



Deuschmann

your ticket to all buses

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



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

1	General introduction	10
2	The UNIGATE® IC	11
2.1	Technical introduction	11
2.2	Availability	11
2.3	Firmware	11
2.4	The serial standard interface	11
2.5	The synchronous serial interface	11
2.6	The Debug-interface	11
2.7	UNIGATE® IC2 hardware survey	12
3	Hardware design.	13
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	Software	15
3.4	Basic line of proceeding	16
3.5	Connection examples	17
3.6	Layout-Beispiele	19
3.7	Handling (mounting the UNIGATE® IC2 on the carrier board)	21
3.8	Requirement for Ethernet-Interface	22
4	The serial interface	23
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
5	Synchronous serial interface	24
5.1	Shift register operation	24

5.1.1	Example-Script	24
5.2	SPI mode	25
5.2.1	Example-Script	25
6	The Debug-interface	26
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
7	Script and configuration	27
7.1	Overview	27
7.2	The configuration mode	27
7.3	Update the Script	27
7.4	Configuration of the UNIGATE® IC2	30
7.4.1	Ethernet	30
7.4.2	RS232/RS485/RS422	30
8	Generating a Script	31
8.1	What is a Script?	31
8.2	Memory efficiency of the programs	31
8.3	What can you do with a Script device?	31
8.4	Independence of buses	31
8.5	Further settings at the gateway	32
8.6	The use of the Protocol Developer	32
8.7	Accuracies of the baud rates at UNIGATE® IC2	32
8.8	Script processing times	32
9	Ethernet	34
9.1	Network-addresses, network-connections	34
9.2	TCP-connection setup	34
9.2.1	TCP-Server channel mode	34
9.3	Data exchange	34
9.3.1	TCP or UDP?	34
9.4	Data exchange mode	34
9.4.1	Direct data exchange	35
9.4.2	MODBUS-TCP.	35
9.4.3	Server	35
9.4.3.1	Webserver.	35
9.4.3.2	FTP-server	35
9.4.3.3	Script update via FTP	37
9.4.3.4	System configuration update via FTP	37
9.4.3.5	File system	37

9.4.3.6	TCP server channel mode	37
9.4.3.7	UDP channel mode	37
9.4.3.8	Raw-channel mode	37
9.4.3.9	BusStatus	37
9.5	Setting the Ethernet-address	37
10	Error handling at UNIGATE® IC2	39
11	Firmware-update.	40
11.1	Overview	40
11.2	Adjusting the firmware-update-mode	40
11.2.1	Adjustment by hardware	40
11.2.2	Adjustment by software	40
11.3	Execution of the firmware-update	40
11.4	Note on safety	40
11.5	Operation mode of the IC2	40
12	Technical data	41
12.1	Mechanics of the UNIGATE® IC2	41
12.1.1	General dimensions of UNIGATE® IC2	41
12.1.2	Dimensions UNIGATE® IC2 (Fast Ethernet only).	41
12.2	Technical data UNIGATE® IC2 - Fast Ethernet	42
13	Accessory	43
13.1	Adapter RS232	43
13.2	Adapter RS485	43
13.3	FirmwareDownloadTool (FDT)	43
13.4	Protocol Developer	43
13.5	Developer Kit UNIGATE® IC-AB IC	43
13.5.1	Developer Board UNIGATE® IC-AB	43
13.5.2	Quick start	44
14	Appendix	45
14.1	Explanations of the abbreviations	45
15	Servicing	47
15.1	Returning a device	47
15.2	Downloading PC software	47

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 – 2020. 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[®] IC2 by Deutschmann Automation any more. All digital functions, software, stack and driver as well as optocoupler are integrated on a UNIGATE[®] 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[®] 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[®] 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[®] IC2 is a further development of the 1st series of the UNIGATE[®] IC. Equipped with a Cortex-M4 controller, the UNIGATE[®] 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 µs.

The UNIGATE[®] IC2 is also PIN-compatible with the 1st series of the UNIGATE[®] IC.

2 The UNIGATE® IC

2.1 Technical introduction

The UNIGATE® IC2 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® IC2 and the required plug connectors.

2.2 Availability

The module is available as Ethernet. Further fieldbuses are either planned or being worked on. They will only differ in the connections of the busses. The meaning of the general pins 1 - 9 as well as 24 and 26 - 32 will remain unchanged also for further fieldbus implementations.

You can find an up-to-date list for all UNIGATE® IC2s under:
<http://www.deutschmann.com>.

2.3 Firmware

UNIGATE® IC2 is programmed via Scripts. On principle any Script, that has been developed for a UNIGATE® IC2, can also be operated on the UNIGATE® IC2.

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® IC2. For more information on this serial interface see chapter 4.

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 IC2 is also operable autonomously without that controller.

