



**Deuschmann**

*your ticket to all buses*

**Instruction Manual  
Universal Fieldbus-Gateway  
UNIGATE® CM - PROFIBUS**



**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® block diagram	12
3.3	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
8.7	Accuracies of the baud rates	20
8.8	Script processing times	21
<b>9</b>	<b>Hardware ports, switches and LEDs</b>	<b>22</b>
9.1	Device labeling	22
9.2	Connectors	22

9.2.1	Connector to the external device (RS-interface)	22
9.2.2	Connector supply voltage and DEBUG-interface	23
9.2.3	CANopen interface connector	23
9.2.4	PROFIBUS DP-connector	23
9.2.5	Power supply	23
9.3	LEDs	24
9.3.1	LED "Power/State"	24
9.3.2	LEDs 1 / 2 / 4 / 8 (Error No. / Select ID)	25
9.3.3	LED "State"	25
9.3.4	LED "Bus Power"	25
9.3.5	LED "Bus"	25
9.3.6	LED "Bus State"	26
9.4	Switches	26
9.4.1	Termination Rx 422 + Tx 422 (serial interface)	26
9.4.2	Rotary coding switches S4 + S5 (serial interface)	26
9.4.3	Termination (PROFIBUS)	27
9.4.4	Rotary coding switches High + Low (PROFIBUS-ID)	27
9.4.5	CANopen termination	27
<b>10</b>	<b>Error handling</b>	<b>28</b>
10.1	Error handling at UNIGATE® CM	28
10.1.1	Error on the Extension	29
<b>11</b>	<b>Installation guidelines</b>	<b>30</b>
11.1	Installation of the module	30
11.1.1	Mounting	30
11.1.2	Removal	30
11.2	Wiring	30
11.2.1	Connection systems	30
11.2.1.1	Power supply	30
11.2.1.2	Equipotential bonding connection	31
11.2.2	PROFIBUS DP communication interface	31
11.2.2.1	Bus line with copper cable	31
11.2.3	Line routing, shield and measures to combat interference voltage	31
11.2.4	General information on line routing	31
11.2.4.1	Shielding of lines	32
<b>12</b>	<b>Firmware Update</b>	<b>33</b>
<b>13</b>	<b>PROFIBUS DP</b>	<b>34</b>
13.1	Description of the DPV1-/DPV2-functions	34
13.1.1	DPV1	34
13.1.2	DPV2	34
13.2	Representation of the data in PROFIBUS DP	34
13.2.1	Configuration telegram	34
13.2.2	Configuration telegram	35
13.2.3	Diagnosis	35
13.2.3.1	Diagnosis in DPV1	37
13.2.4	Data exchange	38

<b>14</b>	<b>Implemented protocols in UNIGATE® CM</b>	<b>39</b>
14.1	Implemented protocols (Transport protocols at the CM interface)	39
14.2	Configuration options for the individual transport protocol	40
14.2.1	Possibilities on the CANopen side	40
14.2.2	Possibilities on the fieldbus side	40
14.3	Operation of the transport protocols	41
14.3.1	CANopen Mapping	41
14.3.1.1	Configuration of the mapping	41
14.3.1.2	Node guarding	42
14.3.1.3	Displaying the node guarding states	43
14.3.1.4	Emergency Messages	44
14.3.1.5	Gateway control byte	44
14.3.1.6	Bit 7 (MSB): EMCY toggle bit	45
14.3.1.7	Bit 6: set operational toggle bit	45
14.3.1.8	Bit 5: RTR - Request to response	45
14.3.1.9	Gateway error byte	45
14.3.1.10	SDO Obj Mapping	45
14.3.1.11	Slave mode	47
14.3.1.12	CAN Baud rate and address	47
14.3.1.13	PROFIBUS Ident Nr. (Gerätetausch V2862)	47
14.3.1.14	Starting phase	48
14.3.1.15	Data cycle time	48
14.3.2	Layer 2 11Bit	48
14.3.3	Universal (L2 11Bit)	49
14.3.4	Universal (L2 11Bit) with COB-ID used	50
14.3.5	Universal (L2 11/29Bit)	50
14.3.6	Universal (L2 11/29Bit) with COB-ID used	52
14.3.7	L2 11Bit (Tgl+FBlen)	52
14.4	Delivery status (factory setting)	52
14.5	Trigger byte	52
14.6	Length Byte	52
14.7	Swap word	52
<b>15</b>	<b>Technical data</b>	<b>53</b>
15.1	Device data	53
15.1.1	Interface data	54
<b>16</b>	<b>Commissioning guide</b>	<b>56</b>
16.1	Note	56
16.2	Components	56
16.3	Installation	56
16.4	Dimensional drawing UNIGATE® CM - PROFIBUS DP	56
16.5	Commissioning	56
16.6	Setting the PROFIBUS address	57
16.7	PROFIBUS connection	58
16.8	Setting the CANopen address and baudrate	58
16.9	CANopen connection	58
16.10	Connection to the process device	58
16.11	Connecting the supply voltage	58

