 Protocol Developer

The Protocol Developer is the development environment for Scripts, that also contain the Debugger. This software package also contains the documentation to all Script-commands. This software is available for download from our homepage at <http://www.deutschmann.com>. The instruction manual for the Protocol Developer, which is available in pdf-format, gives further advise on how to use the software.

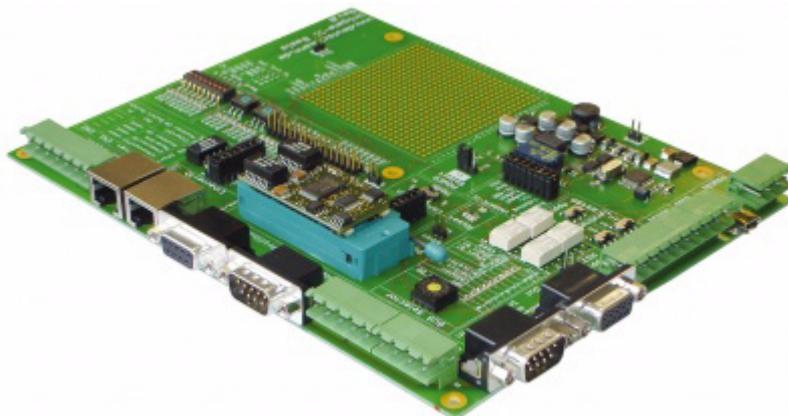
### 14.3 Developer Kit UNIGATE® IC-AB IC

The Developer Kit IC contains

- a Developer Board UNIGATE® IC2 (see chapter 14.3.1)
- a plug-in power pack to supply the Developer Board
- connection cables for appl. RS232, Debug RS232 and appl. RS422/485
- USB-cable
- Software and documentation to complete the packet.

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

The Developer Board was developed so that the fast implementation of the Deutschmann All-in-one bus node UNIGATE® IC2 into your electronic system can be guaranteed. The board is suitable for all Fieldbuses and Industrial Ethernet Buses supported by Deutschmann Automation.

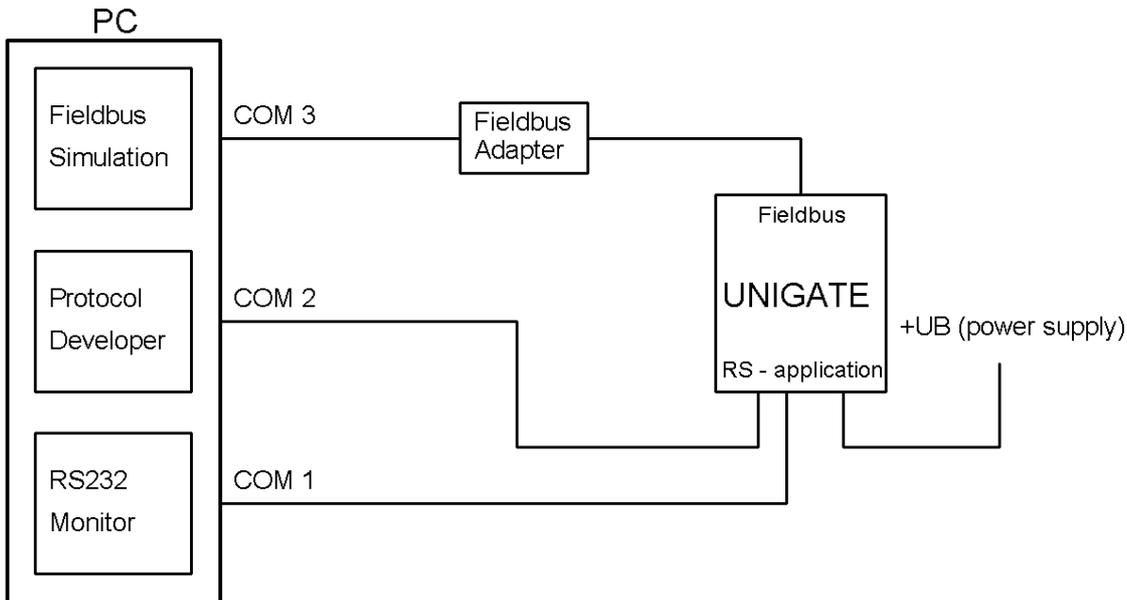


The required UNIGATE® IC2 / ICs are ordered separately. The required voltage (5V or 3.3V, depending on the version) can be adjusted. An RS232-interface or a USB-connection is available for the connection to the PC (Debug-interface).