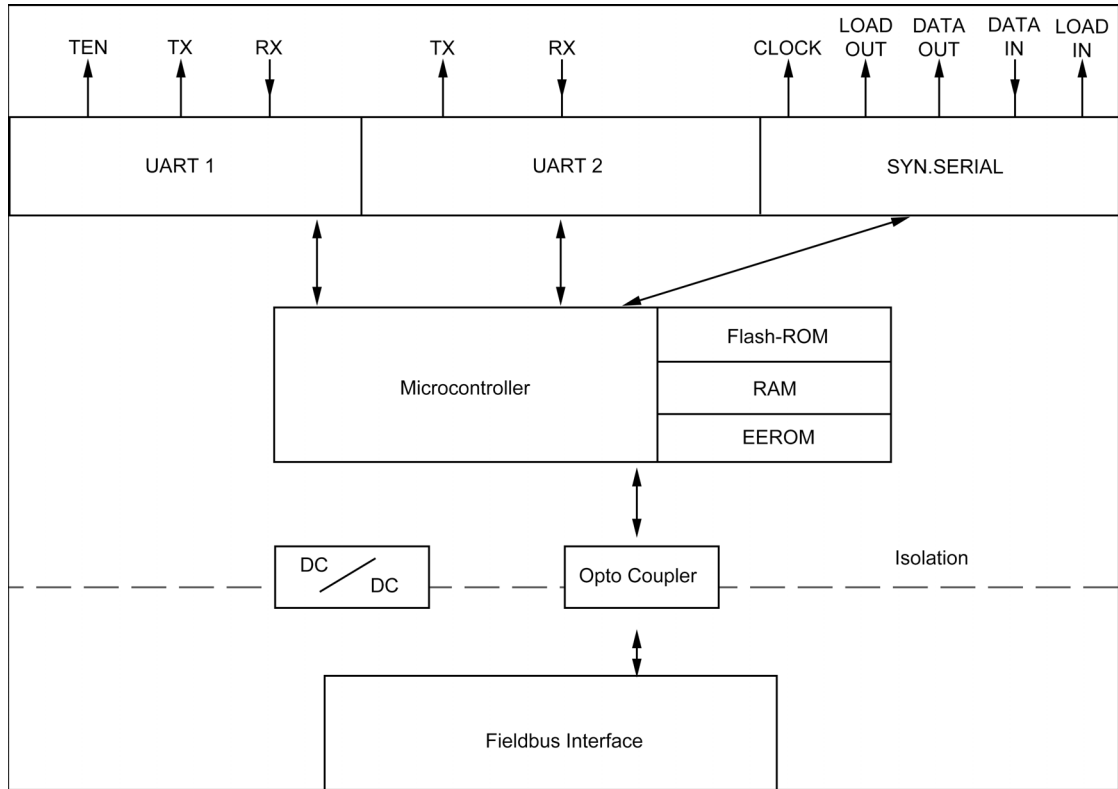
2.6 The Debug-interface

The UNIGATE® IC2 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.

All interfaces can independently be used at the same time.

2.7 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 12.



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 _{Logic}	-BE	boot enable	
3	OUT _{Driver}	Load out (SS0-)	strobe signal for synchronous, serial interface	
4	OUT _{Driver}	Data out (SS1-)	output data for synchronous, serial interface	
5	IN _{Logic}	Data in (MISO)	input data of the synchronous, serial interface	internally pulled up with 10 kΩ
6	OUT _{Logic}	Load in (MOSI)	input data of the synchronous, serial interface; alternatively strobe signal of the output data	internally pulled up with 10 kΩ
7	OUT _{Driver}	Clock (SCK)	clock pulse signal for synchronous, serial interface	
8	IN _{Reset}	-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 _{Logic}	-Config Mode	signal to start the configuration mode	internally pulled up with 10 kΩ
27	OUT _{Logic}	DbgTX	serial Debug TX	

28	IN _{Logic}	DbgRX	serial Debug RX	internally pulled up with 10 k Ω
29	IN _{Logic}	RX	serial data RX	internally pulled up with 10 k Ω
30	OUT _{Logic}	TX	serial data TX	
31	OUT _{Logic}	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 _{IL}	V _{IH}
IN _{Reset}	< 0,7V / 0,5mA	>2,3V / 5 μ A
IN _{Logic}	< 1V / 5 μ A	>2,3V / 5 μ A
	V _{OL}	V _{OH}
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 11.

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 μ s.

3.2.8 LED-LINKACT

A green LED can be connected to that line (see chapter 3.6). It is glowing when the Gateway is connected to a working net (link pulses are being received) and it is flashing in case of network data traffic (Activity). No difference between 10 and 100 MBaud is made.

3.2.9 Error-LED

This line is not supported yet and it is only meant for new future projects.

3.2.10 LED-FE

A red LED can be connected to this line (see chapter 3.6). It is glowing in the state of "Ethernet ok" and in the state "Config-mode" it is flashing.

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

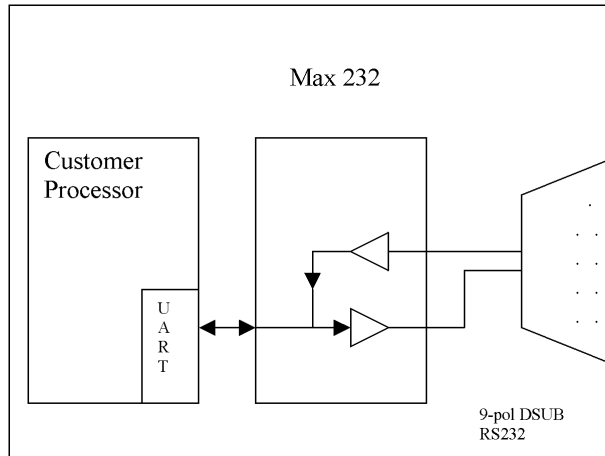
Transmission and receive line of the serial interface. This interface is programmable in accordance with the description in chapter 4.

