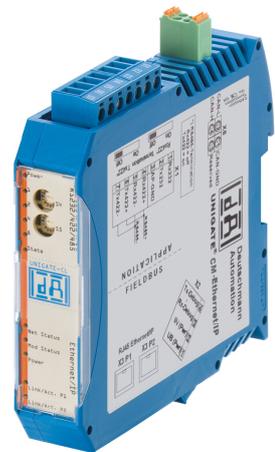




**Deuschmann**

*your ticket to all buses*

**Instruction Manual  
Universal Fieldbus-Gateway  
UNIGATE® CM - EtherNet/IP 2Port**



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



<b>1</b>	<b>Information on CE marking of the module</b>	<b>8</b>
1.1	EU Directive EMC	8
1.2	Scope of application	8
1.3	Note installation guidelines	8
1.4	Installation of the unit	8
1.5	Working on switch cabinets	8
<b>2</b>	<b>Information for the machine manufacturers</b>	<b>9</b>
2.1	Introduction	9
2.2	EU Machinery Directive	9
<b>3</b>	<b>Introduction</b>	<b>10</b>
3.1	UNIGATE® CM software flow-chart	11
3.2	UNIGATE® CM-application diagram	12
<b>4</b>	<b>Operation modes of the Gateway</b>	<b>13</b>
4.1	Configuration mode (config mode)	13
4.2	Test mode	13
4.3	Data exchange mode	14
<b>5</b>	<b>RS-interface</b>	<b>15</b>
5.1	RS-interfaces at the UNIGATE® CM	15
5.2	Buffer sizes at the UNIGATE® CM	15
5.3	Framing Check	15
<b>6</b>	<b>The Debug-interface</b>	<b>16</b>
6.1	Overview of the Debug-interface	16
6.2	Starting in the Debug-mode	16
6.3	Communication parameter for the Debug-interface	16
6.4	Possibilities with the Debug-interface	16
6.5	Commands of the Debug-interface	16
<b>7</b>	<b>Mode of operation of the system</b>	<b>18</b>
7.1	General explanation	18
7.2	Interfaces	18
7.3	Data exchange	18
7.4	Possible data lengths	18
7.5	Startup phase	18
<b>8</b>	<b>Generating a Script</b>	<b>19</b>
8.1	What is a Script?	19
8.2	Memory efficiency of the programs	19
8.3	What can you do with a Script device?	19
8.4	Independence of buses	19
8.5	Further settings at the Gateway	19
8.6	The use of the Protocol Developer	20
<b>9</b>	<b>FTP-Server</b>	<b>21</b>
9.1	Script-update via FTP	21
9.2	System configuration update via FTP	21
<b>10</b>	<b>WEB-Server</b>	<b>22</b>
10.1	Server Side Includes (SSI)	22
10.2	HTML-Forms	23

10.3	Example	23
<b>11</b>	<b>File System</b>	<b>24</b>
<b>12</b>	<b>Hardware ports, switches and LEDs</b>	<b>25</b>
12.1	Device labeling	25
12.2	Connectors	25
12.2.1	Connector to the external device (RS-interface)	25
12.2.2	Connector supply voltage and DEBUG-interface	26
12.2.3	CANopen interface connector	26
12.2.4	EtherNet/IP-connector	26
12.2.5	Power supply	27
12.3	LEDs	27
12.3.1	LED "Power/State"	27
12.3.2	LEDs 1 / 2 / 4 / 8 (Error No. / Select ID)	28
12.3.3	LED "State"	28
12.3.4	LED "Net State"	29
12.3.5	LED "Mod State"	29
12.3.6	LED "Bus Power"	29
12.3.7	LED "Link / Act. P1"	29
12.3.8	LED "Link / Act. P2"	29
12.4	Switches	29
12.4.1	Termination Rx 422 + Tx 422 (serial interface)	29
12.4.2	Rotary coding switches S4 + S5 (serial interface)	30
12.4.3	CANopen termination	30
<b>13</b>	<b>Error handling</b>	<b>31</b>
13.1	Error handling at UNIGATE® CM	31
13.1.1	Error on the Extension	32
<b>14</b>	<b>Installation guidelines</b>	<b>33</b>
14.1	Installation of the module	33
14.1.1	Mounting	33
14.1.2	Removal	33
14.2	Wiring	33
14.2.1	Connection systems	33
14.2.2	EtherNet/IP communication interface	34
14.2.3	Line routing, shield and measures to combat interference voltage	34
14.2.4	General information on line routing	34
<b>15</b>	<b>Firmware Update</b>	<b>36</b>
<b>16</b>	<b>Implemented protocols in UNIGATE® CM</b>	<b>37</b>
16.1	Implemented protocols (Transport protocols at the CM interface)	37
16.2	Configuration options for the individual transport protocol	38
16.2.1	Possibilities on the CANopen side	38
16.2.2	Possibilities on the fieldbus side	39
16.3	Operation of the transport protocols	40
16.3.1	CANopen Mapping	40
16.3.2	Layer 2 11Bit	47
16.3.3	Universal (L2 11Bit)	48
16.3.4	Universal (L2 11Bit) with COB-ID used	48

16.3.5 Universal (L2 11/29Bit)	49
16.3.6 Universal (L2 11/29Bit) with COB-ID used	50
16.3.7 L2 11Bit (Tgl+FBlen)	50
16.4 Delivery status (factory setting)	50
16.5 Trigger byte	50
16.6 Length Byte	50
16.7 Swap word	50
<b>17 Technical data</b>	<b>51</b>
17.1 Device data	51
17.1.1 Interface data	52
<b>18 Commissioning guide</b>	<b>53</b>
18.1 Note	53
18.2 Components	53
18.3 Installation	53
18.4 Dimensional drawing UNIGATE® CM - EtherNet/IP	53
18.5 Commissioning	53
18.6 EtherNet/IP connection	54
18.7 Setting the CANopen address and baudrate	54
18.8 CANopen connection	54
18.9 Connection to the process device	54
18.10 Shield connection	54
18.11 Connecting the supply voltage	54
18.12 Project planning	54
<b>19 Servicing</b>	<b>55</b>
19.1 Returning a device	55
19.2 Downloading PC software	55
<b>20 Annex</b>	<b>56</b>
20.1 Explanations of the abbreviations	56
20.2 Hexadecimal table	57



#### 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 – 2017. 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 Information on CE marking of the module

## 1.1 EU Directive EMC

The following applies to the module described in this User Manual:

Products which bear the CE mark comply with the requirements of EU Directive „Electromagnetic Compatibility“ and the harmonized European Standards (EN) listed therein.

The EU Declarations of Conformity are available at the following location for perusal by the responsible authorities in accordance with the EU Directive, Article 10:

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

## 1.2 Scope of application

The modules are designed for use in the industrial sector and comply with the following requirements.

Scope of application	Requirement applicable to	
	Emitted interference	Interference immunity
Industry	EN 55011, cl. A (2007)	EN 61000-6-2 (2005)

## 1.3 Note installation guidelines

The module complies with the requirements if you

1. comply with the installation guidelines described in the User Manual when installing and operating the module.
2. also follow the rules below on installation of the equipment and on working on switch cabinets.

## 1.4 Installation of the unit

Modules must be installed in electrical equipment rooms/areas or in enclosed housings (e.g. switch boxes made of metal or plastic). Moreover, you must earth the unit and the switch box (metal box) or at least the top-hat rail (plastic box) onto which the module has been snapped.

## 1.5 Working on switch cabinets

In order to protect the modules against static electrical discharge, the personnel must discharge themselves electrostatically before opening switch cabinets or switch boxes.

## **2 Information for the machine manufacturers**

### **2.1 Introduction**

The UNIGATE® module does not constitute a machine as defined by the EU "Machinery" Directive. Consequently, the module does not have a Declaration of Conformity in relation to the EU Machinery Directive.

### **2.2 EU Machinery Directive**

The EU Machinery Directive stipulates the requirements applicable to a machine. The term "machine" is taken to mean a totality of connected parts or fixtures (see also EN 292-1, Paragraph 3.1)

The module is a part of the electrical equipment of the machine and must thus be included by the machine manufacturer in the Declaration of Conformity process.

### 3 Introduction

The UNIGATE® CM-EtherNet/IP 2Port module serves to adapt a CANopen interface to EtherNet/IP networks. The device features 2 EtherNet/IP ports - one port can for instance be used as an outgoing EtherNet/IP port. The terminal unit's protocol is converted in the UNIGATE® via a Script.

The module CM-EtherNet/IP essentially consists of the following hardware components:

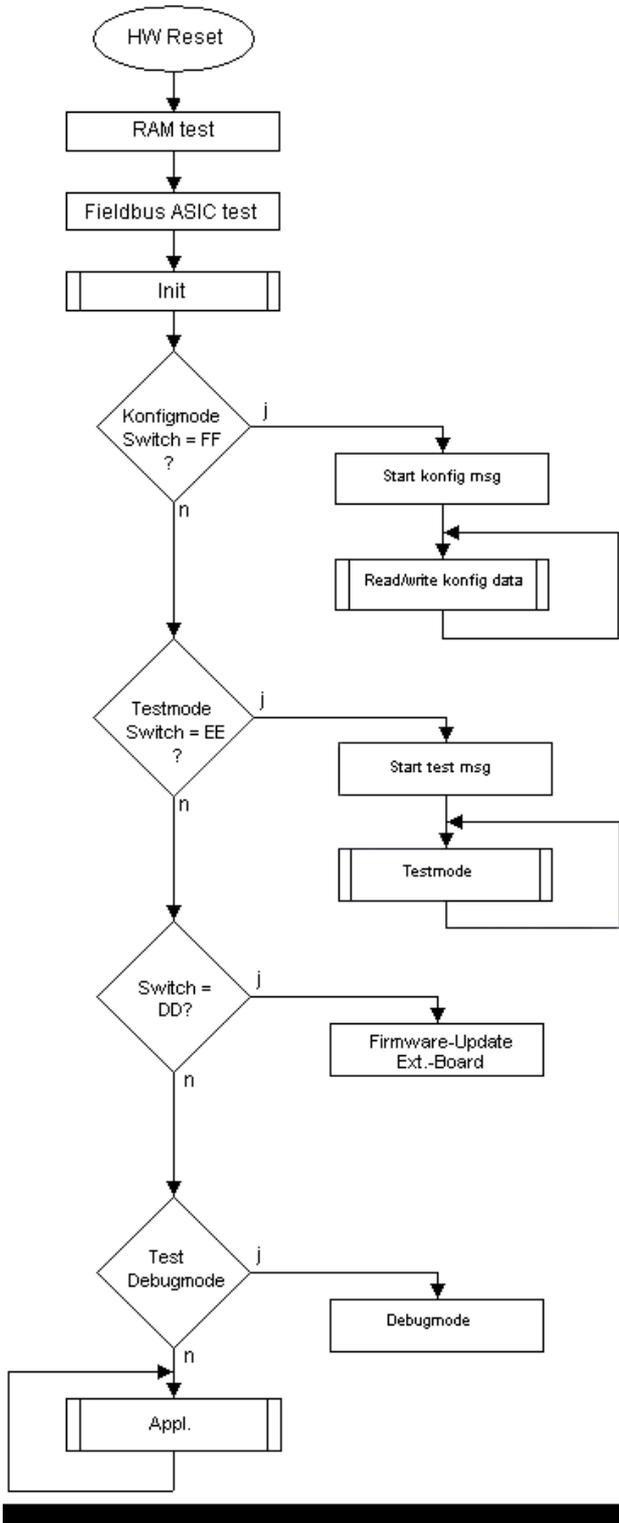
- Electrically isolated EtherNet/IP-interface
- Processor "Fido 1100"
- RAM and FLASH
- Serial interface (RS232, RS485 and RS422)
- Electrically isolated interface to CANopen

