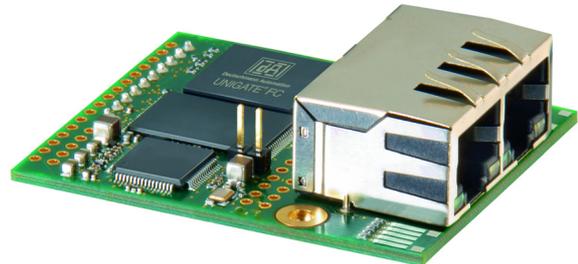


**Deuschmann**

*your ticket to all buses*

**Instruction Manual  
Universal Fieldbus-Gateway  
UNIGATE<sup>®</sup> FC - PROFINET 2Port**



**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>8</b>
<b>2</b>	<b>The UNIGATE® FC</b>	<b>9</b>
2.1	Technical introduction	9
2.2	Availability	9
2.3	Firmware	9
2.4	The serial standard Interface	9
2.5	The synchronous serial Interface	9
2.6	The Debug-Interface	10
2.7	UNIGATE® FC hardware survey	10
<b>3</b>	<b>Hardware design.</b>	<b>11</b>
3.1	Ports	11
3.2	Pinout	12
3.2.1	-Boot enable	12
3.2.2	Load out (SPI-Master: SS0-)	12
3.2.3	Data out (SPI-Master: SS1-)	12
3.2.4	Data In (SPI: MISO)	13
3.2.5	Load In (SPI: MOSI)	13
3.2.6	Clock (SPI: SCK)	13
3.2.7	-Reset In	13
3.2.8	Error LED	13
3.2.9	Ground	13
3.2.10	State-LED	13
3.2.11	-Config Mode	13
3.2.12	DbgTX, DbgRx.	14
3.2.13	TX, RX.	14
3.2.14	TE	14
3.3	Software	14
3.4	Basic line of proceeding	14
3.5	Connection examples	15
3.6	Layout examples	17
3.7	Handling (mounting the UNIGATE® FC on the carrier board)	19
<b>4</b>	<b>The serial interface</b>	<b>20</b>
4.1	Overview	20
4.2	Buffer sizes at the UNIGATE® FC	20
4.3	Initialization of the serial interface	20
4.4	Use of the serial interface	20
4.5	Further operation modes	20
<b>5</b>	<b>Synchronous serial interface</b>	<b>21</b>
5.1	Shift register operation	21
5.1.1	Example-Script	21

---

5.2	SPI mode . . . . .	22
5.2.1	Example-Script . . . . .	22
<b>6</b>	<b>The Debug-interface . . . . .</b>	<b>23</b>
6.1	Overview of the Debug-interface . . . . .	23
6.2	Starting in the Debug-mode . . . . .	23
6.3	Communication parameter for the Debug-interface . . . . .	23
6.4	Possibilities with the Debug-interface . . . . .	23
6.5	Commands of the Debug-interface . . . . .	23
<b>7</b>	<b>Script and configuration . . . . .</b>	<b>24</b>
7.1	Overview . . . . .	24
7.2	The configuration mode . . . . .	24
7.3	Update the script . . . . .	24
7.4	Configuration of the UNIGATE® FC . . . . .	24
7.4.1	RS232/RS485/RS422 . . . . .	24
<b>8</b>	<b>Generating a script . . . . .</b>	<b>25</b>
8.1	What is a script? . . . . .	25
8.2	Memory efficiency of the programs . . . . .	25
8.3	What can you do with a script device? . . . . .	25
8.4	Independence of buses . . . . .	25
8.5	Further settings at the gateway . . . . .	25
8.6	The use of the Protocol Developer . . . . .	26
<b>9</b>	<b>FTP-Server . . . . .</b>	<b>27</b>
9.1	Script-update via FTP . . . . .	27
9.2	System configuration update via FTP . . . . .	27
<b>10</b>	<b>WEB-Server . . . . .</b>	<b>28</b>
10.1	Server Side Includes (SSI) . . . . .	28
10.2	HTML-Formulare . . . . .	29
10.3	Examples . . . . .	29
<b>11</b>	<b>File System . . . . .</b>	<b>30</b>
<b>12</b>	<b>PROFINET . . . . .</b>	<b>32</b>
12.1	Data exchange mode . . . . .	32
12.2	PROFINET address-assignment . . . . .	32
12.3	PROFINET Device Name . . . . .	32
12.4	Project planning . . . . .	32
12.5	Possible data lengths . . . . .	33
12.6	Startup phase . . . . .	33
12.7	Literature . . . . .	33

<b>13</b>	<b>Hardware-ports and light-emitting diodes (LED)</b>	<b>34</b>
13.1	View	34
13.2	Connectors	34
13.2.1	PROFINET connectors	34
13.3	Light indicator	35
13.3.1	LED 1 „Link / Act. P1“	35
13.3.2	LED 3 „Link / Act. P2“	35
13.3.3	LED 5 „(PROFINET) State“	35
13.3.4	LED 7 „State“	35
13.3.5	LED 8-11 „1 / 2 / 4 / 8 (Error No.)“	36
13.3.6	LED 12 „Power“	36
<b>14</b>	<b>Error handling at UNIGATE® FC</b>	<b>37</b>
<b>15</b>	<b>Firmware-Update.</b>	<b>38</b>
15.1	Overview	38
15.2	Execution of the firmware-update	38
15.3	Adjusting the firmware-update-mode	38
15.3.1	Adjustment by hardware	38
15.4	Note on safety	38
15.5	Operation mode of the FC	38
<b>16</b>	<b>Technical data</b>	<b>39</b>
16.1	Mechanics of the UNIGATE® FC	39
16.1.1	General dimensions of UNIGATE® FC	39
16.1.2	Dimension drawing UNIGATE® FC	39
16.1.3	Height of the UNIGATE® FC	40
16.2	Technical data UNIGATE® FC PROFINET	40
<b>17</b>	<b>Accessory</b>	<b>41</b>
17.1	FirmwareDownloadTool (FDT)	41
17.2	Protocol Developer	41
17.3	Developerkit UNIGATE® FC	41
17.3.1	Developerboard UNIGATE® FC	41
17.3.2	Quick start	42
<b>18</b>	<b>Appendix</b>	<b>43</b>
18.1	Explanations of the abbreviations	43
18.2	Hexadecimal table	44
<b>19</b>	<b>Servicing</b>	<b>45</b>
19.1	Downloading PC software, GSDML-file, Script examples etc.	45



#### 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 – 2016. 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.

Art.-No.: V3893E

## 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> FC by Deutschmann Automation any more. All digital functions, software, stack and driver are integrated on a UNIGATE<sup>®</sup> FC in correspondence with the standard. In addition to the reduction of the required size, also different fieldbusses can easily be integrated.

**Through the flexible firmware of UNIGATE<sup>®</sup> FC 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> FC 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.

## 2 The UNIGATE® FC

### 2.1 Technical introduction

The UNIGATE® FC by Deutschmann Automation contains all components that are required for the communication in a fieldbus in one single module. Therefore a developer does not have to take care for that detail any more, only a hardware redesign is necessary in order to integrate the UNIGATE® FC.

### 2.2 Availability

The module is available as PROFINET. Further Industrial Ethernet Buses are either planned or being worked on. The hard- and software-interfaces of the Deutschmann UNIGATE® FC-series are standardized and functionally identical. As a result, the exchangeability between the different UNIGATE® FC-versions is guaranteed.

You can find an up-to-date list for all UNIGATE® FCs under:  
<http://www.deutschmann.com>.