16.12	Shield connection	58
16.13	Project planning	58
16.14	Literature	58
<b>17</b>	<b>Servicing</b>	<b>59</b>
17.1	Returning a device	59
17.2	Downloading PC software	59
<b>18</b>	<b>Annex</b>	<b>60</b>
18.1	Explanations of the abbreviations	60
18.2	Hexadecimal table	61

#### 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-PROFIBUS DP module serves to adapt a CANopen interface to PROFIBUS according to EN 50 170. In this application, it functions as a Gateway and operates as the PROFIBUS DP Slave. It can be operated by any standard-compliant PROFIBUS Master.

The module CM-PROFIBUS DP essentially consists of the following hardware components:

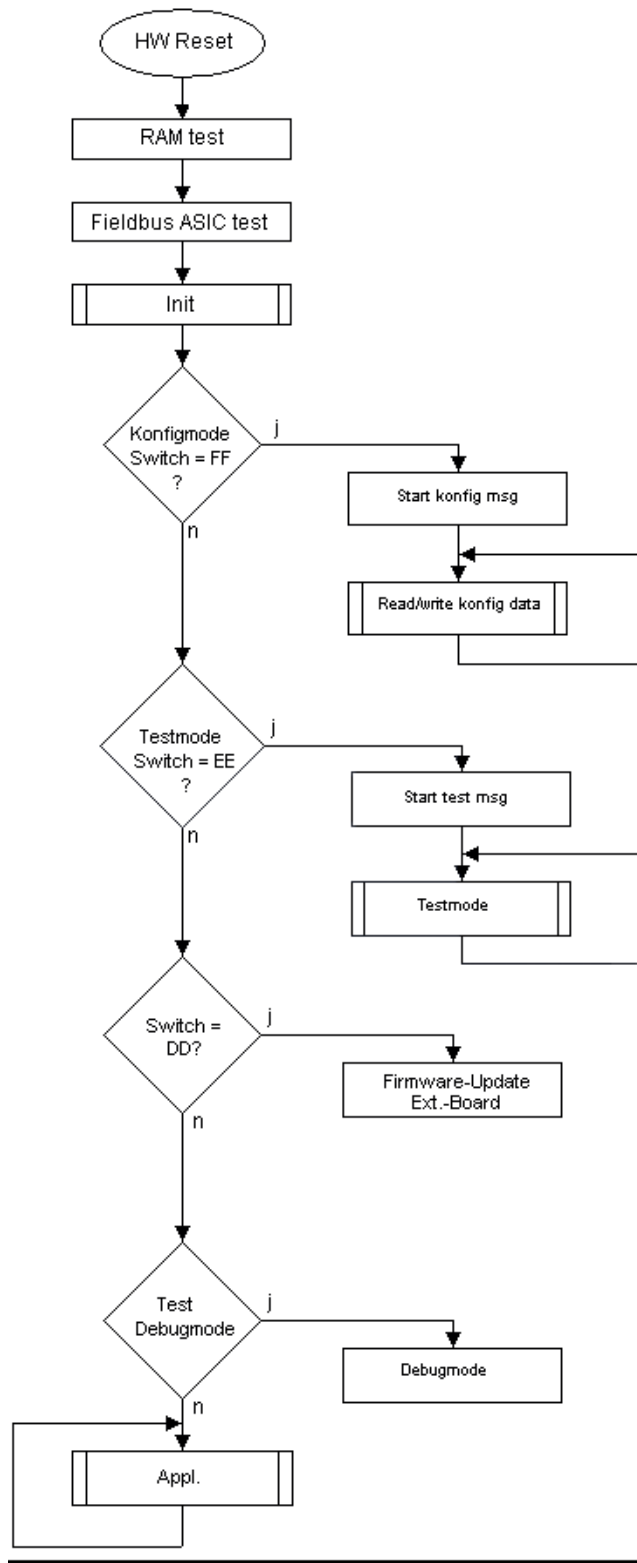
- Electrically isolated RS485 interface to the PROFIBUS DP
- PROFIBUS ASIC
- Microprocessor 89C51RE2
- RAM and EPROM
- Serial interface (RS232, RS485 and RS422)
- Electrically isolated interface to CANopen