#### Conceptual



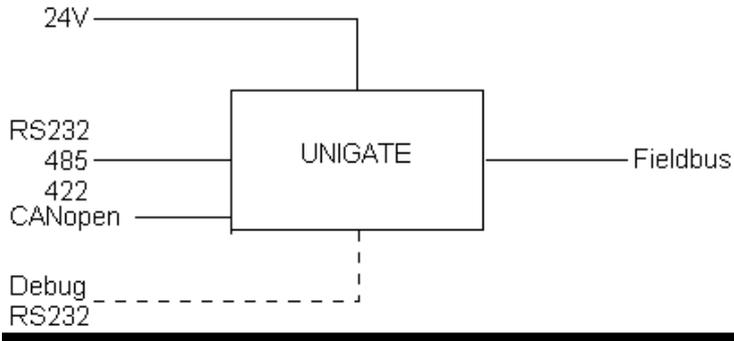
In the further course of this manual, the term "Fieldbus" is frequently used. In these cases this applies to the EtherNet/IP interface.

### 3.1 UNIGATE® CM software flow-chart



### 3.2 UNIGATE® CM-application diagram

The following graph shows a typical connection scheme.



## 4 Operation modes of the Gateway

### 4.1 Configuration mode (config mode)

The configuration mode serves to configure the Gateway. The following adjustments are possible in this mode.

- Loading a Script (e.g. via WINGATE software with "Write Script" under "File")
- Updating the firmware (only CL-basis)
- Configuration of the gateway (via WINGATE Version V2.85 or higher)

The Gateway will be starting in this mode in case both switches S4 as well as S5 are set on position "F" when switching on the Gateway. Right after switching on the Gateway in the configuration mode it will be sending its starting message, that looks analog with the following message:

```
RS-EI-CL (RS+Ext.Board) V4.0.9[41] (c)dA[66,666MHz] Switch=0xFF
Script(C:5719/32512,V:4687/16384)="Universalscript Deutschmann CM" Author="G/S"
Version="V1.0.2" Date=11.01.2016 SN=47110211 IP=0.0.0.0 MAC=00-14-11-15-1A-53
Ext-Board: (CM-IO-DICNET) V1.4 (c)dA 47110002
CAN[MapOff]: ID(11/29)=1 Baud=500000 LSS-ID=1, LSS-BaudIdx=2
```

In the configuration mode the Gateway always operates with the settings 9600 Bauds, no Parity, 8 Databits and 1 Stopbit, the RS-State LED will always be flashing red, the "Error No/Select ID" LEDs are of no account for the user. All software revisions contain the configuration mode.

### 4.2 Test mode

#### Setting of the test mode

The test mode is set by bringing the switches S4 and S5 in position "E". All other switches will not be taken into consideration for the setting of the test mode. Now the Gateway has to be restarted with these settings (by a short disconnection from the power supply).

In the test mode the Gateway always operates with the settings 9600 baud, no parity, 8 databits and 1 stopbit.

The test mode may be helpful to integrate the Gateway in the relevant environment, for instance to test the parameters of the RS-interfaces.

#### Mode of operation of the test mode

After the restart in the test mode the Gateway will be sending the values 0-15 in hexadecimal representation ("0".."F") in ASCII-coding on the serial side every second. Simultaneously the same values are issued binary on the fieldbus-interface.

In this mode the State-LED on the RS-side will be flashing red, the "Error No/Select ID" LEDs will be displaying the value in a binary way, that is issued that moment. Additionally each character that is received at one of the interfaces will also be output at the same interface as a local echo. On the fieldbus-side only the first byte will be used for the local echo, that means on receiving as well as on transmitting only the first byte of the bus data is looked at, the other bus data do not change compared to the last data.

#### 4 Bytes are output on the CANopen interface:

- 1 byte: Echo of the first received byte via CAN
- 2 byte: Read back byte IO8 ! Option
3. byte: value only for internal purposes
4. byte: Output to IO8 ! Option

### 4.3 Data exchange mode

The Gateway has to be in the data exchange mode, so that a data exchange between the CANopen-side of the Gateway and the fieldbus is possible. As long as the Gateway is not in the configuration-, test-, firmware update or debug mode, the data exchange mode is active. In the data exchange mode the Gateway will execute the downloaded Script UniversalScript\_CM (ScriptName „Universalscript Deutschmann CM“).

## 5 RS-interface

### 5.1 RS-interfaces at the UNIGATE® CM

The UNIGATE® CM - EtherNet/IP has the interfaces RS232, RS422 and RS485 available. The hardware always features a DEBUG-interface, see chapter 6.

### 5.2 Buffer sizes at the UNIGATE® CM

UNIGATE® CM features at the serial side a buffer with the size of 1024 bytes 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".

### 5.3 Framing Check

The length of the stop bit received by the Gateway is checked through the function "Framing Check". Here the stop bit generated by the Gateway is always long enough, so that connected participants can evaluate the stop bit.

Please be aware that the function "Framing Check" becomes effective only in case of 8 data bit and the setting "No parity".

An error is detected and indicated by the Error LEDs in case the stop bit does not show the length 1 bit during the activated check.

The possible setting for this parameter can be controlled by the Script (see online help from Protocol Developer). The presetting for the "Stop Bit Framing Check" is "enabled".

## 6 The Debug-interface

### 6.1 Overview of the Debug-interface

The UNIGATE® CM 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® (power up) the firmware will output the binary character 0 (0x00) after a self-test was carried out on this interface. If the UNIGATE® 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.

From Script version [27] on you can also output data with the Script command `"SerialOutputToDebugInterface"`. Please also pay attention to the remark in the manual 'Protocol Developer'.

### 6.5 Commands of the Debug-interface

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

#### Restriction:

- Script commands that make a data exchange with the Ext. Card are not supported in Debug mode.
- The command `„SerialOutputToDebugInterface“`, referred to in chapter 6.4, as well as the command `„SerialInDebugInterface“`, is not supported.
- Redirection of the Debug interface to the Application interface (X1).  
With the script command `„ScriptStartBehaviour (DbgOverApl) ; “` debugging to connector X1 of the application is possible again, so that the commands for the Ext. Cards are supported.

**Note:** The script command `„ScriptStartBehaviour (DbgOverApl) ; “` must be placed directly after the script info.

First, the script must be loaded, because only after a reboot it is possible to debug via the application interface. If the command is taken out, the script has to be started once before you can debug on the Debug interface.

If debugging is redirected to the application, no script or command that wants to carry out communication via the application interface is supported.  
These are e.g.: `SendRS` and `ReceiveSomeCharRS`.

## 7 Mode of operation of the system

### 7.1 General explanation

Communication can be split into seven layers, Layer 1 to Layer 7, in accordance with the ISO/OSI model.

The Deutschmann Automation Gateways convert Layers 1 and 2 of the customized bus system (RS485 / RS232 / RS422) to the corresponding Fieldbus system. Layers 3 and 4 are being covered by the UDP/IP-protocol, TCP/IP-protocol. The Layers 5 and 6 are blank. Layer 7 is converted in accordance with chapter 7.3.

### 7.2 Interfaces

The Gateway features a CANopen and EtherNet/IP interface as well as the serial interfaces RS232-, RS422- and RS485.

### 7.3 Data exchange

All data is transferred by the Gateway in dependence of the downloaded Script.

### 7.4 Possible data lengths

The table below shows the maximum transferable data:

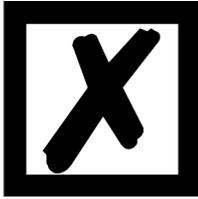
EtherNet/IP	Input data	max. 1060 bytes	variable: here maximum value
	Output data	max. 1060 bytes	variable: here maximum value

CM	With Universal script CM the data length on the CANopen-side is dependant on the protocol		
		CANopen Mapping Layer 2 11Bit Universal (L2 11Bit) Universal (L2 11Bit) with COB-ID used L2 11Bit (Tgl+Fblen)	Universal (L2 11/29Bit) Universal (L2 11/29Bit) with COB-ID used
	In- and Output data	maximum 10 Byte, of which a maximum of 8 Byte is payload	maximum 12 Byte, of which a maximum of 8 Byte is payload

### 7.5 Startup phase

The Master sets up a TCP/IP- or a UDP/IP-connection to the Gateway during the startup phase. Only after a correct termination of the startup phase the data exchange with external devices will take place.

## 8 Generating a Script



**Note: All commands relating to the extension do not work in the debug mode! See chapter 6.5.**

### 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.

### 8.5 Further settings at the Gateway

Most devices require no further adjustments, except for those made in the Script itself.