### 2.3 Firmware

UNIGATE® FC is programmed via scripts. On principle any script, that has been developed for a UNIGATE® IC, can also be operated on the UNIGATE® FC.

### 2.4 The serial standard Interface

Intelligent devices, that already feature a micro controller or a microprocessor, are generally supplied with a serial asynchronous interface with a TTL-level. It is directly connected with the TTL-interface of the UNIGATE® FC. For more information on this serial interface see chapter 4 on page 20.

### 2.5 The synchronous serial Interface

In addition to the standard interface there is also the possibility of the synchronous input and output. That way for instance digital IOs can be connected through shift register components or also analog IOs can be connected through a DA-converter with serial in-/output. For synchronous IOs 256 signals at the most can be used (256 bit). Wiring examples can be found in chapter 3.5 and for software examples see chapter 5. This interface can also be used to connect modules or devices with SPI-interface. It is also possible to build, for instance digital or analogous I/O-modules, with the customer's device not being equipped with an own controller. The fieldbus IC is also operable autonomously without that controller.

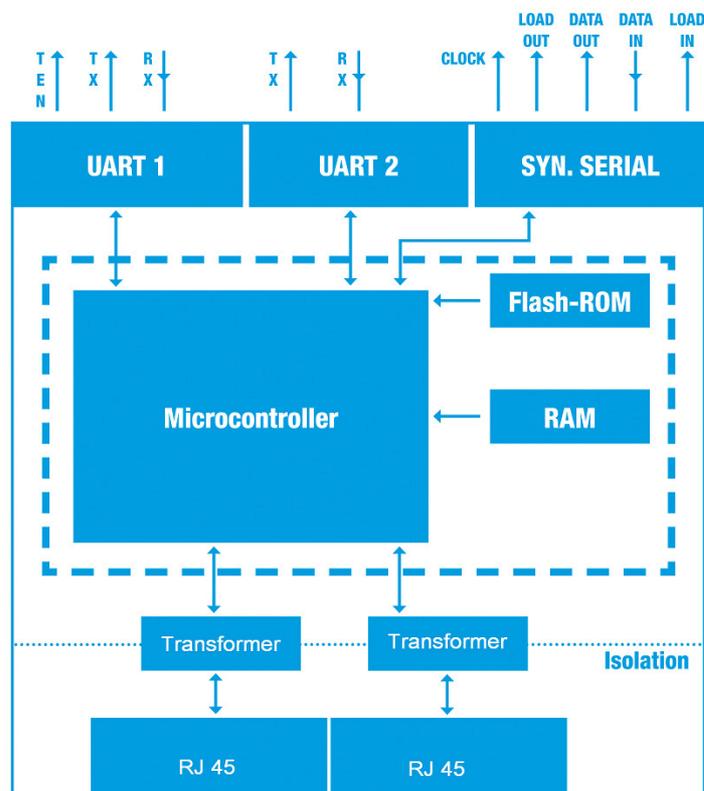
## 2.6 The Debug-Interface

The UNIGATE® FC features a Debug-interface, which allows to process a script step by step and also to monitor or manipulate data. This is indispensable for the development of a script. Usually a script is developed with the software Protocol Developer. For more details take a look at the instruction manual Protocol Developer. You'll find more information on the Debug-Interface in chapter 6 on page 23.

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

## 2.7 UNIGATE® FC hardware survey

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

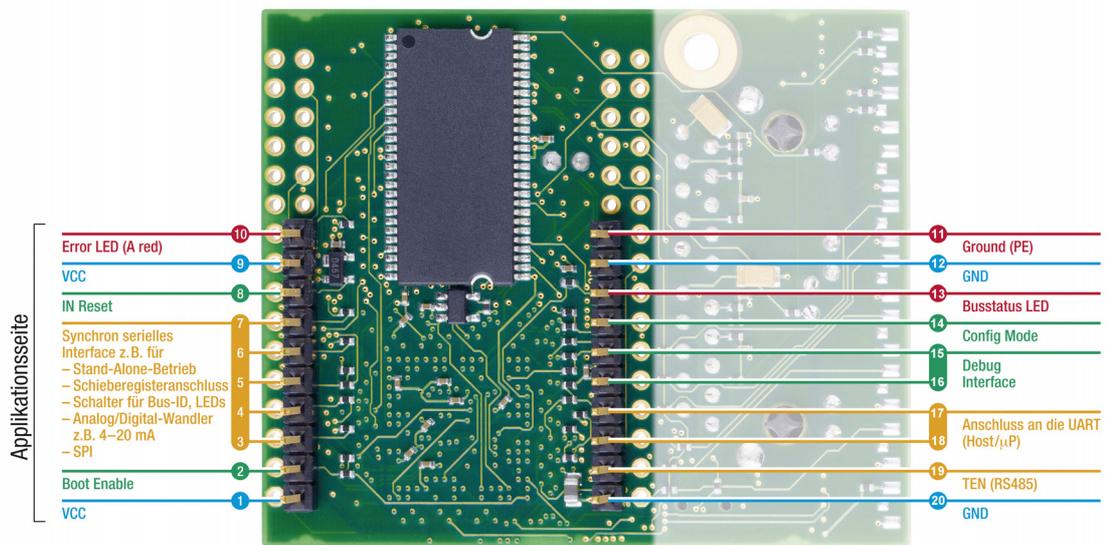


### 3 Hardware design

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

#### 3.1 Ports

UNIGATE<sup>®</sup> FC features 64 pins in its layout with only 20 pins routed to the outside. The exact mechanical dimensions can be taken from chapter 16 on page 39.



## 3.2 Pinout

Pin	Technical specifications	Name	Description	Remark
1*	3.3V ± 5% < 550mA	Vcc	+3.3V voltage supply	
2	IN <sub>Logic</sub>	-BE	-Boot enable	internally pulled up with 10 kΩ
3	OUT <sub>Logic</sub>	Load out (SS0-)	strobe signal of output data of the synchronous, serial Interface	
4	OUT <sub>Logic</sub>	Data out (SS1-)	output data of the synchronous, serial Interface	
5	IN <sub>Logic</sub>	Data in (MISO)	input data of the synchronous, serial interface	internally pulled up with 10 kΩ
6	OUT <sub>Logic</sub>	Load in (MOSI)	strobe signal of the input data of the synchronous, serial interface	
7	OUT <sub>Logic</sub>	Clock (SCK)	clock pulse signal for synchronous, serial interface	
8	IN <sub>Reset</sub>	-Reset in	reset-input of the FC	min. 10µs
9*	connected to pin 1	Vcc	+3.3V voltage supply	
10		Error-LED	Anode of the red LED	
11		earth	Ground (PE)	
12*	connected to pin 20	GND	Ground supply voltage of the FC	
13		State-LED	Bus status LED (Anode of the green LED)	
14	IN <sub>Logic</sub>	-Config Mode	signal to start the configuration mode	internally pulled up with 10 kΩ
15	OUT <sub>Logic</sub>	DbgTX	serial Debug TX	
16	IN <sub>Logic</sub>	DbgRX	serial Debug RX	internally pulled up with 10 kΩ
17	IN <sub>Logic</sub>	RX	serial data RX	internally pulled up with 10 kΩ
18	OUT <sub>Logic</sub>	TX	serial data TX	
19	OUT <sub>Logic</sub>	TE	Transmit enable	Steuerung RS422/485-Driver
20*	GND	GND	Ground supply voltage of the FC	

\* The supply voltage is 3.3V +/- 5 %, max. 550 mA DC.