3.3 Software

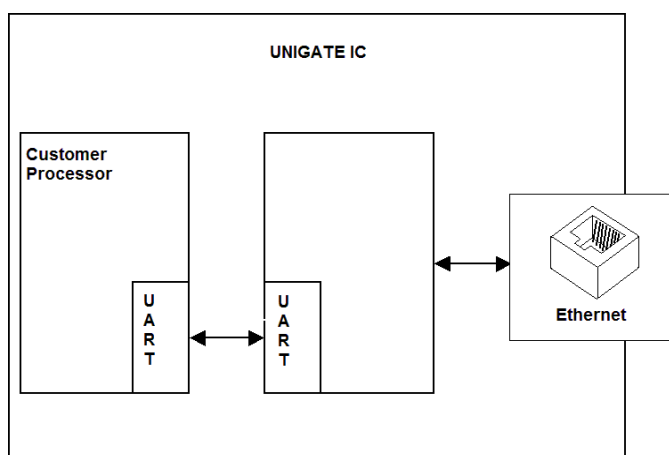
The software executes Script-commands, which in turn control the IC2's hardware and they process their complete protocol by software. The Script itself can be generated by the company Deutschmann Automation or with the software Protocol Developer by yourself. For a detailed description of the Script 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[®] 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.



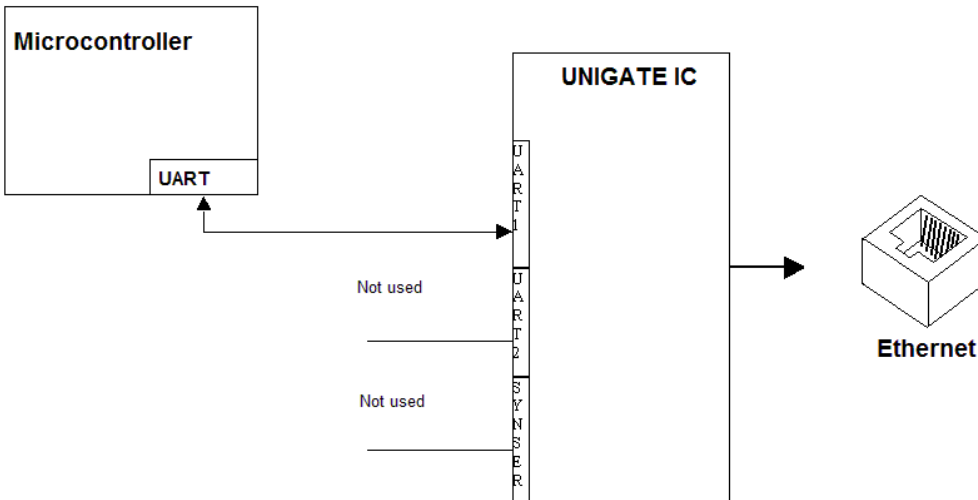
After the RS232-driver has been replaced by the UNIGATE[®] IC2, Ethernet is available at an RJ45-socket.

Deutschmann Automation is also offering an appropriate adapter board. With it existing devices can be adapted without re-design; see chapter 13.

3.5 Connection examples

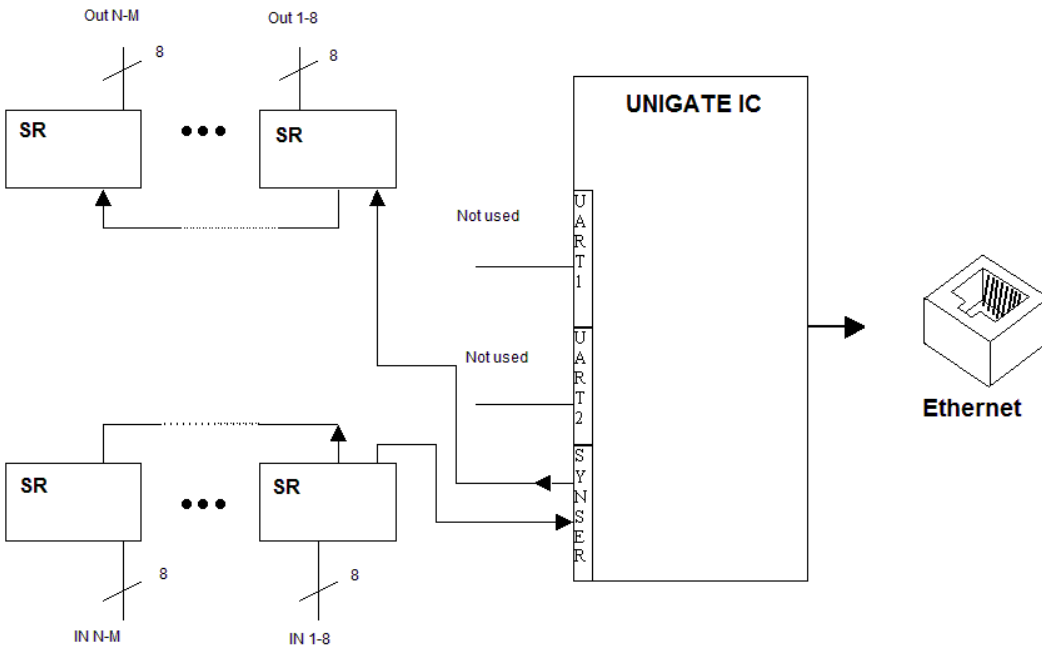
Here you will find some advice that offers help for a re-design. In the following several versions are listed, that should make it easier for you to decide.

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.