## 8.6 The use of the Protocol Developer

The software tool Protocol Developer can be downloaded from our website <http://www.deutschmann.de>

It 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. 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 11).

### 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.comment) is replaced by
DisplayFWVersion	Firmware version
DisplayBLVersion	Bootloader version
DisplaySerial	Serial number
DisplayMacID	Ethernet MAC address
DisplayDeviceName DisplayStationName	Device name
DisplayStationType	Station type
DisplayVendorID	Vendor ID
DisplayDeviceID	Device ID
DisplayIP	UNIGATE's IP address
DisplaySubnet	IP subnet mask
DisplayGateway	IP address 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, Variablenname	content of variable

## 10.2 HTML-Forms

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 „SetHTMLInputString“ assigns a name to a script variable, which then can be used in the HTML form.

## 10.3 Example

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
nvdata	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	

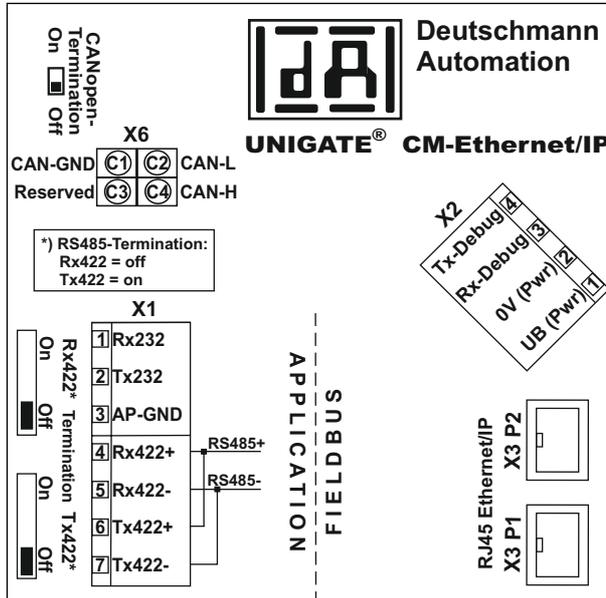
Please note that

- 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 Hardware ports, switches and LEDs

### 12.1 Device labeling



Picture 1: Terminal labeling and termination



Picture 2: Front panel: Rotary switches and LEDs

### 12.2 Connectors

#### 12.2.1 Connector to the external device (RS-interface)

The serial interface is available at the plug accessible on the upper side of the device. Pin assignment X1 (3-pole and 4-pole screw-type plug connector)

Pin No.	Name	Function
1	Rx 232	Receive signal
2	Tx 232	Transmit signal
3	AP-GND	Application Ground
4	Rx 422+ (485+)	Receive signal
5	Rx 422- (485-)	Receive signal
6	Tx 422+ (485+)	Transmit signal
7	Tx 422- (485-)	Transmit signal



For the operation at a 485-interface the two pins labeled "485-" have to be connected together. Also the two pins "485+".

### 12.2.2 Connector supply voltage and DEBUG-interface

Pin assignment X2 (4-pole screw-plug connector, on the bottom side, at the back)

Pin No.	Name	Function
1	UB (Pwr)	10..33 V supply voltage / DC
2	0 V (Pwr)	0 V supply voltage / DC
3	Rx-Debug	Receive signal Debug
4	Tx-Debug	Transmit signal Debug



The 0V (Pwr)-signal can be used as reference (Ground) for the Debug interface.

Pin 3 Rx-Debug: Avoid long cable + open (not connected at the end), otherwise it may malfunction.

### 12.2.3 CANopen interface connector

The accessible plug at the top (rear) is the CANopen interface.

Pin assignment X6 (4 pole connector)

Pin No.	Name	Function
C1	CAN-GND	CAN Ground
C2	CAN-L	Dominant Low
C3	Reserved	Reserved - please do not wire
C4	CAN-H	Dominant High

### 12.2.4 EtherNet/IP-connector

The plugs (labeled: RJ45 EtherNet/IP) for the connection to EtherNet/IP are available on the bottom side of the device.

Pin assignment X3 P1 (RJ45 up front, i. e. which is located closer to the front panel)

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

Pin assignment X3 P2 (RJ45 at the back, i. e. which is located closer to the DIN-rail)

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

### 12.2.5 Power supply

The device must be powered with 10-33 VDC, The voltage supply is made through the 4-pole screw-plug connector at the device's bottom side.

Please note that the devices of the series UNIGATE® should not be operated with AC voltage.

### 12.3 LEDs

The Gateway UNIGATE® CM - EtherNet/IP features 10 LEDs with the following significance:

LED Power/State	red/green	Interface state CANopen
LEDs 1 / 2 / 4 / 8 (Error No. / Select ID)	green	User-defined / general Gateway error
LED State	red/green	User-defined / general Gateway error
LED Net State	red/green	Interface state EtherNet/IP / network status
LED Mod State	red/green	Module-state
LED Bus Power	green	Supply voltage EtherNet/IP (total device)
LED Link / Act. P1	green	Ethernet-link pulse found / network data traffic Port 1
LED Link / Act. P2	green	Ethernet-link pulse found / network data traffic Port 2

#### 12.3.1 LED "Power/State"

The Power/State LED signals the status and operating condition of the CANopen interface and can have the following conditions in the data exchange mode:

#### Indicator states and flash rates

The following Indicator states are distinguished:

LED on	constantly on
LED off	constantly off
LED	flickering iso-phase on and off with a frequency of approximately 10 Hz: on for approximately 50 ms and off for approximately 50 ms.
LED blinking	iso-phase on and off with a frequency of approximately 2,5 Hz: on for approximately 200 ms followed by off for approximately 200 ms.
LED single flash	one short flash (approximately 200ms) followed by a long off phase (approximately 1000 ms).
LED double flash	a sequence of two short flashes (approximately 200ms), separated by an off phase (approximately 200ms). The sequence is finished by a long off phase
LED triple flash	(approximately 1000 ms). a sequence of three short flashes (approximately 200ms), separated by an off phase (approximately 200ms). The sequence is finished by a long off phase (approximately 1000 ms).

If one bicolor Status LED is used instead of two single color LEDs, this LED shall indicate both the physical bus status and the status of the CANopen state machine. This bicolor LED shall be red and green.

**CANopen ERROR LED (red)**

The CANopen Error LED indicates the status of the CAN physical layer and indicates errors due to missing CAN messages (SYNC, GUARD or HEARTBEAT).

ERROR LED	State	Description
Off	No error	The Device is in working condition.
Single flash	Warning limit reached	At least one of the error counters of the CAN controller has reached or exceeded the warning level (too many error frames).
Flickering	AutoBaud/LSS	Auto Baudrate detection in progress or LSS services in progress (alternately flickering with RUN LED).
Double flash	Error Control Event	A guard event (NMT-Slave or NMT-master) or a heartbeat event (Heartbeat consumer) has occurred.
Triple flash	Sync Error	The SYNC message has not been received within the configured communication cycle period time out (see Object Dictionary Entry 0x1006).
On	Bus Off	The CAN controller is bus off

If at a given time several errors are present, the error with the highest number is indicated (e.g. if NMT Error and Sync Error occur, the SYNC error is indicated).

**CANopen RUN LED (green)**

The CANopen RUN LED indicates the status of the CANopen network state machine.

CAN RUN LED	State	Description
Flickering	AutoBaud/LSS	Auto Baudrate detection in progress or LSS services in progress (alternately flickering with RUN LED).
Single flash	STOPPED	The device is in STOPPED state.
Blinking	PRE-OPERATIONAL	The device is in the PRE-OPERATIONAL state.
On	OPERATIONAL	The device is in the OPERATIONAL state.

Whilst the device is executing a reset the CANopen RUN LED shall be off.

In case there is a conflict between turning the LED on green versus red, the LED may be turned on red. Apart from this situation, the bicolor status LED shall combine the behavior of the CAN Error LED.

**More conditions in Configuration-, Test- or Update-Mode**

green/red flashing	UNIGATE® is in test mode
red flashing	UNIGATE® is in configuration mode / error (see error table chapter 13.1.1)
red bright	CL basis stopped, PC connection with Ext.-Board active (Firmware update, see chapter 15)

**12.3.2 LEDs 1 / 2 / 4 / 8 (Error No. / Select ID)**

If these 4 LEDs flash and LED "State" simultaneously lights red, the error number is displayed in binary notation (conversion table, see Annex) in accordance with the table in chapter "Error handling". Additionally these LEDs are controllable via Script.

**12.3.3 LED "State"**

Lights green	Controllable via Script
Flashes green	Controllable via Script
Flashes green/red	Controllable via Script
Lights red	General Gateway error (see LEDs Error No.), controllable via Script
Flashes red	UNIGATE is in the configuration / test mode, controllable via Script

### 12.3.4 LED "Net State"

Flashes green/red	Self-test
Lights green	Connection available, data exchange active
Flashes green	Waiting for connection establishment
Flashes green/red	Timeout during connection
Lights red	IP-address assigned twice

### 12.3.5 LED "Mod State"

Flashes green/red	Self-test
Lights green	Data exchange active
Flashes green	Standby / module is waiting for connection establishment
Flashes green/red	Module-error
Lights red	Substantial module-error

### 12.3.6 LED "Bus Power"

This LED is connected directly to the supply voltage of the EtherNet/IP-side.

### 12.3.7 LED "Link / Act. P1"

This LED is directly controlled by the Ethernet-Controller and shines when the gateway at Port 1 (X3 P1) is in an operable network (link-pulses are received) and it flickers when network data traffic takes place.

### 12.3.8 LED "Link / Act. P2"

This LED is directly controlled by the Ethernet-Controller and shines when the gateway at Port 2 (X3 P2) is in an operable network (link-pulses are received) and it flickers when network data traffic takes place.