<b>3.3V ± 5%, &lt; 550mA</b>	V <sub>IL</sub>	V <sub>IH</sub>
IN <sub>Reset</sub>	< 0.75V / 0.4mA	>2.5V / 10µA (max. 4V)
IN <sub>Logic</sub>	< 0.8V / 0.4mA	>2V / 10µA (max. 3.6V)
	V <sub>OL</sub>	V <sub>OH</sub>
OUT <sub>Logic</sub>	< 0.5V / 1mA	>2.3V / 1mA (max. 3.6V)

### 3.2.1 -Boot enable

The FC is started in the firmware update mode with the level GND during the power up process. See also chapter 15 on page 38.

### 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 emitted 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® FC, 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 needs to last at least 10 µs.

### 3.2.8 Error LED

A red LED can be connected to this line (siehe Kapitel 3.6). It can have the following conditions:

Off	No error
Flash	PROFINET user flash test
On	Error

### 3.2.9 Ground

This pin is directly connected to earth line (drilling Ø 3 mm) of the UNIGATE® FC.

### 3.2.10 State-LED

A green LED can be connected to this line (see chapter 3.6). It shows the bus status and can have the following conditions:

Off	Bus not started
Flash 10Hz	Error
Flash 2 Hz	Bus started, waiting for connection / configuration mode
On	connected

### 3.2.11 -Config Mode

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

### 3.2.12 DbgTX, DbgRx

They are transmission and receive line as well of the FC's Debug-interface. For the function description of the Debug-interface see chapter 6 on page 23.

### 3.2.13 TX, RX

Transmission and receive line of the serial interface. This interface is programmable in accordance with the description in chapter 4 on page 20.

### 3.2.14 TE

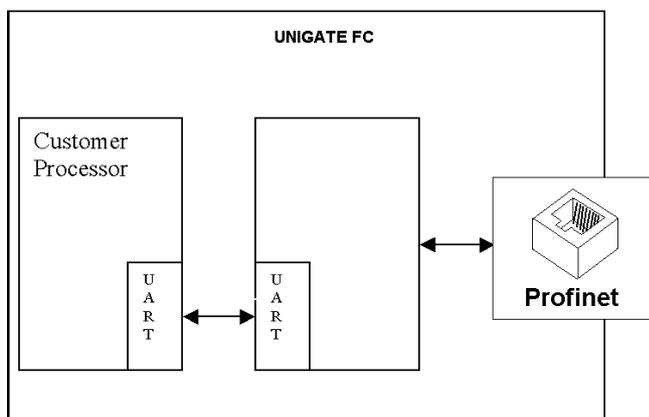
The transmit enable signal allows the connection of RS485 drivers to the IC's serial interface. The signal is set to High whenever the FC sends via the line TX.

## 3.3 Software

The software executes script-commands, which in turn control the FC's hardware and they process their complete protocol by software. The script itself can be generated by the company Deutschmann Automation or with the software Protocol Developer by yourself. For a detailed description of the script.commands of the Protocol Developer see the instruction manual Protocol Developer and the online documentation concerning script-commands.

## 3.4 Basic line of proceeding

In theory it is enough to replace the RS232-driver that is included in your application by the UNIGATE® FC.

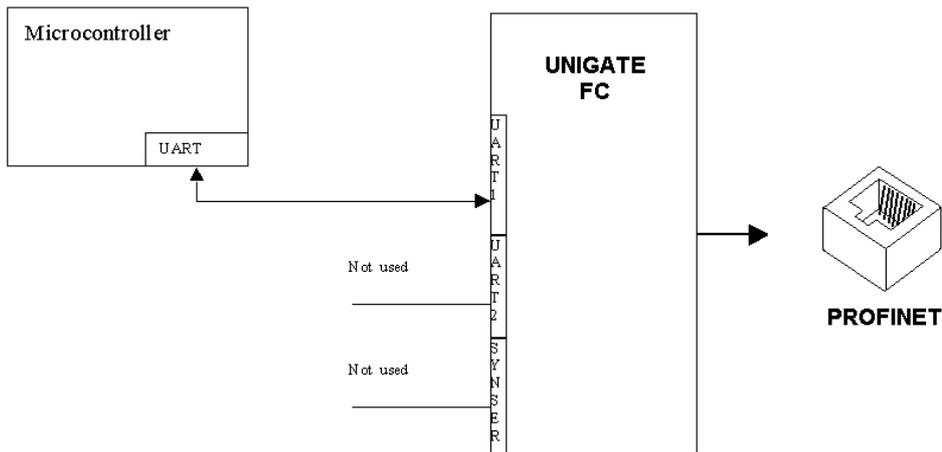


After the RS232-driver has been replaced by the UNIGATE® FC, PROFINET is available at an RJ45-socket.

### 3.5 Connection examples

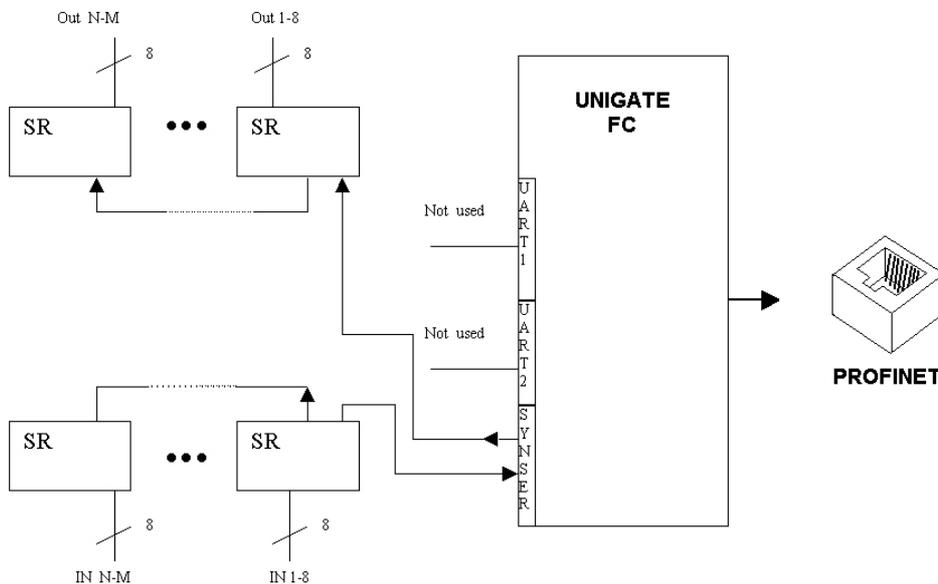
Here you will find some advise that offers help for a re-design. In the following several versions are listed, that should make it easier for you to decide.

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



The UNIGATE<sup>®</sup> FC independently processes the communication with the customer's device via the TTL-interface.