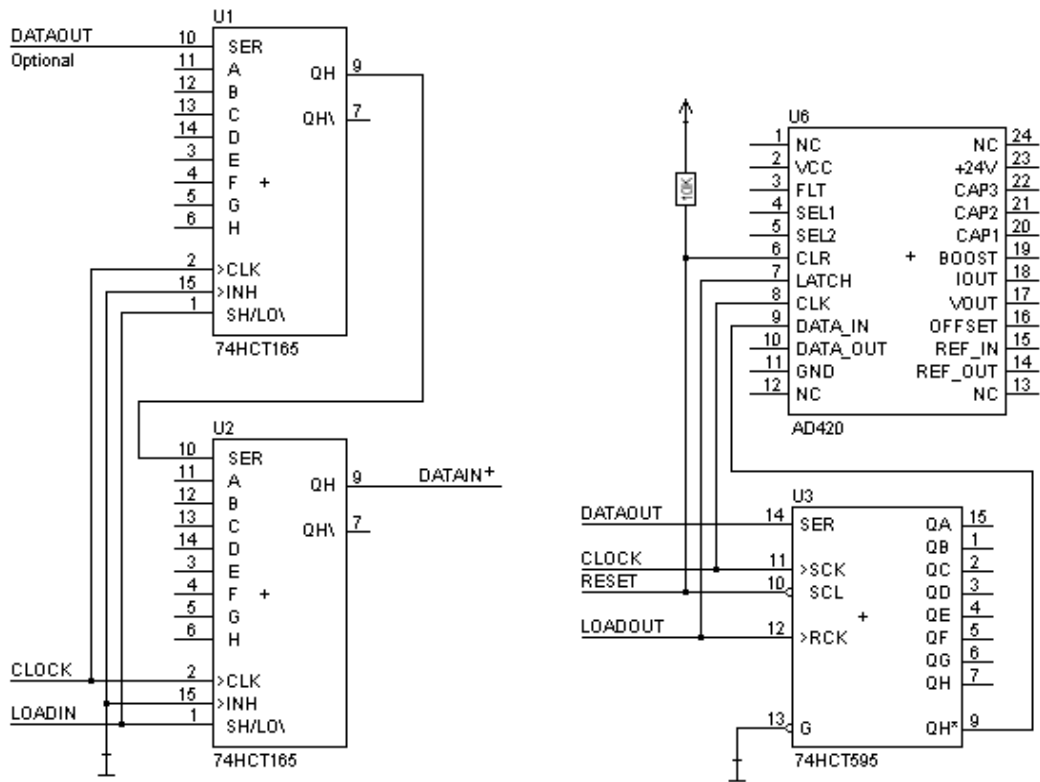
Version 2: Use of UNIGATE® IC2 for digital or analog I/O-modules



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.

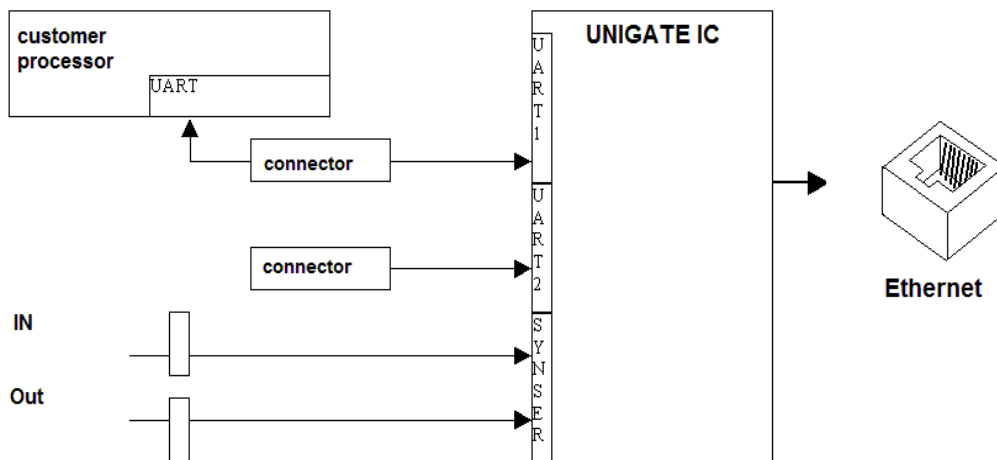


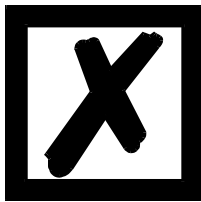
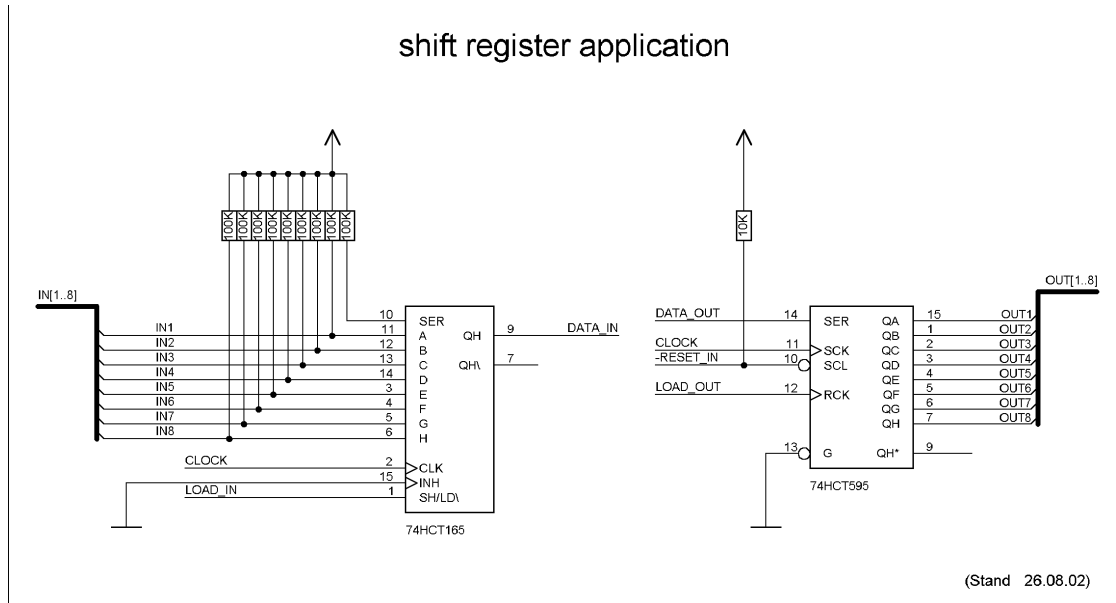
Version 3: Example for digital I/Os

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.