## 12.4 Switches

The Gateway features 5 switches with the following functions:

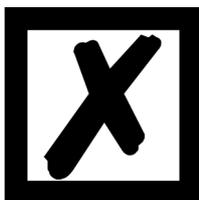
Termination Rx 422	switchable Rx 422-terminating resistor for the serial interface
Termination Tx 422	switchable Tx 422- or RS485-terminating resistor for the serial interface
Rotary coding switch S4	ID High for serial interface i. e. configmode
Rotary coding switch S5	ID Low for serial interface i. e. configmode
CANopen-Termination	switchable CANopen terminating resistor

### 12.4.1 Termination Rx 422 + Tx 422 (serial interface)

If the Gateway is operated as the physically first or last device in an RS485-bus or as 422, there must be a bus termination at this Gateway. In order to do this the termination switch is set to position ON. The resistor (150  $\Omega$ ) integrated in the Gateway is activated. In all other cases, the switch remains in position OFF.

Please refer to the general RS485 literature for further information on the subject of bus terminations.

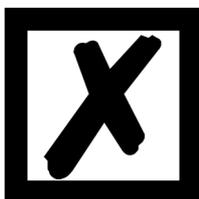
If the integrated resistor is used, please allow for the fact that this also activates a pull-down resistor (390  $\Omega$ ) to ground and a pull-up resistor (390  $\Omega$ ) to VCC.



At RS48 only the Tx 422-switch must be set to ON.  
The Rx 422-switch has to be on OFF.

### 12.4.2 Rotary coding switches S4 + S5 (serial interface)

These two switches can be read out through the Script command "Get (RS\_Switch, Destination)" and the value can be used for further functions. This value is read in when the Gateway is switched on or always after a Script command has been executed. The switch positions „DD“ (Firmware Update Ext.-Board), "EE" (testmode) and "FF" (config mode) are assigned, as seen in the table below, and can not be read out through the above script command.



The switch position "DD" (ie, S4 and S5 in position "D") is reserved for internal purposes, Firmware Update Ext.-Board. The Gateway should only be switched into this mode for a Firmware-Update. For details see chapter 15.

#### Switch positions

Switch postions S4	Switch postions S5	Function	Description
D	D	Firmware-Update Ext.-Board	(Description see chapter 15)
E	E	Test mode	(Description see chapter 4.2) <b>Note:</b> This mode can only be terminated by a reboot.
F	F	Config mode	(Description see chapter 4.1) <b>Note:</b> This mode can only be terminated by a reboot.

### 12.4.3 CANopen termination

If the Gateway is operated as the first or last physical device in the CANopen®, there must be a bus termination at this Gateway. In order to do this, either a bus terminating resistor must be activated in the connector or the resistor (120 Ω) integrated in the Gateway must be activated. In order to do this, slide the slide switch to position ON. In all other cases, the slide switch must remain in position OFF. Please refer to the general Fieldbus literature for further information on the subject of bus termination.

## 13 Error handling

### 13.1 Error handling at UNIGATE® CM

If the Gateway detects an error, the error is signalled by the "State" LED lighting red and, simultaneously, the error number being indicated by means of LEDs "Error No." as shown in the table below. A distinction can be made between two error categories:

Serious errors (1-5): 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.

For user-defined errors the flash frequency is 0.5 hertz. The error is displayed as long as defined by "Set Warning Time".

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

LED8	LED4	LED2	LED1	Error no. resp. ID	Error description System error (2x/sec)	Error description with Uni- versal Script-CM (Delivery status) User error (1x/sec)
0	0	0	0	0	Reserved	
0	0	0	1	1	Hardware fault	
0	0	1	0	2	EEROM error	
0	0	1	1	3	Internal memory error	CL hardware not supported
0	1	0	0	4	Fieldbus hardware error or wrong Fieldbus-ID	
0	1	0	1	5	Script error	Hardware for "Universal CM" script not supported
0	1	1	0	6	Reserved	unknown protocol
0	1	1	1	7	RS-transmit buffer overflow	
1	0	0	0	8	RS-receive buffer overflow	
1	0	0	1	9	RS timeout	
1	0	1	0	10	General fieldbus error	
1	0	1	1	11	Parity-or frame-check-error	
1	1	0	0	12	Reserved	
1	1	0	1	13	Fieldbus configuration error	
1	1	1	0	14	Fieldbus data buffer overflow	
1	1	1	1	15	Reserved	

Table 1: Error handling at UNIGATE® CM

A detailed error diagnosis can be recorded with an activated "Diagnosis Monitor" via the application interface. For this you only need to connect/start a RS232 Monitor tool, such as Deutschmann Starterkit tool "RS232 module" via a PC. (9600/1/8/N).

(AplDiag=ON)	diagnosis output at application interface = on
(START)	CM-UNIGATE is booting
(CM-online)	CM-UNIGATE is powered up (booting finished)
(CL-State:00000002)	fieldbus state has changed = not yet in data exchange
(CL-State:00000001)	fieldbus state has changed = not yet in data exchange
(CL-State:000000E0)	fieldbus state has changed = data exchange
(CL-Bus Activ)	First time in data exchange after switching on
(Layer 2)	configured transport protocol is processed cyclically

An error can be as follows: „(E:21)0x00(ExtErr:030)0x00“.

**Note:** "E" shows a script error, the value afterwards in Hex stands for the error number. The error number can be evaluated in the Protocol Developer Help under Appendix -> Return codes.

"ExtErr:" is for a more detailed fault localization. The script source code can be searched with the following number (ExtErr:**030**).

### 13.1.1 Error on the Extension

An error on the extension will be signaled through red flashing of the Power/State-LED. The LED of the according error number turns off. This is followed by a short, then the flashing sequence is repeated.

Example: With an SRAM-Error 3 the LED lights red, turns off 3 times, lights up again for a moment and everything starts over. The following errors are possible on the extension.

Error Number	Error description
1	HARDWARE_ERROR
2	STACK_ERROR FLASH_INIT_ERR
3	SRAM_ERROR FLASH_ERROR
4	CL_KOMM_ERROR, e.g. CL Firmware doesn't support an extension yet
5	BUS_ID_ERROR FLASH_CHECK_ERR
6	CL_KOMM_RX_ERR
7	CL_KOMM_TX_ERR
8	NSW_DATA_ERROR
9	TIMOUT_ERROR
10	TX_ERROR
11	RX_ERROR
12	ID_ERROR, e.g. double Dicnet-ID, or ID not in net 24V_ERROR, overload (only at Option I/O8)
13	PARA_ERROR
14	BUS_ERROR
15	NSW_PROG_ERROR

## 14 Installation guidelines

### 14.1 Installation of the module

The module with the dimensions 23 x 115 x 117 mm (W x D x H) has been developed for switch cabinet use (IP 20) and can thus be mounted only on a standard mounting channel (deep DIN-rail to EN 50022).

#### 14.1.1 Mounting

- Engage the module from the top in the top-hat rail and swivel it down so that the module engages in position.
- Other modules may be rowed up to the left and right of the module.
- There must be at least 5 cm clearance for heat dissipation above and below the module.
- The standard mounting channel must be connected to the equipotential bonding strip of the switch cabinet. The connection wire must feature a cross-section of at least 10 mm<sup>2</sup>.

#### 14.1.2 Removal

- First disconnect the power supply and signal lines.
- Then push the module up and swivel it out of the top-hat rail.

Vertical installation

The standard mounting channel may also be mounted vertically so that the module is mounted turned through 90°.

### 14.2 Wiring

#### 14.2.1 Connection systems

The following connection systems must resp. may be used when wiring the module:

- Standard screw-type/plug connection (power supply + RS)
- 8-pin RJ45-plug-in connection (EtherNet/IP connection)

a) In the case of standard screw-type terminals, one lead can be clamped per connection point. It is best to then use a screwdriver with a blade width of 3.5 mm to firmly tighten the screw.

Permitted cross-sections of the line:

- Flexible line with wire-end ferrule: 1 x 0.25 ... 1.5 mm<sup>2</sup>
- Solid conductor: 1 x 0.25 ... 1.5 mm<sup>2</sup>
- Tightening torque: 0.5 ... 0.8 Nm

b) The plug-in connection terminal strip is a combination of standard screw-type terminal and plug connector. The plug connection section is coded and can thus not be plugged on the wrong way round.

##### 14.2.1.1 Power supply

The device must be powered with 10...33 V DC.

- Connect the supply voltage to the 4-pole plug-in screw terminal in accordance with the labelling on the device.

##### 14.2.1.2 Equipotential bonding connection

The connection to the potential equalization automatically takes place if it is put on the DIN-rail.

### 14.2.2 EtherNet/IP communication interface

This interface is located on the module in the form of two 8-pin RJ45 sockets on the bottom side of the housing.

- Plug the EtherNet/IP-connector(s) into the RJ45 socket labelled "RJ45 EtherNet/IP" until it snaps in.
- Please make sure that the length of the line to the adjacent Ethernet participants does not fall below 0.6 m.

### 14.2.3 Line routing, shield and measures to combat interference voltage

This chapter deals with line routing in the case of bus, signal and power supply lines, with the aim of ensuring an EMC-compliant design of your system.

### 14.2.4 General information on line routing

- Inside and outside of cabinets

In order to achieve EMC-compliant routing of the lines, it is advisable to split the lines into the following line groups and to lay these groups separately.

- ⇒ Group A:
  - shielded bus and data lines (e. g. for RS232C and printers etc.)
  - shielded analogue lines
  - unshielded lines for DC voltages  $\geq 60$  V
  - unshielded lines for AC voltage  $\geq 25$  V
  - coaxial lines for monitors
- ⇒ Group B:
  - unshielded lines for DC voltages  $\geq 60$  V and  $\geq 400$  V
  - unshielded lines for AC voltage  $\geq 24$  V and  $\geq 400$  V
- ⇒ Group C:
  - unshielded lines for DC voltages  $> 400$  V

The table below allows you to read off the conditions for laying the line groups on the basis of the combination of the individual groups.

	Group A	Group B	Group C
Group A	1	2	3
Group B	2	1	3
Group C	3	3	1

Table 3: Line laying instructions as a function of the combination of line groups

- 1) Lines may be laid in common bunches or cable ducts.
- 2) Lines must be laid in separate bunches or cable ducts (without minimum clearance).
- 3) Lines must be laid in separate bunches or cable ducts inside cabinets but on separate cable racks with at least 10 cm clearance outside of cabinets but inside buildings.

#### 14.2.4.1 Shielding of lines

Shielding is intended to weaken (attenuate) magnetic, electrical or electromagnetic interference fields.