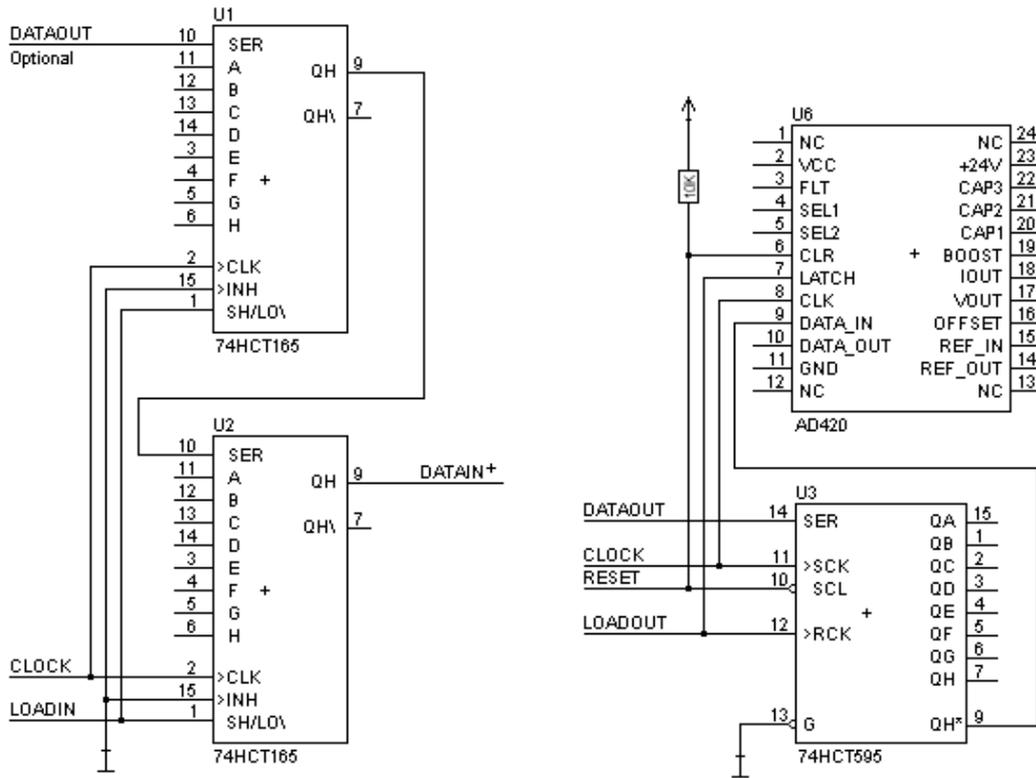
Version 2: Use of UNIGATE<sup>®</sup> FC for digital or analog I/O-modules



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 FC.

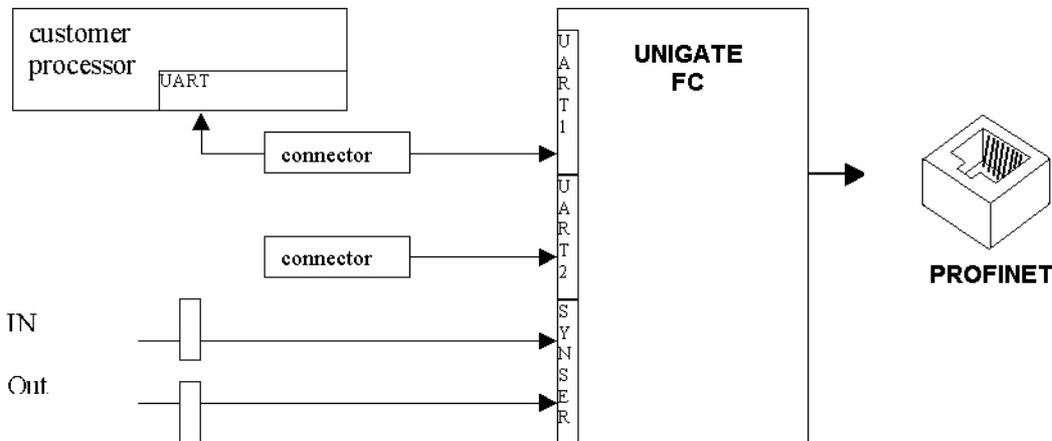


Version 3: Example for digital I/Os

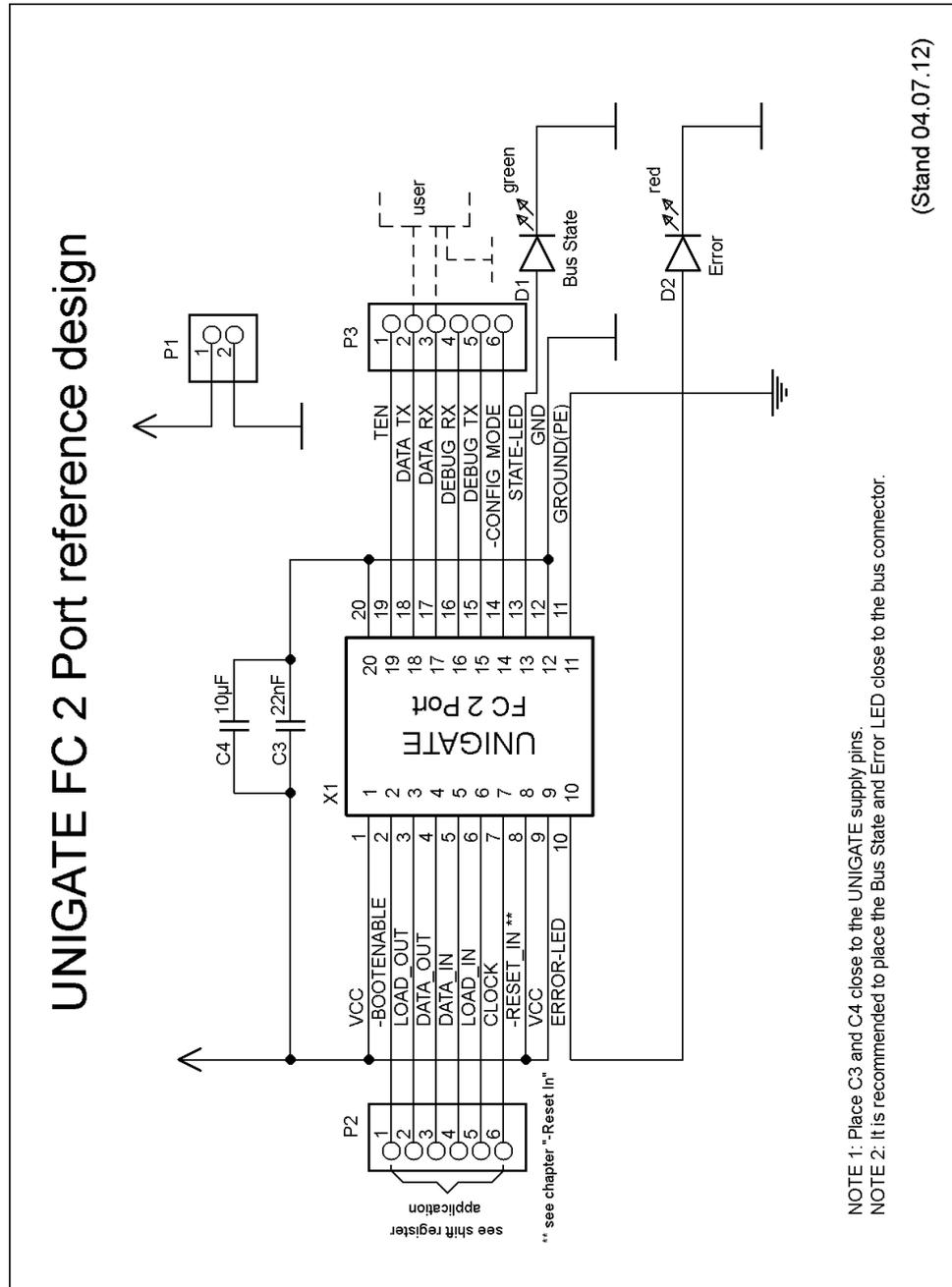
The serial synchronous and the asynchronous interface as well can be operated by UNIGATE® FC at the same time. Here, the possibility results that an existing application can be extended by additional digital or analog I/Os.