#### Conceptual



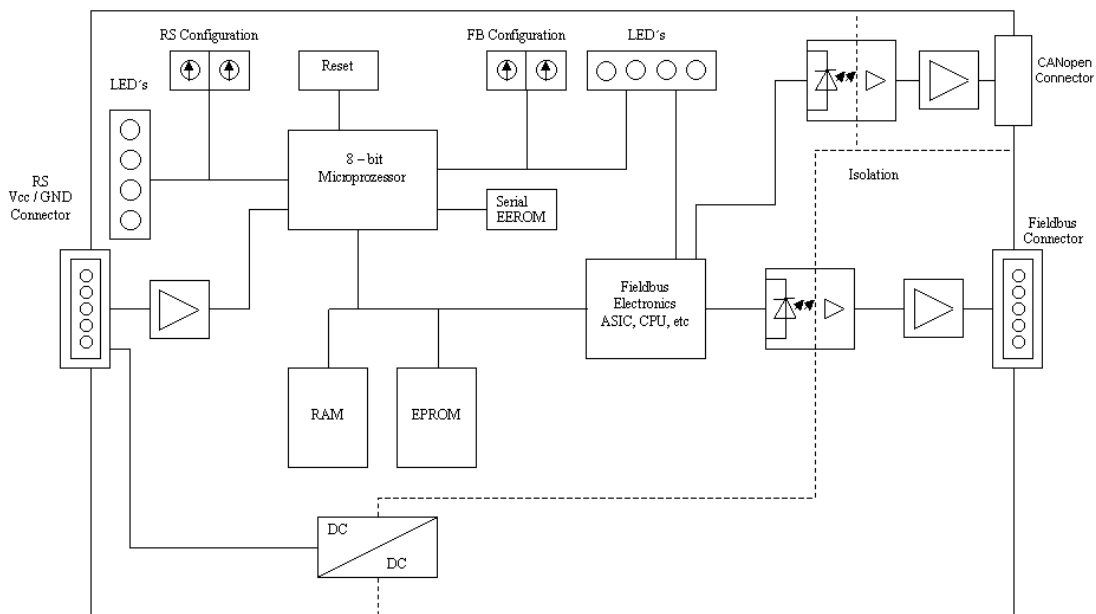
In the further course of this manual, the term "Fieldbus" is frequently used. In these cases this applies to the PROFIBUS DP interface.

### 3.1 UNIGATE® CM software flow-chart



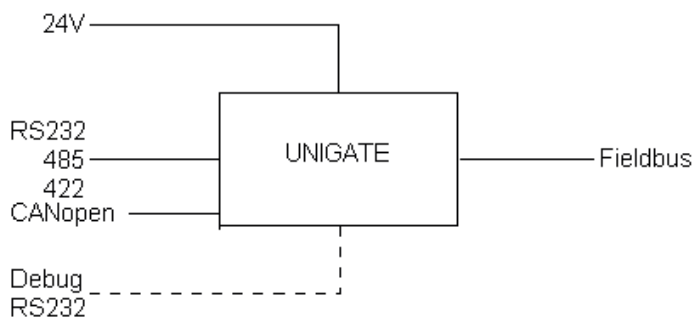
### 3.2 UNIGATE® block diagram

The following picture shows a typical UNIGATE®-module design.



### 3.3 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
- Updating the firmware (only CL- basis)
- Configuring 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-PBV1-CL (RS+Ext.Board) V8.1.2[41] (c)dA[40MHz] Switch=0x00FF Script(C:5719/16128,V:4687/8192)="Universalscript Deutschmann CM" Author="G/S" Version="V1.0.2"
Date=11.01.2016 SN=47110001 ID=0
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 - PROFIBUS DP 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<sup>®</sup> features a Debug-interface, that allows a step-by-step processing of a Script. Normally this interface is only required for the development of a Script.

### 6.2 Starting in the Debug-mode

When applying power to the UNIGATE<sup>®</sup> (power up) the firmware will output the binary character 0 (0x00) after a self-test was carried out on this interface. If the UNIGATE<sup>®</sup> 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 to 6 are blank, and Layer 7 is converted in accordance with chapter 7.3.