Interference currents on cable shields are discharged to earth via the shielding bus which is connected conductively to the chassis or housing. A low-impedance connection to the PE wire is particularly important in order to prevent these interference currents themselves becoming an interference source.

Wherever possible, use only lines with braided shield. The coverage density of the shield should exceed 80%. Avoid lines with foil shield since the foil can be damaged very easily as the result of tensile and compressive stress on attachment. The consequence is a reduction in the shielding effect.

In general, you should always connect the shields of cables at both ends. The only way of achieving good interference suppression in the higher frequency band is by connecting the shields at both ends.

The shield may also be connected at one end only in exceptional cases. However, this then achieves only an attenuation of the lower frequencies. Connecting the shield at one end may be more favorable if

- it is not possible to lay an equipotential bonding line
- analogue signals (a few mV resp. mA) are to be transmitted
- foil shields (static shields) are used.

In the case of data lines for serial couplings, always use metallic or metallized plugs and connectors. Attach the shield of the data line to the plug or connector housing.

If there are potential differences between the earthing points, a compensating current may flow via the shield connected at both ends. In this case, you should lay an additional equipotential bonding line.

Please note the following points when shielding:

- Use metal cable clips to secure the shield braiding. The clips must surround the shield over a large area and must have good contact.
- Downstream of the entry point of the line into the cabinet, connect the shield to a shielding bus. Continue the shield as far as the module, but do not connect it again at this point!

## 15 Firmware Update

The UNIGATE® CM consists of a basic hardware and an ext. board. The firmware versions are output in the configuration mode (see Chapter 4.1). The start-up message of the basic appears, following the message of the ext. board that looks similar to the following:

```
Ext-Board: (CM-IO-DICNET) V1.2 (c)dA 47110001 CAN: ID=1 Baud=125000 LSS-ID=1,
LSS-BaudIdx=4
```

The switch position "DD" serves for the Firmware-Update of the ext. board. The gateway will start in this mode when the switches S4 and S5 are in position "D" during start up of the gateway. Then the following message appears.:

"If you want to download a new firmware for the expansion board press the FDT 'Firmware Download' button within 10 seconds."

Within 10 seconds the firmware update can be started via the Firmware-Download-Tool FDT (Firmware download).

There will be a timeout if the firmware download is not started within 10 seconds.

\*\*\*Timeout: Restart UNIGATE!\*\*\*

The firmware update of the basic hardware takes place in configuration mode. (see chapter 4.1)

## 16 Implemented protocols in UNIGATE® CM

On delivery the UNIGATE® CM is supplied with the Script "Universalscript Deutschmann CM". The configuration is carried out in the configuration mode (see chapter 4.1) with the software WINGATE from Version V2.85 and wcf-file from 406.



**Attention:**

If a Reset Device is carried out it is possible (depending on the firmware version of the UNIGATE®) that the "Universal Script" will get lost and must be played in again.

The Script can be found on the Deutschmann Support-DVD in the folder `\Software\ProtocolDeveloper\Example\Universal\UniversalScript CM\`.

### 16.1 Implemented protocols (Transport protocols at the CM interface)

- CANopen Mapping
- Layer 2 11Bit
- Universal (L2 11Bit)
- Universal (L2 11Bit) with COB-ID used
- Universal (L2 11/29Bit)
- Universal (L2 11/29Bit) with COB-ID used
- L2 11Bit (Tgl+FBlen)



**Note:** A data exchange is only possible when the EtherNet/IP is in „Data-Exchange“.



**Note:** For all protocols the CAN Baudrate has to be set via WINGATE, Parameter "(CM) CAN Baudrate".

## 16.2 Configuration options for the individual transport protocol

### 16.2.1 Possibilities on the CANopen side

(CM) Transport protocol (CM) Parameter	CANopen Mapping	Layer 2 11Bit	Universal (L2 11Bit)	Universal (L2 11Bit) COB-ID	Universal (L2 11/29Bit)	Universal (L2 11/29Bit) COB-ID	L2 11Bit (Tgl+FBlen)
(CM) Profile	✓						
(CM) Slave Mode	✓						
(CM) CAN Baudrate	✓	✓	✓	✓	✓	✓	✓
(CM) CAN Field-bus ID	✓						
(CM) COB-ID used(1/16)				✓		✓	
,	,	,	,	,	,	,	,
,	,	,	,	,	,	,	,
,	,	,	,	,	,	,	,
,	,	,	,	,	,	,	,
,	,	,	,	,	,	,	,
(CM) COB-ID used(16/16)				✓		✓	

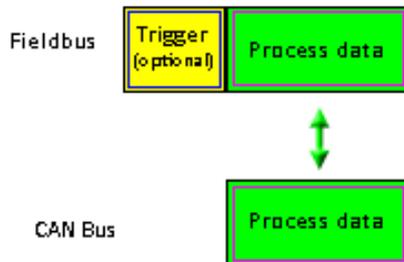
**16.2.2 Possibilities on the fieldbus side**

(CM) Transport protocol Fieldbus Parameter	CANopen Mapping	Layer 2 11Bit	Universal (L2 11Bit)	Universal (L2 11Bit) COB-ID	Universal (L2 11/29Bit)	Universal (L2 11/29Bit) COB-ID	L2 11Bit (Tgl+ FBlen)
Blocklength (0-1060) fieldbus input	✓	✓	✓	✓	✓	✓	✓
Blocklength (0-1060) fieldbus output	✓	✓	✓	✓	✓	✓	✓
Fieldbus Data exchange	✓		✓	✓	✓	✓	
Fieldbus length- byte			✓	✓	✓	✓	
IP address UNIGATE	✓	✓	✓	✓	✓	✓	✓
Subnet mask	✓	✓	✓	✓	✓	✓	✓
IP address Gate- way	✓	✓	✓	✓	✓	✓	✓
DHCP	✓	✓	✓	✓	✓	✓	✓
QoS	✓	✓	✓	✓	✓	✓	✓
Swap word	✓		✓	✓	✓	✓	

## 16.3 Operation of the transport protocols

### 16.3.1 CANopen Mapping

#### Data structure



The Gateway can be configured through the software WINGATE® that is also supplied. This Gateway module from the UNIGATE range of Deutschmann Automation allows a coupling between **fieldbus** and **CANopen**®.

Here the Gateway on the CANopen® side acts as a standard-compliant master and on the EtherNet/IP side as a standard-compliant slave so that the problems that appear again and again (to couple the CANopen® Slave participant to a EtherNet/IP network) are solved.

#### Data transfer

The data exchange between the Profibus and the CANopen® will be fixed with the Mapping which is configured via "WINGATE®".

Mapping means the illustration of data of one fieldbus to any other fieldbus. As data all data can be used which are made available from the corresponding busses.

The Mapping works in both directions from the Master to the Slave as well as reverse.

Example:

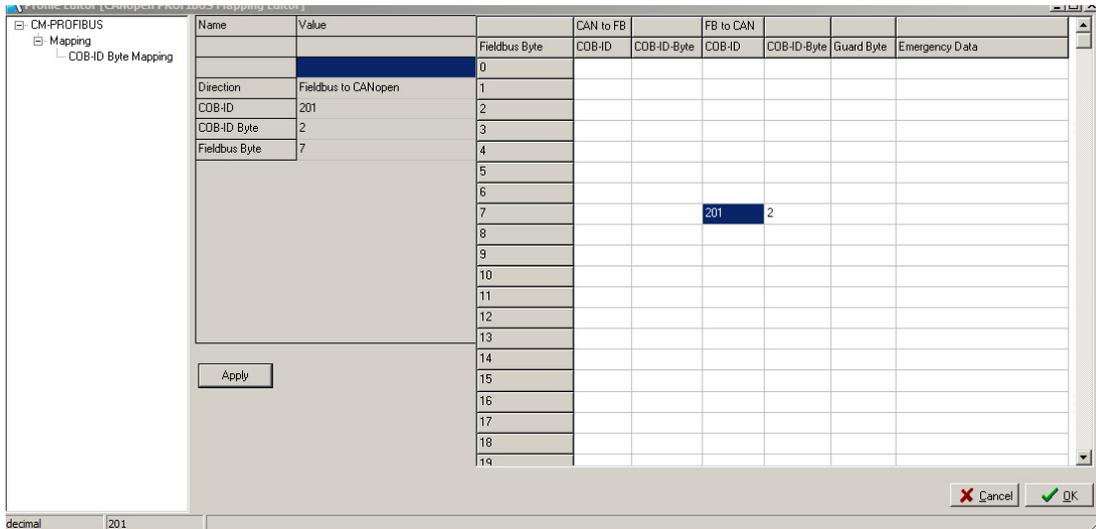
That way it can be defined that the 2nd byte of the COB-ID201 (CANopen®-participant) is transferred into the Profibus data as 7th byte.

At present 256 mappings are possible (128 Tx and 128 Rx Mappings). The data is transferred as PDOs only. You can setup maximum of 16 Tx and 16 Rx PDOs. Each PDO can have up to 8 bytes (means 8 mappings).  $16 \text{ PDO} * 8 \text{ byte} = 128 \text{ mappings}$ . SDO accesses are also supported (see chapter 16.3.1.10)

#### 16.3.1.1 Configuration of the mapping

In order to create or work on such a mapping, please connect the UNIGATE with the PC via the RS23-interface and there start the software "WINGATE®".

After the start of the software an upload from the connected device will be carried out automatically. The corresponding matching parameters to this device are displayed. Select the CANopen Mapping protocol and look for the entry "(CM) Profile" in the table and click doubled onto the value "map ...". After a short period a dialogue is displayed. With it, it is possible to process all map entries (similar to the following picture):



**Example: UNIGATE CM PROFIBUS**

Now the Mappings can be changed, deleted or new ones can be added. Alternatively, the values can also be entered in the right table. The entry must be confirmed with "Apply", only then the values are taken.

Afterwards you only have to download the data to the Gateway (menu File\Download) and UNIGATE® works with the new configuration.