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		

3.7 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.

3.8 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 μ 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. 615 008 140 121

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 625 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® 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 3.5.

```
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 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 Script and configuration

7.1 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.

7.2 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:

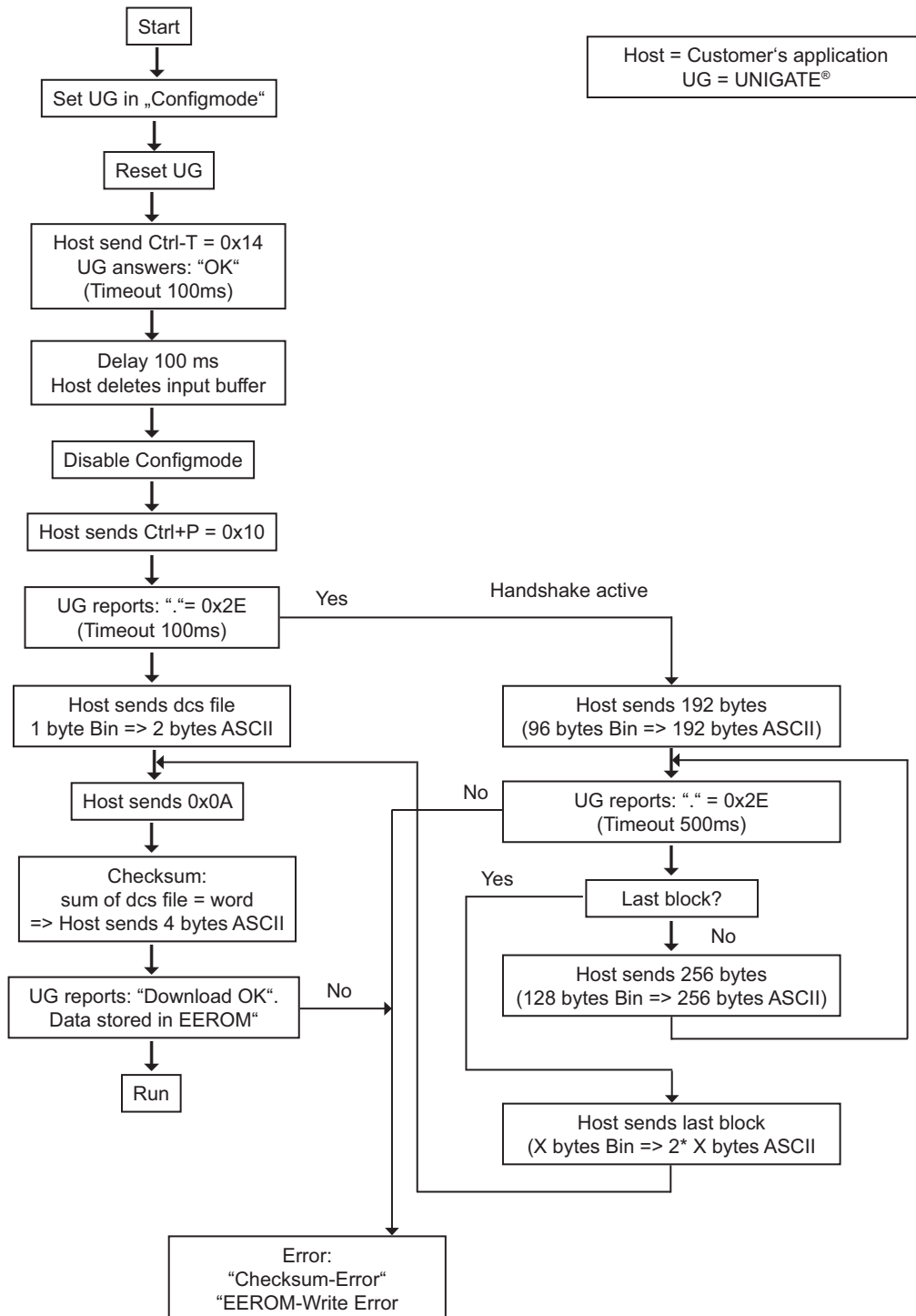
```
IC2FE V1.0.0[44] (c)dA[120MHz] Script(C:190/3968,V:118/16384)=""Leer" Author="Deutschmann Automation GmbH" Version="1.0.2" Date=06.11.2019 SN=47110001 IP=0.0.0.0 MAC=00-14-11-01-02-03 Data-Flash=1MByte
```

The configuration of a UNIGATE® IC2 Fast Ethernet is restricted to setting the Ethernet-address (see also chapter 9.5).

7.3 Update the Script

- The preferred option is to insert the UNIGATE® IC2 into the base board from Deutschmann (Developer Board UNIGATE® IC2-AB) and use the Deutschmann tools (software WINGATE with „Write Script“ under „File“ or with the software ScriptProgramTool).
- Your host can also automatically replace the script in your application. In the following flow chart the handshake is shown.

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"

7.4 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.

7.4.1 Ethernet

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

7.4.2 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.

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 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.

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 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.

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.

8.7 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%.

8.8 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 μ 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.

9 Ethernet

Each Ethernet-Client processor with TCP/IP network-protocol in a 10/100 Mbaud Ethernet can exchange data with the gateway.

9.1 Network-addresses, network-connections

After switching on, the gateway is waiting for data communication with a network-participant. In this case the gateway is always prepared to react to ICMP messages (e. g. **PING** Request and Reply). The functions of the ARP-protocol are also available.

9.2 TCP-connection setup

In the characteristic application in the TCP/IP-protocol, after switching on, the gateway is in the state "Passive Open" under an IP-address and port number that was determined in the configuration. A Client processor is in the position to open a TCP/IP connection with the gateway under the port determined in the configuration. After a successful connecting, data can be exchanged.

9.2.1 TCP-Server channel mode

A TCP connection in which the IC2-FE is the server is closed after 65635 seconds (18.2 hours) if no data has been transferred during this time.

9.3 Data exchange

At the protocol TCP/IP, a full duplex data connection is available after establishing a connection between the gateway and the attached Ethernet-Client. The data stream is protected by the TCP/IP-protocol in both directions.

When operating with UDP/IP-protocol, no connection is established before data is transferred. In this case the application is responsible for the control of the data flow as well as for the backup of the data.

9.3.1 TCP or UDP?

TCP is a connection-oriented protocol: Here a fixed logical connection between Client and Server exists in the course of the data communication. TCP features all mechanisms, that are required, in order to open a connection and to guarantee a flawless data transfer across the network and then again close the connection. For that reason the protocol software creates and administers own buffers for each connection. In case a lot of UNIGATE[®]-servers are to be addressed from one processor at the same time, storage limits in the processor might result in an extreme case.

In contrast to TCP, UDP does not have mechanisms available to repeat data packets or to check them on their completeness. The data communication with UDP is to be recommended if a parent protocol guarantees a flawless transmission between the terminal device at the UNIGATE[®]-server and the application on the TCP/IP-station anyhow.

9.4 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. As long as the gateway is not in the configuration mode or the test mode, the data exchange mode is active. In the data exchange mode the gateway will carry out the set protocol with those parameters preset through WINGATE.

9.4.1 Direct data exchange

In the Ethernet the data from the RS-interface are represented unchanged via a TCP/IP connection.

9.4.2 MODBUS-TCP

The Modbus protocol is already available as Script for the serial interface for all Script gateways. Now this additional function can be adjusted at the Fast Ethernet gateway so that the Modbus data can be made available as TCP-Modbus data packet on the Ethernet side.

(MODBUS-TCP).

For this purpose the channel that is used for this feature has to be set up as MODBUS-TCP channel in the Script.

For this function a running Modbus-Script is required.

9.4.3 Server

9.4.3.1 Webserver

Websites can be called up through a communication channel that has been set up as webserver channel via a Script. Here the internal data system is used. Several websites can be placed or rather websites, image files and Java-applets can be called up. Additional connections are opened by the browser for loading the applet and the images.