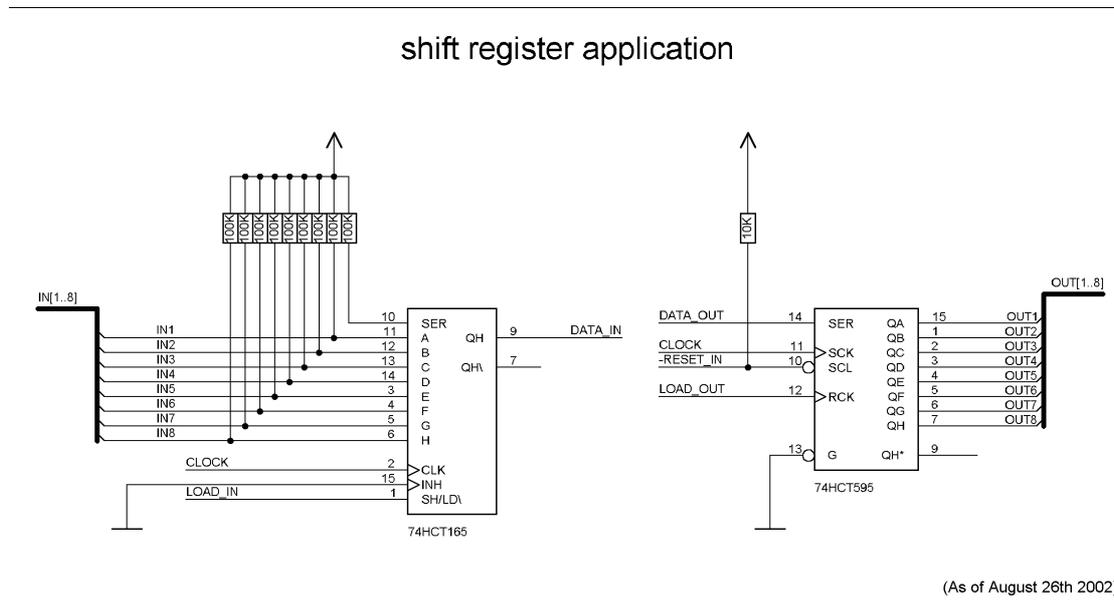
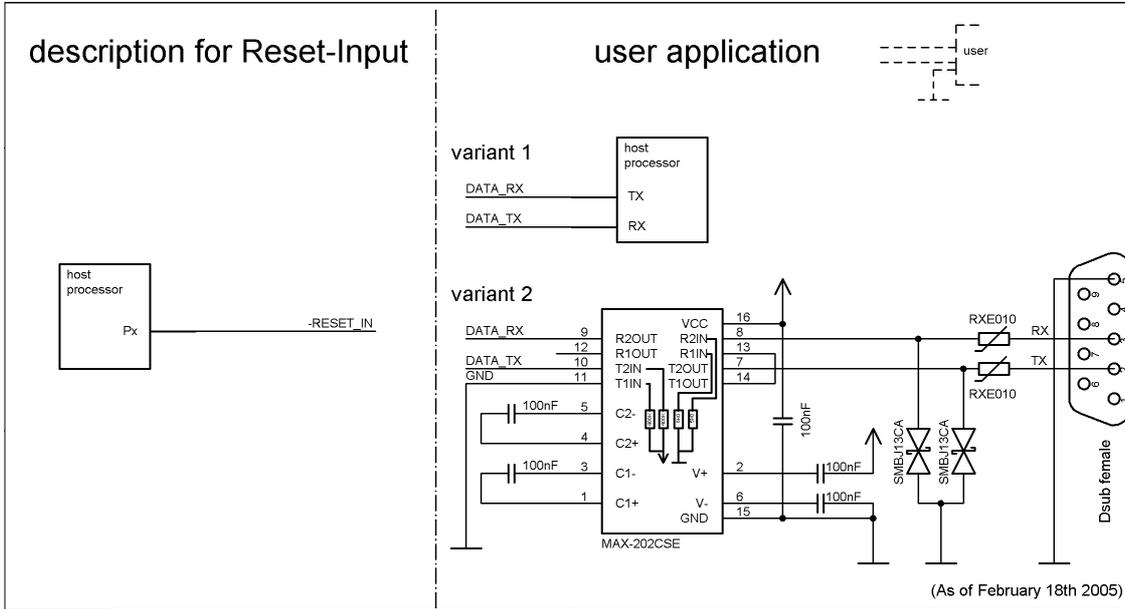
In chapter 5.2 you find an example for a script, that operates these I/Os.

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.



3.6 Layout examples





### 3.7 Handling (mounting the UNIGATE® FC 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® FC's installation on the carrier board:

- Mounting on a socket in the carrier board. If necessary solder the UNIGATE® FC to 2 or 4 pins in the socket. Normally the FC can easily be pulled out after the soldering points have been removed.
- Mounting via hole (drilling 3 mm) in the FC board. At the same time a good PE connection is obtained.
- Make arrangements for two holes next to the socket in the layout. After the UNIGATE® FC was plugged in the socket pull an isolated wire over the FC and solder it on the carrier board at the specified holes.
- Fasten the UNIGATE® FC 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® FC. 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.

## 4 The serial interface

### 4.1 Overview

The serial interface is the most important connection between the UNIGATE® FC 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® FC's serial interface constitutes the basis for it. The UNIGATE® FC allows to connect of controllers with a baudrate of 110 baud to 520 kbaud. 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® FC, 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 FC.

In any case the FC will take over the communication with the Fieldbus independently.

### 4.2 Buffer sizes at the UNIGATE® FC

UNIGATE® FC features a buffer at the serial side with the size of 1024 byte for input data and output data each.

The FIFO of the application interface (RS-interface) can be changed in any Gateway form Script revision 26 on, that is capable for Script. For it please check in the Protocol Developer under „Device Control“-„Hardware“.

### 4.3 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.

### 4.4 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.

### 4.5 Further operation modes

In the configuration mode the serial interface also serves to configure the standard software or to carry out a Script-update. More details on that can be found in chapter 7 on page 24.

## 5 Synchronous serial interface

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

It allows

- the expansion of the FC 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 approximately 45 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 3.5.

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

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

---

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

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

// 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 between 1 and 5 MHz, depending on the hardware. For details please see the IC-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<sup>®</sup> FC 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<sup>®</sup> FC (power up) the firmware will output the binary character 0 (0x00) after a self-test was carried out on this interface. If the FC 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 online-documentation for the Protocol Developer as well as in the instruction manual Protocol Developer.

## 7 Script and configuration

### 7.1 Overview

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

### 7.2 The configuration mode

If the pin „ConfigMode“ is pulled to GND during the PowerUp or Reset, then the UNIGATE® FC starts in the configuration mode. In this mode it is possible to communicate with the FC without processing the regular software. In this mode it is possible to change the UNIGATE® FC's settings of the standard software or to write a new script in the UNIGATE® FC. It shows its start in the configuration mode by issuing a status message (it may take up to 1 minute), which might look as follows:

```
FC-PN-SC V2.36 [36](c)dA Script(C:51/16320,V:0/8192)=""Leer" Author="Deutschmann
Automation GmbH" Version="1.0" Date=21.08.2001 SN=47110001 IP=172.16.48.62 MAC=00-
14-11-15-19-81
```

### 7.3 Update the script

- The preferred version is the one, where the FC is inserted into the basis board, available from Deutschmann Automation and the Deutschmann tools (e. g. software WINGATE with "Write Script" under "File") are used.
- On request the procedures, how to overwrite the included script can be disclosed by Deutschmann Automation in order to automatically replace the script in an application.
- It is also possible to change the Script via FTP (see chapter 9.1 'Script-update via FTP').