**It is allowed to map the same byte of a PDO several times to different positions in the fieldbus. However, a maximum of 8 mappings per PDO may be entered!**



**At the mapping from fieldbus to CANopen® several mappings can be configured for each Profibus-byte. There the total number of these multiple mappings must not exceed 31.**



**If possible for the PDO-mappings only COB-IDs from the range 181H..580H should be used (see DS301-specification for CANopen®).**

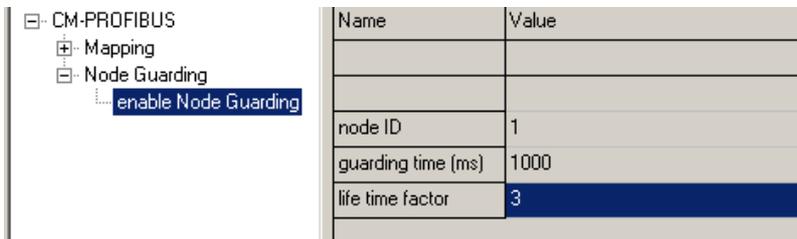
### 16.3.1.2 Node guarding

If a CANopen® slave controlled by node guarding, it cyclically receives the node guard request to answer the inquiry through the CANopen® master. The slave for his part answers this node guarding inquiry with the corresponding answer. That way, both master and slave can control themselves mutually.

If the slave receives no node guard inquiry it assumes that the bus connection is broken or the master went down.

#### Settings

As presetting it applies tfor all slaves that no node guarding is carried out. Shall that node guarding be activated for a slave, an entry "enable node guarding" has to be added to the current WINGATE configuration. The parameters "node ID", "guarding time" and "life time factor" are to be determined in every entry.



Settings to node guarding (Example UNIGATE CM PROFIBUS)

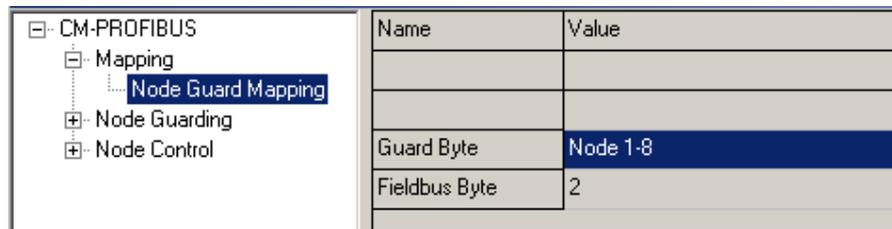
The parameter node ID indicates the ID of the CANopen® node on which the control is supposed to happen. With the guarding time the interval in which the the master calls on the slave to send its guarding answer is to be indicated in milliseconds.

The life time factor states how often the answer of the slave may be missing before the master assumes a failure of the slave.

These settings are available from Gateway software-revision 1.1 on.

### 16.3.1.3 Displaying the node guarding states

The node guard mapping is used to represent the node guard states of the single slaves to the fieldbus. Always 8 slaves are represented in one byte; so exactly one bit of information is reserved for one. This information can be shown in any desired spot in the fieldbus data of the Gateway. So they are shown in the input area of the fieldbus master for this Gateway.



Display of the node guarding information (Example UNIGATECM PROFIBUS)

#### Arrangement of the bits for the nodes 1- 8:

Node 1-8

Byte MSB ( bit 7)	...	LSB (bit 0)
SlavenNode ID 8	...	Slave node ID 1

If no node guard is activated for a slave, the slave went down or still no node guarding information was exchanged, the corresponding bit for this slave is 0. The bit is 1, if node guarding is activated for this slave and the slave answered in the corresponding time period to the node guarding inquiries.

Example: For node 1 and 2 node guarding is activated. Slave 1 doesn not answer to the node guarding and slave ID 2 works regularly. A mapping of information is entered into the fieldbus in the 2nd byte.

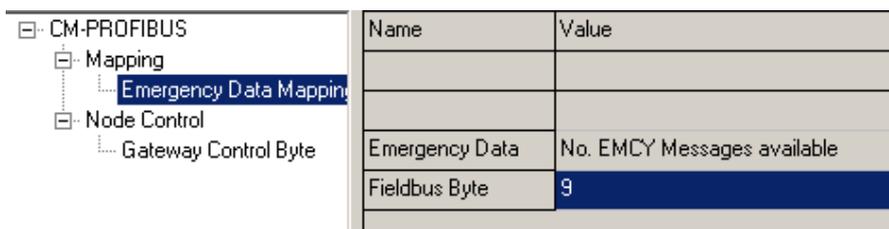
**Fieldbus data of the Gateway:**

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
xx	xx	0x40	xx	xx

These settings are available from Gateway software-revision 1.1 on.

**16.3.1.4 Emergency Messages**

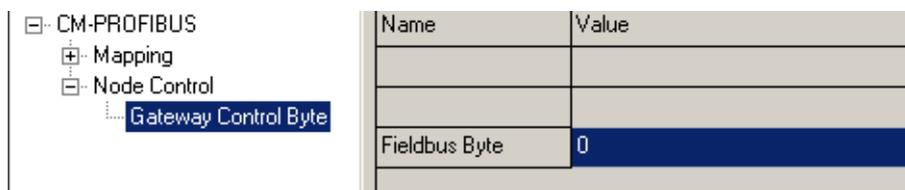
The gateway can received up to 10 Emergency messages of all those slaves, for which node guarding was activated and writes them into a queue. Simultaneously an internal counter is raised which contains the number of received EMCY messages. The data of the current messages as well as the number of the received EMCY messages and the node ID of the sender of the message can be shown in the fieldbus output data of the gateway. With the control byte the next message can be recalled. If the data of the EMCY message is shown in the bus data, the new data in the fieldbus is already visible with the toggle of the data



Setting of the emergency messages (Example UNIGATE CM PROFIBUS)

**16.3.1.5 Gateway control byte**

Some characteristics of the Gateway can be controlled by fieldbus data. For this the information of the fieldbus has to be transmitted to the gateway, that means that the information has to be in the input area of the gateway. Only if this information is shown from the fieldbus in the Gateway, the data can be analysed by the Gateway.



Setting of the Gateway control byte (Example UNIGATE CM PROFIBUS)

Gateway control byte							
Bit 7 EMCY toggle bit	Bit 6 set OP toggle bit	Bit 5 Request to response	res.	res.	res.	res.	res.

**16.3.1.6 Bit 7 (MSB): EMCY toggle bit**

With this bit the next relevant EMCY message can be transferred to the current data buffer, no matter whether the data is shown in the fieldbus or not. The counter is lowered by 1, provided that it does not equal 0.

**16.3.1.7 Bit 6: set operational toggle bit**

If this bit is changed the Gateway sends the message 'set operational' to the slaves. This feature is to be used if some slaves are slower than the master, and are only then ready when the master already sent this message.

This setting is available from Gateway software-revision 1.1 on.

**16.3.1.8 Bit 5: RTR - Request to response**

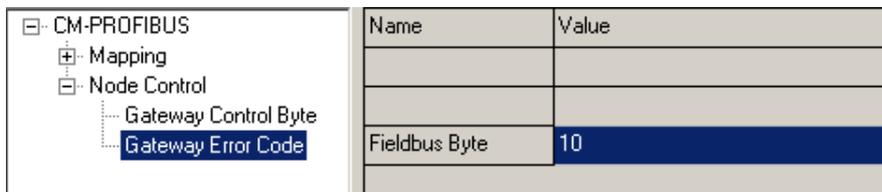
With this bit (transition from 0 -> 1) RTR-telegrams can be sent to any PDO mapped as Rx. This RTR results in the fact that the addressed PDOs are sent and with it they are updated in the Gateway.

If the RTR-function is used, at least 1 Rx-PDO has to be mapped. Furthermore the amount of mapped Rx-PDOs must not exceed 50, since otherwise it will result in an overrun.

This setting is available from Gateway software-revision 2.3 on.

**16.3.1.9 Gateway error byte**

The Gateway's current error state itself can be shown in the fieldbus as well as the other values of the node guarding or the slave data.



**Setting of the error byte (Example UNIGATE CM PROFIBUS)**

This setting is available from Gateway software-revision 1.1 on.

**16.3.1.10 SDO Obj Mapping**

Available from Gateway software-revision 2.2 on!

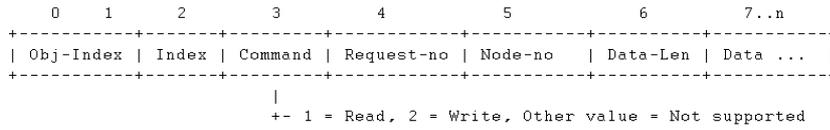
SDO-transfer through a "window" in the fieldbus (FB). Here the following fixed data record structure is used in the FB, that is mapped to the FB-data from the FB-index on. With it a difference is made between:

- FB-Idx (In) = Data from the FB-Master (PLC) to UNIGATE
- FB-Idx (Out) = Data to the FB-Master (PLC) from UNIGATE

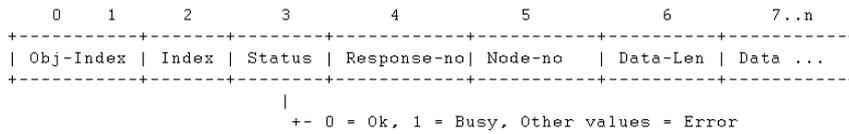
The user has to see to it that enough space is reserved for the longest data record in the FB-data.

The SDO-transfer is always initiated by the FB-Master and it is completed when Request-no. in the FB-output-data is identical to the Response-no. in the FB-input-data.

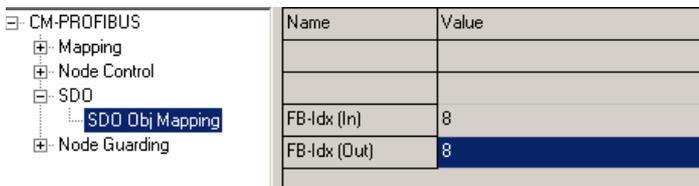
**Data record structure (FB-input = data from the FB-Master):**



**Data record structure (FB-output = data to the FB-Master):**



**Example:**



In this example the PB-Idx (In) and PB-Idx (Out) equal 8

**Please note:**

The counting starts at index "0" = 1. byte, i. e. in our example the value "8" corresponds to the 9. byte!