In addition, it must be ensured that files are available in the file system that can be called up as a website.

The web server runs independently of the script.

The web server is only active in run mode. Not in config mode. The web server can be activated via script using the parameter WebServer (value: 1) and deactivated again (value: 0).

The web server no longer uses any of the 4 possible channels if they are activated.

9.4.3.2 FTP-server

The FTP server is active both in Config.Mode and in Run Mode. A script can be activated via the FTPServer parameter (value: 1) and deactivated again (value: 0).

The FTP server is password protected. A connection can be established with the predefined user name "deutschmann" and the password "deutschmann"

The FTP server no longer uses any of the 4 possible channels if they are activated.

The FTP server does not offer any special support for spaces in file names. However, it is possible to transfer files with spaces in the name to / from the FTP server if the spaces are handled correctly by the FTP client. However, this always depends on the program used.

Umlauts should not be used in file names. A connection to the FTP server can e.g. with command line clients (MSYS64, Windows 10). Below is a list of server commands:

Supported FTP Commands

Code	Command	Description
USER	User Name	Starts login with name identifying the user.
PASS	Password	Continues login with the user's password.
QUIT	Logout	Closes the user connection.
SYST	System	Identifies the operating system at the server.
NOOP	No Operation	Sends an OK reply.
XNOP	No Operation	Same as NOOP
PWD	Print Working Directory	Returns the name of the current working directory.
XPWD	Print Working Directory	Same as PWD
CWD	Change Working Directory	Changes the current working directory of the user.
XCWD	Change Working Directory	Same as CWD
CDUP	Change Directory Up	Changes working directory to the parent of the current directory.
XCUP	Change Directory Up	Same as CDUP
MKD	Make Directory	Creates a sub directory in the current working directory.
XMKD	Make Directory	Same as MKD
RMD	Remove Directory	Removes the directory.
XRMD	Remove Directory	Same as RMD
Type	Representation Type	Supports ASCII and Image types.
PORT	Data Port	Specifies the data port to be used in data connection.
PASV	Passive	Requests the server to listen on a data port and wait for a connection.
LIST	List	Sends a directory listing to the user.
NLST	Name List	Sends a directory listing to the user.
RETR	Retrieve	Sends a file content to the user.
STOR	Store	Saves a captured user file on server.
APPE	Append	Appends a captured user file to an existing file on server.
DELE	Delete	Deletes a specified file from server.
RNFR	Rename From	Specifies the name of existing file to rename (must be followed by RNTO).
RNTO	Rename To	Renames an existing file to new name.
HELP	Help	Returns a list of supported commands.
SIZE	Size	Returns the size of a specified file.
MDTM	Last-modified Time	Returns last-modified time of a specified file.

Verfügbare Funktionen:

PUT = create and write file
 GET = read file
 DIR = read Directory
 DEL = delete file
 DEL *.* = Delete all files and initialize (format) the file system.



Warning:
The Script- and config-file will be deleted also.

Subdirectories are not intended. For that reason you cannot switch to subdirectories and subdirectories cannot be created. For the FTP-server the two channels have to be set up initially in the Script.

The FTP-server is running irrespective of the Script.

In the config-mode the FTP-server is automatically available.

9.4.3.3 Script update via FTP

The dcs-file generated by the Protocol Developer has to be stored as "script.dcs" by the FTP on the Gateway. When the Gateway starts up, it identifies and transfers the file and then it deletes the file "script.dcs".

9.4.3.4 System configuration update via FTP

A WINGATE gwc-file has to be stored on the Gateway as "SYSTEM.GWC" via FTP. When the Gateway is started, it identifies this file, transfers the new configuration and then deletes the file "SYSTEM.GWC".

9.4.3.5 File system

A file system is available. The gateway is equipped with an additional memory module that can store data of the file system permanently. It is possible to store up to 128 files in the file system. The file names may exist of up to 31 characters. At present the maximum memory capacity is 1 MByte. On request up to 8MByte can be assembled.

9.4.3.6 TCP server channel mode

A TCP connection in which the IC2-FE is the server is closed after 65635 seconds (18.2 hours) if no data has been transferred during this time.

9.4.3.7 UDP channel mode

With the FEX it was possible to initialize a channel with destination IP 0 and port 0 and thus receive UDP packets from all IPs and all ports via this socket. However here it does not work anymore.

9.4.3.8 Raw-channel mode

In the IC2-FE, channel 0 can only be switched to raw mode if the program is not ARP or IP (v4 or v6). The channel number does not matter, but there can only be one such channel.

9.4.3.9 BusStatus

With the IC2-FE, no channels are used for the functions Unigate Scan, FTP and web server. This means that ReadBusStatus is 0x20222222 after BusStart, as long as no additional channels are initialized by the script.

9.5 Setting the Ethernet-address

Setting the address through the configuration

The UNIGATE® IC2 has to be in the configuration mode (see also chapter 7.2 'The configuration mode'). With WINGATE it is now possible to set the address. This address is preserved until it will be changed again.

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
```