**Note:**

**After a change (download of a new configuration or a new Script) it takes up to one minute until the FC has saved this change and will be ready for operation.**

### 7.4 Configuration of the UNIGATE® FC

UNIGATE® FC is delivered with an empty script.

The configuration of the UNIGATE® FC - PROFINET is restricted to the setting of the Ethernet-address.

#### 7.4.1 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.

## 8 Generating a script

### 8.1 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.

### 8.2 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 bytes 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.

### 8.3 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.

### 8.4 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 a PROFINET 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.

### 8.5 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.

## **8.6 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 FTP-Server

This UNIGATE® features an integrated FTP-Server, which allows access to the file system of the device. The FTP-Server is password protected. With the predefined user name "deutschmann" and the password "deutschmann" a connection can be established. Ever since the Firmware-Version 2.3.6 the login data can be defined by setting the Script parameters „FTP\_Password" and „FTP\_UserName".

**Note:** On write access to the file system the actual write process to the non-volatile memory is taking place delayed (5 seconds after the last write access to the file system or when disconnecting the FTP connection). Before restarting the device one must wait for the write process to the non-volatile memory to complete to prevent data loss.

### 9.1 Script-update via FTP

In the Protocol Developer save the compiled script (DCS file) using the file name "script.dcs". Transfer the file to the UNIGATE®. After completion of the write process, please disconnect from the FTP server and restart the device. Upon device startup the UNIGATE® takes over the script contained in the file and deletes the file afterwards.

### 9.2 System configuration update via FTP

In WINGATE save the configuration (GWC file) using the file name "system.gwc". Transfer the file to the UNIGATE®. After completion of the write process, please disconnect from the FTP server and restart the device. Upon device startup the UNIGATE® takes over the configuration contained in the file and deletes the file afterwards.

## 10 WEB-Server

This UNIGATE® has an integrated web server. The delivery condition has a HTML-Page in the file system, which shows the system parameters of the device.

The Web Page can be changed by the customer as desired, or replaced with a self-created web page. It can be written into the file system via FTP (see chapter 9).

### 10.1 Server Side Includes (SSI)

„Server-side includes“ functionality (SSI) is available in order to generate dynamic web pages, which for example can portray process data.

The web server only scans the index.html file and files with the extensions "shtml" or "sml" for SSI directives. In all other HTML files the SSI directives are ignored.

If a client is asking for a web page with SSI directives the web server replaces the directives with the web server's instructions with the appropriate values and sends the generated page.

SSI directives are listed within HTML comments and have the following syntax:

```
<!--#exec="command"-->
```

The following commands are a selection of supported commands.

A complete list of all commands can be found in the Online help of the Protocol Developer.

Command	Instruction (incl. Commentary) is replaced with
DisplayFWVersion	Firmware version
DisplayBLVersion	Bootloader version
DsisplaySerial	serial number
DisplayMacID	Ethernet MAC-adress
DsplayDeviceName	Device name
DisplayStationName	
DisplayStationType	Station Type
DisplayVendorID	Vendor ID
DisplayDeviceID	Device ID
DisplayIP	IP-adress of the UNIGATE®
DisplaySubnet	IP-subnet mask
DisplayGateway	IP-adress of the Default Gateway

To issue any process data, additional Script variables can be given to the Web-Server with the Script-Commands „SetHTMLOutputString“. The command assigns the variable a name, which then can be used in the following command:

Command	Instruction (incl. Commentary) is replaced with
DisplayScriptVar, Variablennam	content of variable

## 10.2 HTML-Formulare

The supporting of HTML Forms allows the Web Server to design interactive web pages. So, for example, input forms for process values can be realized. The script command „SetHTMLInput-String“ assigns a name to a script variable, which then can be used in the HTML form.

## 10.3 Examples

The interaction between Gateway-Script and HTML page can be taken from the example HTML page „ssi.shtml“ and the example Script „example\_Set\_HTML\_String.dss“. Both files are accessible in the download area of our website [www.deutschmann.com](http://www.deutschmann.com).

## 11 File System

The UNIGATE® has a file system which is being used by the integrated FTP-Server and the integrated Web-Server (view chapter 10). Furthermore it's also used by some devices for storing of device parameters.

If necessary the device parameter files can be created and initialized by the UNIGATE® automatically. The following table shows an example of the contents of the file system in delivery condition (contents may vary depending on the device, subject to change without prior notice).

**Table 1 File system content at delivery (example)**

File	Remark
devname.txt	device name
nvddata	non-volatile device parameter
bg.jpg	device web site
bg_container.jpg	
bg_footer.jpg	
bg_topnavi.gif	
bottom.gif	
da-logo.gif	
deutschmann.css	
down.gif	
favicon.ico	
h4_pre.gif	
header_schatten.jpg	
headline.jpg	
index.html	
line.gif	
profinet.gif	
rnavmain_bg.gif	
ssi.html	
teaser_header.jp	

**Please note that**

- files can not be overwritten or edited on the device. If you want to replace a file with a modified version, you must delete the file stored on the device first, and write the modified file again or transfer to the device.
- during a writing procedure, the execution of the firmware will cut out for a few seconds (depending on the length of the file). The access to the file system is buffered, so the actual writing access is delayed by several seconds.

It is possible to reformat the file system via FTP. It will lose all data stored in the file

system. To perform the formatting of your command-line please use the FTP client on Microsoft® Windows® (Command Prompt -> ftp.exe). Make a connection to the FTP server of the UNIGATE® and then run the command "del\*.\*". The string "\*.\*" is understood as a request by UNIGATE® to format the file system.

## 12 PROFINET

If we're speaking of PROFINET, then the 100 Mb/s-version is automatically spoken of. Full-duplex and switched Ethernet. The end device's protocol is being converted in the UNIGATE® via a Script.

The UNIGATE® supports PROFINET I/O-RT.

### 12.1 Data exchange mode

The Gateway has to be in the data exchange mode, so that a data exchange between the RS-side of the gateway and the fieldbus is possible. This mode is always active as long as the gateway is not in the configuration mode, the firmware update- or Debugmode. In the data exchange mode the Gateway will carry out the downloaded Script.

### 12.2 PROFINET address-assignment

Note: The Gateway does not have an assigned IP-address in delivery condition!

During normal operation (data exchange mode) the IP-address is assigned to the Gateway by the PROFINET-IO-controller (SPS). The Gateway also has a device name that is used to communicate with. The IP address can also be assigned manually or with a DHCP-Server.

### 12.3 PROFINET Device Name

Note: In delivery condition the Gateway is not assigned a device name!

The device name is assigned via the configuration software of the Gateway. Alternatively the device name can be changed via FTP (File „devname.txt“) or the Script.

According to PROFINET specification the following rules apply:

- It consists of one or more names separated by a period pieces.
- The total length is 1 to 240 characters.
- The length of a part of the name is 1 to 63 characters.
- A part of the name consists entirely of lowercase letters, numbers and the hyphen.
- Neither the first nor the last character of a part of the name is a hyphen.
- The first part of the name does not begin with "port-xyz" or "port-xyz-abcde", where as a, b, c, d, e, x, y and z are numerals.
- The name does not have the form "k.l.m.n", where k, l, m and n are numbers between 0 and 999.

### 12.4 Project planning

Use any project planning tool for project planning.

A sample GSD-file (that is based on the XML-oriented standard GSDML) can be found on the Internet ([www.deutschmann.com](http://www.deutschmann.com)) free of charge. The generating of the GSD-file is also provided as a service from Deutschmann Automation.