We want to query the manufacturer hardware version (object - index 1009h) from a connected CANopen® Slave. The CANopen® Slave has the Node ID = 5.

The Data-Len Request's size has to be selected as large as the receive data length that is mainly to be expected! Otherwise data that exceeds the set length is cut off.

If the size Data-Len Request is the same or larger than the receive data length, then the actual data length comes back as Data-Len Response.

Data from the FB-Master (PLC)

Data to the FB-Master (PLC)

Request

Response

FB-Idx (In)	Data (Hex)	Description
0	...	1. byte
1	...	2. byte
2	...	3. byte
3	...	4. byte
4	...	5. byte
5	...	6. byte
6	...	7. byte
7	...	8. byte
8	10	Obj-Index
9	09	Obj-Index
10	00	Sub-Index
11	01	Command (1 = read)
12	01	Auftrag-Nr (1. Auftrag)
13	05	Knoten-Nr (Node-ID = 5)
14	02	Data Len (data length = 2)
15	...	Data...
16	...	17. byte

FB-Idx (In)	Data (Hex)	Description
0	...	1. byte
1	...	2. byte
2	...	3. byte
3	...	4. byte
4	...	5. byte
5	...	6. byte
6	...	7. byte
7	...	8. byte
8	10	Obj-Index
9	09	Obj-Index
10	00	Sub-Index
11	00	Status (0 = ok)
12	01	Auftrag-Nr (1. Auftrag)
13	05	Knoten-Nr (Node-ID = 5)
14	02	Data Len (data length = 2)
15	47	Data... (47h = hardw.vers. „G“)
16	...	17. byte

### 16.3.1.11 Slave mode

The Slave mode can be activated through WINGATE®. In case the Slave mode is enabled, then only the starting sequence of the master function is not transferred. It is not a CANopen® Slave conforming to the standards and an EDS file is not supplied.

----- CM -----	
(CM) Transport protocol	<b>CANopen Mapping</b>
(CM) Profile	<b>map ...</b>
(CM) Slave Mode	<b>enabled</b>
(CM) CAN Baudrate	<b>125 kbs</b>
(CM) CAN Fieldbus ID	<b>0</b>

### 16.3.1.12 CAN Baud rate and address

The CAN baud rate and Node-ID (only in Slave mode) can be set via WINGATE.

----- CM -----	
(CM) Transport protocol	<b>Mapping</b>
(CM) Profile	<b>map ...</b>
(CM) Slave Mode	<b>enabled</b>
(CM) CAN Baudrate	<b>125 kbs</b>
(CM) CAN Fieldbus ID	<b>0</b>

### 16.3.1.13 Starting phase

During the starting phase the UNIGATE® sends the NMT-command "BusStart" every 2 seconds, that means all connected CANopen®-Slaves are set into the condition "Operational".

As soon as the Gateway has received the first PDO, that is available in the Mapping, this starting phase is left, which means that the NMT-command is not sent any more. There is no such thing as a Timeout here since the CANopen®-Slaves do not have to react to this NMT-command. In case a CANopen®-Slave is later connected in the net, it will consequently not receive this NMT-command any more. However, it can be re-activated any time through the FB-command-byte (bit 6) in order to put later CANopen®-Slaves into the condition "Operational".

### 16.3.1.14 Data cycle time

Cycle times of data through the Gateway:

Among other things the actual cycle time of data from CANopen® to the fieldbus or the other way round depends on the following parameters:

- Amount of Mappings
- Length of the configured fieldbus data
- Amount of COB-IDs used at the Mappings
- Amount of changed fieldbus data between two transmissions
- Amount of mapped bytes in a COB-ID
- Baud rate in the fieldbus or CAN

On account of the large amount of parameters and their interdependence a formula for the calculation of the cycle time cannot be stated. From practical measurements cycle times of 10 ms up to 100ms were observed, whereas the maximum time at maximum mappings was measured. In practice, on "regular" conditions, a cycle time of of some milliseconds is to be expected. In order to optimize the cycle time, the following guidelines should be kept:

- Amount of Mappings as small as possible
  - Length of the fieldbus-configuratio as short as possible
- Use of all 8 bytes in the COB-IDs if possible => Use of the least possible amount of COB-IDs.

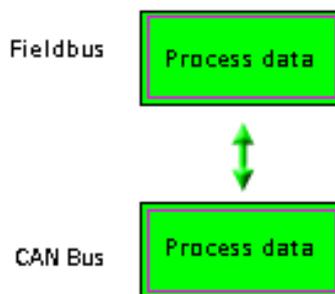
### 16.3.2 Layer 2 11Bit

Transparent data exchange with Fieldbus.

A maximum of 10 Byte of data can be mapped on the fieldbus side. (2 Byte COD-ID, 8 Byte payload).

The fastest possible data exchange is possible with this setting.

#### Data structure



#### Data mapping on the fielbus:

##### Example:

```

01 81 11 22 33 44 55 66 77 88 00 00 00
01 81 A1 A2 A3 A4 A5 A6 A7 A8 00 00 00 ...
  
```

A CAN message with COB-ID 181 and the data content "11 ... 88" in the first frame and "A1 ... A8" in the second frame has been sent to the fieldbus.

Example for sending a message from fieldbus to CAN:

COB-ID should be 182, data content „0x01 ... 0x04“

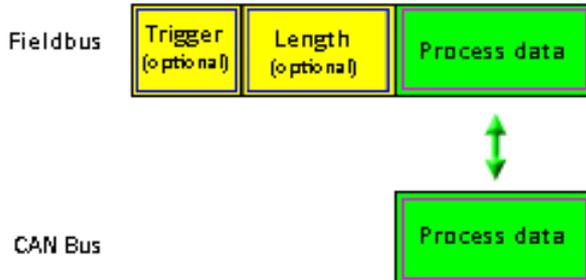
SPS Out (in Hex): 01 82 01 02 03 04

No further configuration like trigger bytes or fieldbus length byte are possible.

There is no diagnostic output for the fieldbus status on the CAN side as well as on the fieldbus side.

### 16.3.3 Universal (L2 11Bit)

#### Data structure



Identical with protocol "Layer 2 11Bit" (see chapter 16.3.2) However there are additional configurations possible on the fieldbus side:

Fieldbus trigger byte (i.e. fieldbus data exchange = On Trigger), fieldbus length byte and swap word.

Die Scriptdurchlaufzeit erhöht sich da zusätzliche Routinen durchlaufen werden.

### 16.3.4 Universal (L2 11Bit) with COB-ID used

Additional configurations are possible:

Up to 16 COB-ID's can be configured to which the CAN interface can react to.

All others are filtered (not processed).

If the value "0000" is displayed in the first entry „(CM) COB-ID used(1/16)“ it will also be reacted to messages with COB-ID NULL

If in one of the other entries „(CM) COB-ID used(2...16/16)“ the value is NULL, it is considered the end of the list and all subsequent entries will not be processed.

#### Example:

----- CM -----	
(CM) Transport protocol	<b>Universal (L2 11Bit) with COB-ID used</b>
(CM) CAN Baudrate	<b>125 kbs</b>
(CM) COB-ID used(1/16)	<b>0000</b>
(CM) COB-ID used(2/16)	<b>0181</b>
(CM) COB-ID used(3/16)	<b>0201</b>
(CM) COB-ID used(4/16)	<b>0080</b>
(CM) COB-ID used(5/16)	<b>0000</b>
(CM) COB-ID used(6/16)	<b>0000</b>
(CM) COB-ID used(7/16)	<b>0000</b>
(CM) COB-ID used(8/16)	<b>0000</b>
(CM) COB-ID used(9/16)	<b>0000</b>
(CM) COB-ID used(10/16)	<b>0000</b>
(CM) COB-ID used(11/16)	<b>0000</b>
(CM) COB-ID used(12/16)	<b>0000</b>
(CM) COB-ID used(13/16)	<b>0000</b>
(CM) COB-ID used(14/16)	<b>0000</b>
(CM) COB-ID used(15/16)	<b>0000</b>
(CM) COB-ID used(16/16)	<b>0000</b>

In this example only the COB-ID's „0000“ (hex), „0181“ (hex), „0201“ (hex) and „0080“ (hex) are processed, i.e. transferred to the fieldbus.

### 16.3.5 Universal (L2 11/29Bit)

A maximum of 13 Byte of data are displayed on the fieldbus side (1 Byte frame info, 4 Byte COB-ID, 8 Byte payload).

Additional configurations on the fieldbus side are possible:

Fieldbus trigger byte (i.e. data exchange = On Trigger), fieldbus length byte and Swap word.

#### Data display on the fieldbus side (SPS receives CAN) „11 Bit Mode“:

1. Byte number of received data from CAN (Low Nibble)  
MSB = 0 => 11Bit Frame received
2. Byte not used
3. Byte not used
4. Byte COB-ID High Byte
5. Byte COB-ID Low Byte
6. Byte Data n
7. Byte Data n+1
8. ...

Example 1. Line: 8 Payload, COB-ID 181, Payload 11 ... 88

Example 2. Line: 3 Payload, COB-ID 203, Payload 01 02 03

```
08 00 00 01 81 11 22 33 44 55 66 77 88
03 00 00 02 03 01 02 03 00 00 00 00 00
```

#### Data display on the fieldbus side (SPS receives from CAN) “29 Bit Mode”:

1. Byte number of received data from CAN (Low Nibble)  
Bit 7 (MSB) = 1 => 29Bit Frame received  
Bit 6 = RTR  
Bit 5 ... 0 = not used
2. Byte COB-ID High Byte
3. Byte COB-ID
4. Byte COB-ID
5. Byte COB-ID Low Byte
6. Byte Data n
7. Byte Data n+1
8. ...

Example 1. Line: MSB = 1 => 29 Bit Frame, 8 Payload, COB-ID 181, Payload 11 ... 88

Example 2. Line: MSB = 1 => 29 Bit Frame, 3 Payload, COB-ID 203, Payload 01 02 03

```
88 00 00 01 81 11 22 33 44 55 66 77 88
83 00 00 02 03 01 02 03 00 00 00 00 00
```

SPS sends a 29 Bit data frame to COB-ID 456, 5 payload, data 01 02 03 04 05

Example:

SPS (Out): 85 00 00 04 56 01 02 03 04 05

To CAN Bus:

ID (hex)	Daten (hex)
456	01 02 03 04 05 06 20 20

### 16.3.6 Universal (L2 11/29Bit) with COB-ID used

Identical with protocol Universal (L2 11/29Bit), see chapter 16.3.5, with additional configuration possibility.