10 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

11 Firmware-update

11.1 Overview

In the firmware-update-mode the firmware can be replaced via the UNIGATE[®] IC2's serial interface.

11.2 Adjusting the firmware-update-mode

11.2.1 Adjustment by hardware

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

11.2.2 Adjustment by software

If the UNIGATE[®] IC2 is in the configuration mode (see chapter 7.2) it can be brought to the firmware-update-mode interactively through the command CTRL-F (0x06). After sending the command a security query follows, that has to be answered with J or N (J = Yes, N = No). After a positive confirmation the IC2 is re-started in the firmware-update-mode.

11.3 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 13).

11.4 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.

11.5 Operation mode of the IC2

Standard-operation mode

This mode is required for the regular use of the IC2. In this mode the IC2 will process all Script-commands and normally exchange the corresponding user data. The bus as well is operated in this mode through the IC2.

Configuration mode

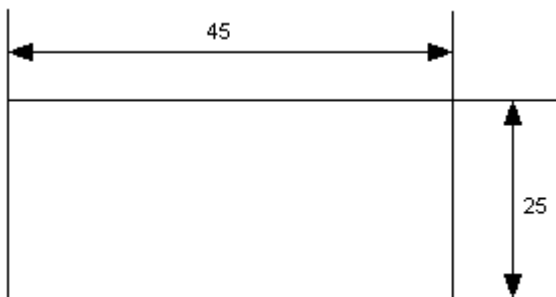
In the configuration mode the UNIGATE[®] IC2 will carry out a self-test after the start (or after a reset). After a successful self-test it will wait for further commands. Here it is possible to load a translated Script into the unit or to initialize the firmware-download-mode.

12 Technical data

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

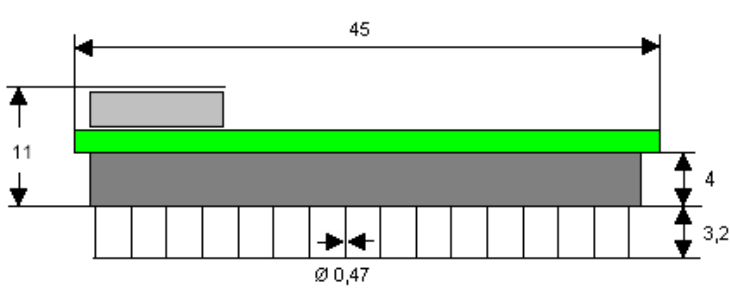
12.1 Mechanics of the UNIGATE® IC2

12.1.1 General dimensions of UNIGATE® IC2

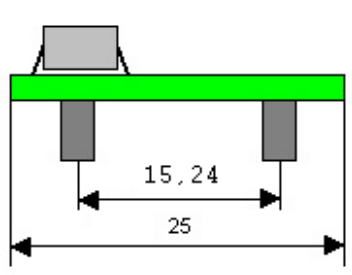


Max. height = 20 mm; including pin

12.1.2 Dimensions UNIGATE® IC2 (Fast Ethernet only)



The pins of UNIGATE® IC2 - Ethernet are arranged with a grid spacing of 2.54 mm.