## 12.5 Possible data lengths

The table below shows the maximum transferable data:

Input data	max. 1440 bytes	Variable: maximum value in this case
Output data	max. 1440 bytes	Variable: maximum value in this case

## 12.6 Startup phase

The Gateway is parameterized and configured by the PROFINET-IO-controller during the startup phase. Only after a correct termination of the startup phase the data exchange with external devices will take place.

## 12.7 Literature

We recommend the book "Der Schnelleinstieg in PROFINET", author M. Popp and K. Weber, to help you quickly get to grips with the subject of PROFINET. The book (German language) can be ordered from the PROFIBUS User Organisation, Order No. 4.181.

Address:

PROFIBUS Nutzerorganisation e.V.

Haid-und-Neu-Str. 7

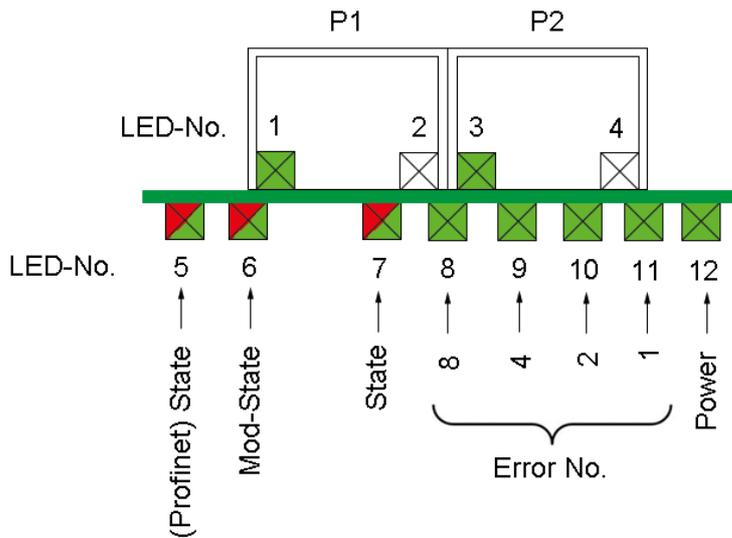
D-76131 Karlsruhe

Germany

Tel: +49 721 9658 590

## 13 Hardware-ports and light-emitting diodes (LED)

### 13.1 View



### 13.2 Connectors

#### 13.2.1 PROFINET connectors

The connectors for the connection to PROFINET is located on the upper side of the device. Pin assignment P1 + P2 (2 x RJ45)

Pin No.	Name	Function
1	TD+	Transmission line +
2	TD-	Transmission line -
3	RD+	Receive line +
4		
5		
6	RD-	Receive line -
7		
8		

### 13.3 Light indicator

The UNIGATE® FC - PROFINET 2Port Gateway has 12 LEDs with the following significance:

1	LED Link / Act. P1	green	Ethernet-link-pulse found / network data Port 1
2	n.c	-	no function
3	LED Link / Act. P2	green	Ethernet-link-pulse found / network data Port 2
4	n.c	-	no function
5	LED (Profinet) State	red/green	Interface state Profinet
6	LED Mod-State	-	no function
7	LED State	red/green	customer defined / general Gateway error
8-11	LED 1 / 2 / 4 / 8 (Error No.)	green	customer defined / general Gateway error
12	LED Power	green	supply voltage

#### 13.3.1 LED 1 „Link / Act. P1“

This LED is controlled directly by the PROFINET-processor and illuminates if the Gateway at Port 1 (P1) is located in an operational network (it receives link pulses) and flickers at network traffic.

#### 13.3.2 LED 3 „Link / Act. P2“

This LED is controlled directly by the PROFINET-processor and illuminates if the Gateway at Port 2 (P2) is located in an operational network (it receives link pulses) and flickers at network traffic.

#### 13.3.3 LED 5 „(PROFINET) State“

off	waiting for Script-Command „Bus-Start“
green, illuminated	Connection with IO-Controller, data exchange activ
green, flashing	PROFINET initialised. Waiting for connection with IO-Controller
red, flashing	PROFINET flash indication

#### 13.3.4 LED 7 „State“

green illuminated	controlable via Script
green, flashing	controlable via Script
green/red, flashing	controlable via Script
red, illuminated	general Gateway Error (see LED's Error No.), über Script steuerbar
red, flashing	UNIGATE® is in configuration mode/test mode, controlable via Script

### **13.3.5 LED 8-11 „1 / 2 / 4 / 8 (Error No.)“**

If these 4 LED's are flashing and the LED „State“ is illuminated red at the same time, binary coding is happening (for conversion table view Appendix) according to the table in chapter „Error handling“. Additionally these scripts are controllable via Script.

### **13.3.6 LED 12 „Power“**

This LED is connected directly to the supply voltage of the UNIGATE® FC.

## 14 Error handling at UNIGATE® FC

If the Gateway recognizes an error it will be shown with the LED 7 „State“ being illuminated red and at the same time the error number according to the table over the LEDs 8-11, will show „Error No.“.

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 or wrong Fieldbus ID
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® FC

## 15 Firmware-Update

### 15.1 Overview

UNIGATE® FC has an internal memory for the firmware. It can be replaced via the UNIGATE® FC's Ethernet-(PROFINET)-interface.

### 15.2 Execution of the firmware-update

The safest way for the firmware-update is the use of the basic board combined with the software "FDT.EXE" (firmware-download-tool). These tools are available from Deutschmann Automation (see chapter 17 on page 41).

A Firmware-update can be carried out in the following operational modes:

Configuration mode (via the IP-address UNIGATE®)

Firmware-update-mode (via the fixed IP-address 10.10.10.10)

### 15.3 Adjusting the firmware-update-mode

#### 15.3.1 Adjustment by hardware

UNIGATE® FC can be brought to the firmware-update-mode by the hardware. For it the signal  $\overline{BE}$  (boot enable) has to be pulled to the potential GND during the Power-up-process.

### 15.4 Note on safety

The firmware-update should only be carried out when there is no other possibility left. A firmware-update-process that has already been started CANNOT be undone. With it the previously used firmware is permanently unusable.

### 15.5 Operation mode of the FC

#### Data exchange mode

This mode is required for the regular use of the FC. In this mode the FC will process all script-commands and normally exchange the corresponding user data. The bus as well is operated in this mode through the FC.

#### Configuration mode

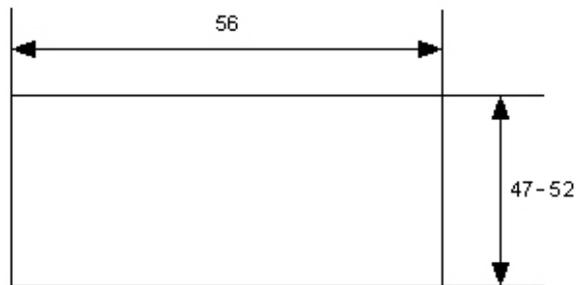
In the configuration mode the UNIGATE® FC will carry out a self-test after the start (or after a reset). After a successful self-test it will wait for further commands. Here it is possible to load a translated Script into the unit or to configure the device.