Up to 16 COB-ID's can be configured to which the CAN interface can react to as described in chapter 16.3.4.

### 16.3.7 L2 11Bit (Tgl+FBlen)

See protocol Universal L2 11Bit, chapter 16.3.3, but with fixed trigger and fieldbus length byte (processing time/script run time optimized).

## 16.4 Delivery status (factory setting)

- Layer 2 11Bit, protocol description see chapter 16.3.2.

## 16.5 Trigger byte

Since the data is always transferred cyclically on the fieldbus, the gateway must detect when the user wishes to send new data via the serial interface. This is normally done by the gateway comparing the data to be transferred via the fieldbus with the old data stored internally - data exchange on change (fieldbus data exchange -> On Change). In many cases however, this cannot be used as the criterion, e.g. whenever the same data is to be sent. For this reason, the user can set control of transmission via a trigger byte (fieldbus data exchange -> On Trigger). In this mode, the gateway always sends (and only then) when the trigger byte is changed.

Accordingly, the application program in the control in Normal mode cannot detect whether the gateway has received several identical telegrams. If Trigger-Byte mode is activated, the gateway increments the trigger byte each time a telegram has been received.

The first byte in the fieldbus input/output data buffer is used as the trigger byte if this mode is activated.

## 16.6 Length Byte

The user can configure whether the transmit length is also to be stored as a byte in the input/output data area (Fieldbus length byte -> active). In transmit direction, as many bytes as specified in this byte are sent. On reception of a telegram the gateway enters the number of characters received.

## 16.7 Swap word

With activated "Swap word" the data is word-swapped to and from the fieldbus. I.e. High and Low byte are word-swapped and transferred in a 16-bit word to the CAN bus. It affects the entire fieldbus Buffer.

## 17 Technical data

### 17.1 Device data

The technical data of the module is given in the table below.

No.	Parameter	Data	Explanations
1	Location	Switch cabinet	DIN-rail mounting
2	Enclosure	IP20	Protection against foreign bodies and water to IEC 529 (DIN 40050)
4	Service life	10 years	
5	Housing size	23 x 115 x 117 mm (screw-plug-connector included) 23 x 1115 x 100 mm (screw-plug connector not included)	W x D x H
6	Installation position	Any	
7	Weight	160 g	
8	Operating temperature	-25°C ... +85°C	
9	Storage/transport temperature	-40 °C ... +85 °C	
10	Atmospheric pressure during operation during transport	795 hPa ... 1080 hPa 660 hPa ... 1080 hPa	
11	Installation altitude	2000 m 4000 m	Unrestricted Restricted - Ambient temperature ≤ 40°C
12	Relative humidity	Max. 80 %	No condensation, no corrosive atmosphere
14	External power supply	10...33 V DC	Standard power supply unit to DIN 19240
15	Current consumption at 24 VDC	Typ. 160 mA max 200 mA	
17	Reverse voltage protection	Yes	But does not function!
18	Short-circuit protection	Yes	
19	Overload protection	Poly-switch	Thermal fuse
20	Undervoltage detection (USP)	≤ 9 V DC	
21	Emergency power supply	≥ 5 ms	Device fully operable

Table: Technical data of the module

### 17.1.1 Interface data

The table below lists the technical data of the interfaces and ports on the device. The data has been taken from the corresponding Standards.

No.	Interface designation Physical interface	EtherNet/IP Ethernet 100 BaseT	RS232-C RS232-C	RS485/RS422 RS485/RS422	CANopen® RS485
1	Standard		DIN 66020	EIA Standard	CiA® DS 102
2	Transmission mode	Symmetrical asynchronous serial full-duplex  → Difference signal	Asymmetrical asynchronous serial full-duplex  → Level	Symmetrical asynchronous serial half-duplex full-duplex at RS422  → Difference signal	Symmetrical asynchronous serial half-duplex  → Difference signal
3	Transmission method	Multimaster CSMA/CD	Master / slave	Master / slave	Master / slave
4	Number of users : - Transmitters - Receivers	512 512	1 1	32 32	32 32
5	Cable length: - Maximum  - Depending on baud rate	1 00 m	15 m  no	1200 m  <93.75 kBd→1200 m 312, kBd→500 m 625 kBd→250 m	1300 m 50 kBd → 1300 m 100 kBd → 640 m 200 kBd → 310 m 500 kBd → 112 m 1 MBd → 40 m
6	Bus topology	Star	Point-to-point	Line	Line
7	Data rate: - Maximum - Standard	100 Mbit/s	120 kBit/s 2.4 k/B 4.8 k/B 9.6 kBit/s 19.2 kBit/s 38.4 kBit/s	520 kBaud 2.4 kBit/s 4.8 kBit/s 9.6 kBit/s 19.2 kBit/s 57.6 kB 312.5 kB	1Mbit/s  125 kB 250 kB 500 kB 1 MB
8	Transmitter: - Load - Maximum voltage - Signal, unloaded - Signal, loaded	100 Ω	3 ... 7 kΩ ± 25 V ± 15 V ± 5 V	54 Ω - 7 V ... 12 V ± 5 V ± 1.5 V	54 Ω - 7 V ... 12 V ± 5 V ± 1.5 V
9	Receiver: - Input resistance - Maximum input signal - Sensitivity	100 Ω	3 ... 7 Ω ± 15 V ± 3 V	12 Ω - 7 V ... 12 V ± 0.2 V	12 Ω - 7 V ... 12 V ± 0.2 V
10	Transmit range (SPACE): - Voltage level - Logic level		+ 3 ... + 15 V 0	- 0.2 ... + 0.2 V 0	<b>CANopen®</b>
11	Transmit pause (MARK): - Voltage level - Logic level		- 3 ... -15 V 1	+ 1.5 ... +5 V 1	<b>RS485</b>

Table: Technical data of the interfaces at the module

## 18 Commissioning guide

### 18.1 Note

Only trained personnel following the safety regulations may commission the UNIGATE®.

### 18.2 Components

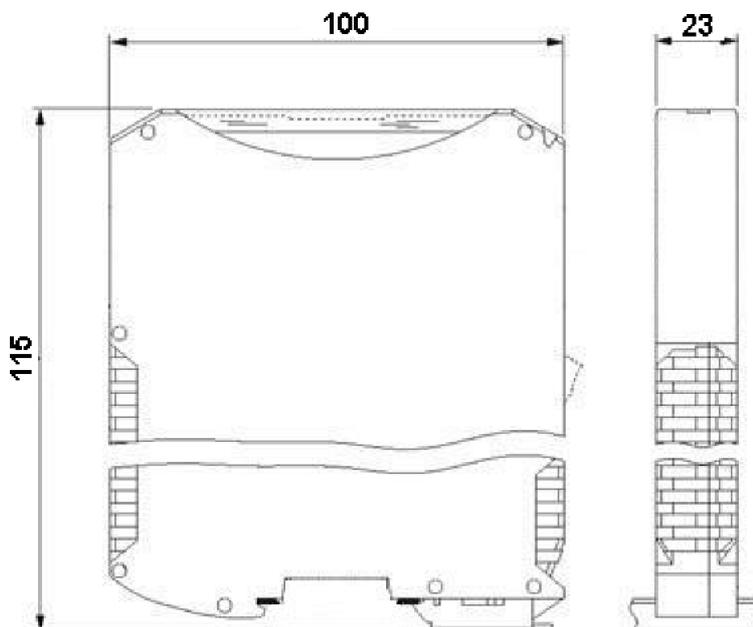
You will require the following components to commission the UNIGATE®:

- UNIGATE®
- Connection cable from gateway to the process
- Connection cable for CANopen connector
- Connector for EtherNet/IP connection to the Gateway
- Ethernet cable (this cable is generally available on site!)
- 10..33 V DC power supply (DIN 19240)
- Type file or EDS file and user manual (a sample EDS file as well as the user manual can be ordered separately or downloaded free of charge from our homepage at [www.deutschmann.de](http://www.deutschmann.de)).

### 18.3 Installation

The UNIGATE® CM - EtherNet/IP module features protection type IP20 and is thus suitable for switch cabinet use. The device is designed for snapping onto a 35 mm DIN-rail.

### 18.4 Dimensional drawing UNIGATE® CM - EtherNet/IP



### 18.5 Commissioning

It is essential that you perform the following steps during commissioning in order to ensure that the module operates correctly:

## 18.6 EtherNet/IP connection

Connect the device to the EtherNet/IP at the interface labelled "RJ45 EtherNet/IP".

## 18.7 Setting the CANopen address and baudrate

The setting of the CANopen-Node-ID and the baudrate is carried out via WINGATE or the script.



**Attention:**

*The CANopen address set must correspond to the planned address!*

*All users in the CANopen have to use the same Baud rate!*

## 18.8 CANopen connection

Connect the device to the CANopen at the interface X6.

## 18.9 Connection to the process device

Please also read the manual for the process device when commissioning the process device.

## 18.10 Shield connection

Earth the top-hat rail onto which the module has been snapped.

## 18.11 Connecting the supply voltage

Please connect 10..33 DC voltage to the terminals provided for this.

## 18.12 Project planning

Use any project planning tool for project planning.

If the required EDS file was not supplied with your project planning tool, a sample file can be found on the Internet ([www.deutschmann.com](http://www.deutschmann.com)).

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

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

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

Technical Support hours are as follows:

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

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

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

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

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

### 19.1 Returning a device

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

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

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

### 19.2 Downloading PC software

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

## 20 Annex

### 20.1 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
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

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 <sup>®</sup> 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 <sup>®</sup> IC for the respective product)
IB	=	Interbus
IBL	=	Interbus
LN62	=	LONWorks62
LN512	=	LONWorks512
ModTCP	=	ModbusTCP
MPI	=	Siemens MPI <sup>®</sup>
PL	=	Powerlink

PN = PROFINET IO  
PBDP = 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

## 20.2 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