DIP-Spacing Code 6

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

12.2 Technical data UNIGATE® IC2 - Fast Ethernet

Characteristics	Explanation
Supply voltage	3,3 V \pm 5 %, 200 mA (typ.), 210 mA (max.)
Interface	2 UART interfaces, 1 synchronous serial interface
Physical separation Ethernet-side	Standard
TCP/IP baud rate	Adjustable 10/100 MBaud
UART-baud rate	Up to 7,5 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
Operating temperature	-40°C..+85°C
Storage / transport temperature	-40°C..+100°C
Built-in position	Any

13 Accessory

The following tools are available from Deutschmann Automation.

13.1 Adapter RS232

In an application the adapter RS232 offers the possibility to replace an existing driver MAX 232 (only in DIL-16-housing) by this adapter. This board allows the use of the IC2 according to chapter 3.4 on page 16. Please note that with it the Ethernet does not offer a connection conforming to the standards. With a plug adapter, however, at least the operation of Ethernet is possible.

The hardware is only meant for development purposes. It offers the possibility to make an existing application capable for bus connection in no time and to test the IC2's utilizability and functionality.

13.2 Adapter RS485

From the functionality's point of view the RS485 adapter is the same as the RS232 adapter. It offers the possibility to replace a module LS 176 (only in DIL-8-housing) by the IC2.

There are the same restrictions as for the RS232 adapter.

13.3 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.

13.4 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.

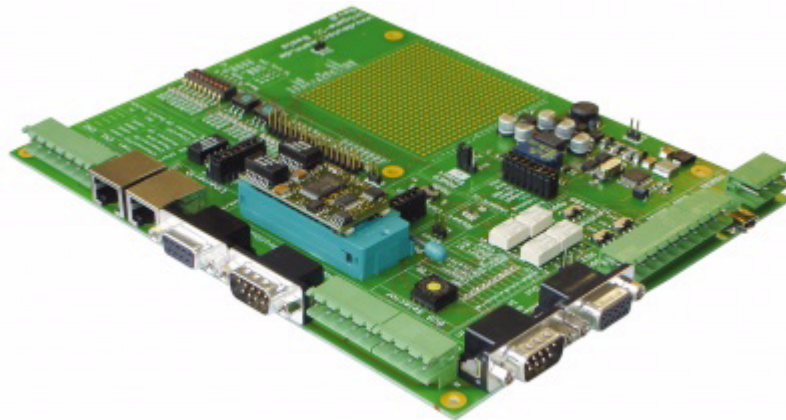
13.5 Developer Kit UNIGATE® IC-AB IC

The Developer Kit IC contains

- a Developer Board UNIGATE® IC2 (see chapter 13.5.1)
- 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.

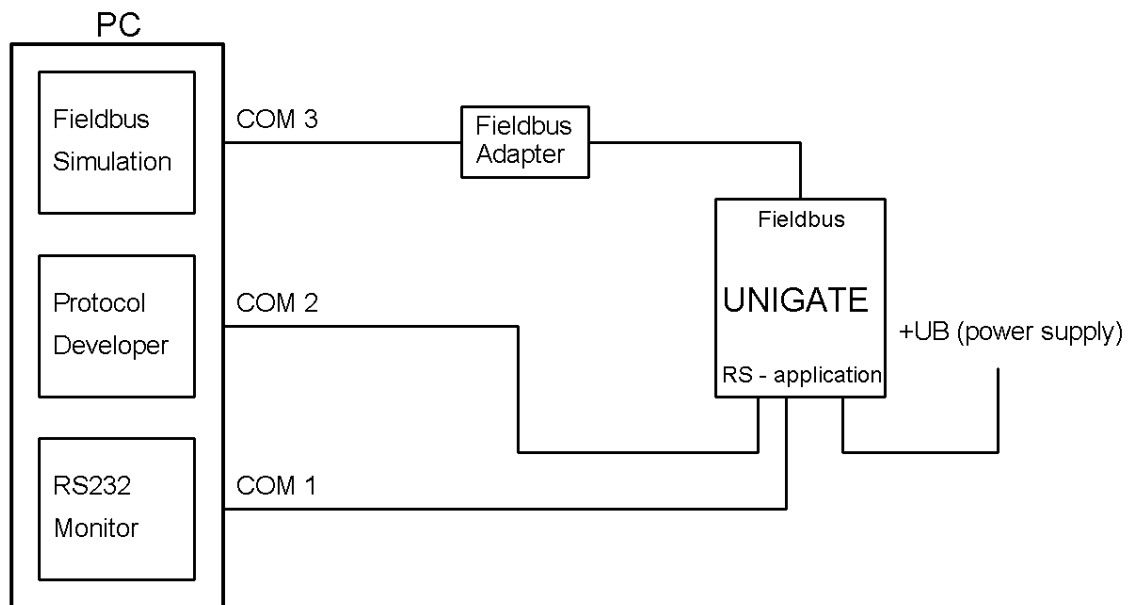
13.5.1 Developer Board UNIGATE® IC-AB

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).

13.5.2 Quick start



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

14 Appendix

14.1 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® 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
ModTCP	=	ModbusTCP
MPI	=	Siemens MPI®
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

15 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 or directly in our Wiki on 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

15.1 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 (± 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.

15.2 Downloading PC software

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