## 16 Technical data

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

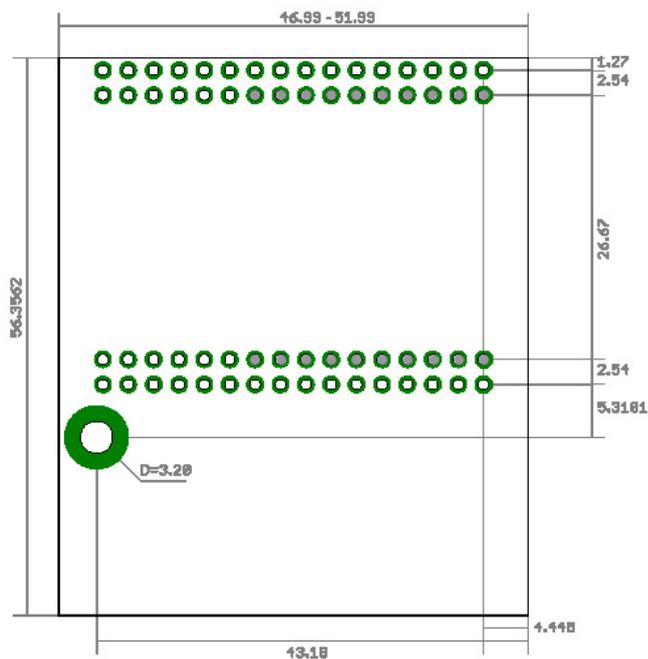
### 16.1 Mechanics of the UNIGATE® FC

#### 16.1.1 General dimensions of UNIGATE® FC



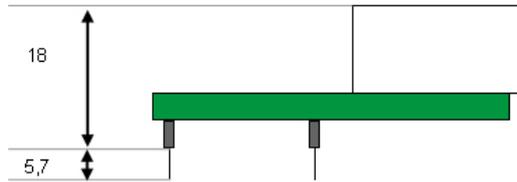
Max. height = 24 mm; incl. pin

#### 16.1.2 Dimension drawing UNIGATE® FC



The pins of the UNIGATE® FC - PROFINET 2Port are arranged 2, 54 mm in the contact pitch.

### 16.1.3 Height of the UNIGATE® FC



### 16.2 Technical data UNIGATE® FC PROFINET

Characteristics	Explanation
Supply voltage	3.3V ± 5 %, max. 550 mA DC
Interface	2 UART interfaces, 1 synchronous serial interface
IP-address	Assigned by the Profinet-IO-controller (SPS)
Bus-baud rate	100 MBaud
UART-baud rate	Up to 520 Kbaud (adjustable via script)
Supported Ethernet protocols	UDP/IP, TCP/IP
Technology	32-bit-processor
Others	I/Os, analogous signals, shift registers, LEDs, switches and so on can be connected externally
Dimensions	47-52 mm x 56 mm x 18 mm (WxDxH)
Weight	Approx. 25 g
Operating temperature	-40°C ..+85°C
Storage / transport temperature	-40°C..+100°C
Built-in position	Any

## 17 Accessory

The following tools are available from Deutschmann Automation.

### 17.1 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 FC. The software describes the procedure of an update itself.

### 17.2 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.

### 17.3 Developerkit UNIGATE® FC

The Developerkit FC contains

- a Developerboard UNIGATE® FC, Article number V3852e (see chapter 17.3.1)
- a plug-in power pack to supply the Developerboard
- connection cables for appl. RS232, Debug RS232 and appl. RS422/485
- USB-cable
- Software and documentation to complete the packet.

#### 17.3.1 Developerboard UNIGATE® FC

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



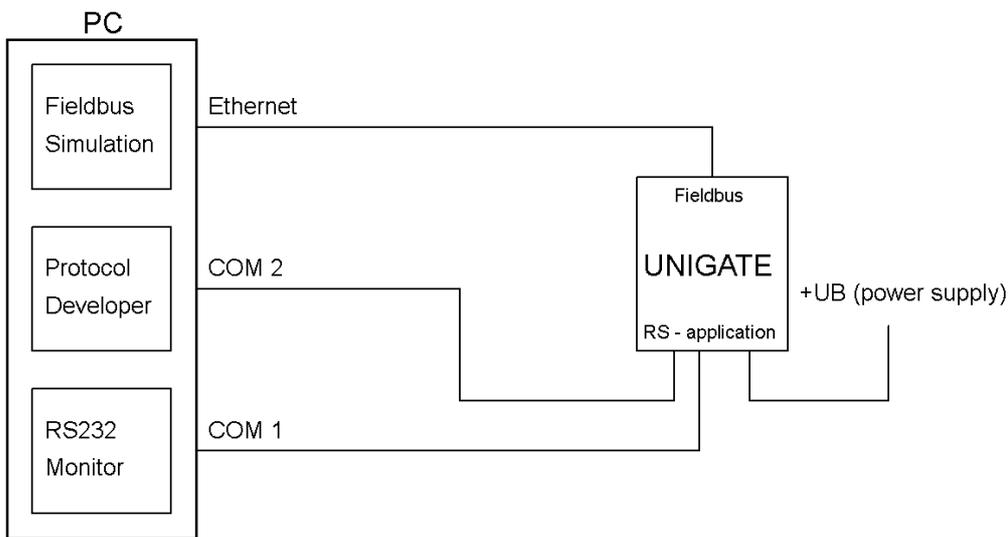
The required UNIGATE® FCs are ordered separately. An RS232-interface or a USB-connection is available for the connection to the PC (Debug-interface).

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).

### 17.3.2 Quick start



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

## 18 Appendix

### 18.1 Explanations of the abbreviations

#### General

CL	=	Product group CL (Compact Line)
CX	=	Product group CX
FC	=	Produktgroup FC (Fast Connect)
GT	=	Galvanic separation RS-side
GY	=	Housing color gray
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
IC	=	Product group IC (IC-design DIL32)
16	=	Script memory expanded to 16KB
5V	=	Operating voltage 5V
3.3V	=	Operating voltage 3.3V

#### Fieldbus

ASI	=	AS-Interface (AS-i)
BI	=	BACnet/IP
BMS	=	BACnet MSTB
CO	=	CANopen
C4	=	CANopen V4
C4X	=	CANopen V4-version X (see comparison table UNIGATE® IC for the respective product)
DN	=	DeviceNet
EC	=	EtherCAT
EI	=	Ethernet/IP
FE	=	Ethernet 10/100 MBit
FEX	=	Ethernet 10/100 MBit-version X (see comparison table UNIGATE® IC for the respective product)
IB	=	Interbus
IBL	=	Interbus
LN62	=	LONWorks62
LN512	=	LONWorks512

MPI	=	Siemens MPI®
PL	=	Powerlink
PN	=	PROFINET-IO
PBDP	=	PROFIBUS DP
PBDPL	=	PROFIBUS DP-version L (see comparison table UNIGATE® IC for the respective product)
PBDPX	=	PROFIBUS DP-version X (see comparison table UNIGATE® IC for the respective product)
PBDPV0	=	PROFIBUS DPV0
PBDPV1	=	PROFIBUS DPV1
RS	=	Serial RS232/485/422

## 18.2 Hexadecimal table

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

## 19 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 keep the following information ready for your call:

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

You can reach us during hotline hours which are as follows

Monday to Thursday from 8 am to noon and from 1 pm to 4 pm, Friday from 8 am to noon. (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 hotline +49 6434 9433-33

Fax sales department +49 6434 9433-40  
Fax technical hotline +49 6434 9433-44

E-Mail technical hotline [hotline@deutschmann.de](mailto:hotline@deutschmann.de)

### 19.1 Downloading PC software, GSDML-file, Script examples etc.

You can download the current version of WINGATE<sup>®</sup>, a topical sample-GSDML-file and Script examples free of charge from our Internet server.

Here you will also find the software tool Protocol Developer for UNIGATE<sup>®</sup> CL / FC / IC and SC.

<http://www.deutschmann.com>