The application can be connected either through the USB, RS232, RS485 or the RS422.

The bus-connections according to standard or market standard are available to test the respective bus-side. Optionally Deutschmann Add-on packages (bus-master simulation) are available. The board contains 32 bit input and 16 bit output, provided with one LED each. Different connectors allow an easy coupling to your processor. A hole matrix field with the most important signals (voltage, IOs) allows a customized hardware extension (e. g. to connect a D/A converter).

**14.3.2 Quick start**



For a transparent data exchange you will find example Scripts for the respective Fieldbus under "File->New" in the Protocol Developer.

## 15 Appendix

### 15.1 Explanations of the abbreviations

#### General

CL	=	Product group CL (Compact Line)
CM	=	Product group CM (CANopen Line)
CX	=	Product group CX
EL	=	Product group EL (Ethernet Line)
FC	=	Product group FC (Fast Connect)
GT	=	Galvanic separation RS-side
GY	=	Housing color gray
MB	=	Product group MB
RS	=	Product group RS
SC	=	Product group SC (Script)
232/485	=	Interface RS232 and RS485 switchable
232/422	=	Interface RS232 and RS422 switchable
DB	=	Additional RS232 DEBUG-interface
D9	=	Connection of the RS through 9-pin D-SUB instead of 5-pin screw-plug connector
PL	=	Board only without DIN-rail module and without housing cover
PD	=	Board only without DIN-rail module and with housing cover
AG	=	Gateway installed in a die-cast aluminum housing
EG	=	Gateway installed in a stainless steel housing
IC2	=	Produktgruppe IC2 (IC2-Bauform DIL32)
IC	=	Product group IC (IC-design DIL32)
IO8	=	Option I/O8
16	=	Script memory expanded to 16KB
5V	=	Operating voltage 5V
3,.3V	=	Operating voltage 3.3V

#### Fieldbus

CO	=	CANopen
C4	=	CANopen V4
C4X	=	CANopen V4-version X (see comparison table UNIGATE <sup>®</sup> IC for the respective product)
DN	=	DeviceNet
EC	=	EtherCAT
EI	=	Ethernet/IP
FE	=	Ethernet 10/100 MBit/s
FEX	=	Ethernet 10/100 MBit/s-version X (see comparison table UNIGATE <sup>®</sup> IC for the respective product)
IB	=	Interbus
IBL	=	Interbus
LN62	=	LONWorks62
LN512	=	LONWorks512
ModTCP	=	ModbusTCP
MPI	=	Siemens MPI <sup>®</sup>
PL	=	Powerlink
PN	=	Profinet-IO
PBDP	=	ProfibusDP

PBDPL = ProfibusDP-version L (see comparison table UNIGATE® IC for the respective product)  
PBDPX = ProfibusDP-version X (see comparison table UNIGATE® IC for the respective product)  
PBDPV0 = ProfibusDPV0  
PBDPV1 = ProfibusDPV1  
RS = Serial RS232/485/422

## 16 Servicing

Should questions arise that are not covered in this manual you can find further information in our

- FAQ/Wiki area on our homepage [www.deutschmann.com](http://www.deutschmann.com) or directly in our Wiki on [www.wiki.deutschmann.de](http://www.wiki.deutschmann.de)

If your questions are still unanswered please contact us directly.

### Please note down the following information before calling:

- Device designation
- Serial number (S/N)
- Article number
- Error number and error description

Your request will be recorded in the Support center and will be processed by our Support Team as quickly as possible (Usually in 1 working day, rarely more than 3 working days.).

Technical Support hours are as follows:

Monday to Thursday from 8 am to midday and from 1 pm to 4 pm, Friday from 8 am to midday (CET).

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

Central office and sales department +49 6434 9433-0  
Technical Support +49 6434 9433-33

Fax sales department +49 6434 9433-40  
Fax Technical Support +49 6434 9433-44

E-mail Technical Support [support@deutschmann.de](mailto:support@deutschmann.de)

### 16.1 Returning a device

If you return a device, we require as comprehensive a fault/error description as possible. We require the following information in particular:

- What error number was displayed?
- What is the supply voltage ( $\pm 0.5$  V) with Gateway connected?
- What were you last doing or what last happened on the device (programming, error on power-up, ...)?

The more precise information a fault/error description you provide, the more exactly we will be able to pinpoint the possible causes.

### 16.2 Downloading PC software

You can download current information and software free of charge from our Internet server. <http://www.deutschmann.com>