### 7.2 Interfaces

The Gateway features a CANopen and fieldbus 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

:

PROFIBUS	Input data	max. 244 Bytes	variable: maximum value in this case
	Output data	max. 244 Bytes	variable: maximum value in this case
	Parameters	8 Bytes	no user parameters *
	Configuration data	max. 16 Bytes	dependant on configuration *
	Diagnosis	max. 8 Bytes	1 user diagnostic byte = error code *

\*) Contrary to this default setting these parameters can be altered through special Script commands.

The commands can be found in the Protocol under “Bus specific“-“PROFIBUS“:

Set Profibus Config Data

Set Profibus Parameter InBuf

Set Profibus Diagnose

Further information can be found in the Protocol Developer’s online help.

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 Gateway is parameterized and configured by the Master 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. However, there are also exceptions to it. These settings are made by means of the software WINGATE. If you know our UNIGATE<sup>®</sup>-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

The baud rate of the serial interface is derived from the processor's crystal frequency.

Meanwhile all Script-Gateways are working with a crystal frequency of 40 MHz.

You can enter any desired integer baud rate into the Script. After that the firmware adjusts the baud rate, that can be derived the most precisely from the crystal frequency.

The baud rate the Gateway is actually working with (BaudIst) can be determined as follows:

$$\begin{aligned} \text{BaudIst} &= (\text{F32} / \text{K}) \\ \text{F32} &= \text{Crystal frequency [Hz]} / 32 \\ \text{K} &= \text{Round}(\text{F32} / \text{BaudSoll}); \\ &\quad \text{Round}() \text{ is a commercial roundoff} \end{aligned}$$

Example:

The actual baud rate is to be calculated, when 9600 baud are pre-set, where the Gateway is operated with 40 MHz:

$$\begin{aligned} \text{F32} &= 40000000 / 32 = 1250000 \\ \text{K} &= \text{Round}(1250000 / 9600) = \text{Round}(130.208) = 130 \\ \text{BaudIst} &= 1250000 / 130 = 9615.38 \end{aligned}$$

I. e.: The baud rate actually adjusted by the Gateway is 9615.38 baud

The resulting error in per cent can be calculated as follows:

$$\text{Error}[\%] = (\text{abs}(\text{BaudIst} - \text{BaudSoll}) / \text{BaudSoll}) * 100$$

In our example the following error results:

$$\text{Error} = (\text{abs}(9615.38 - 9600) / 9600) * 100 = 0.16\%$$

In practise errors below 2% can be tolerated!

In the following please find a listing of baud rates at a 40 MHz-crystal frequency with the corresponding errors:

4800 baud: 0.16%  
9600 baud:0.16%  
19200 baud:0.16%  
38400 baud:1.35%  
57600 baud:1.35%  
62500 baud:0%  
115200 baud:1.35%  
312500 baud:0%  
625000 baud:0%

## 8.8 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 0.5 ms 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.

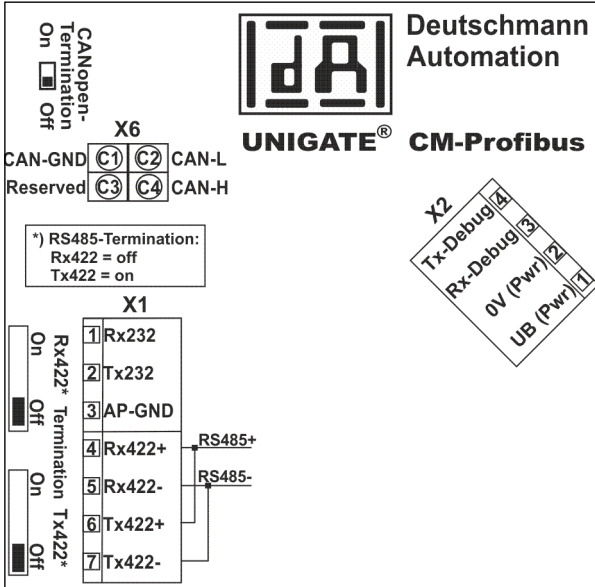
By means of the tasks mentioned above, the following recommendation can be formulated in order to receive a rather fast Script processing:

- 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.

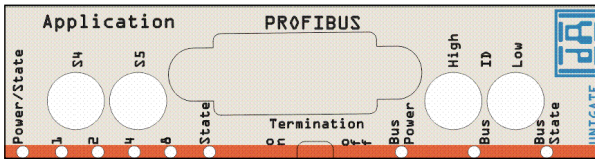
If the processing time should be too large in spite of these measures, there is the possibility to generate a customized Script command, that executes several tasks in one Script command. Please contact our support department for this purpose.

## 9 Hardware ports, switches and LEDs

### 9.1 Device labeling



Picture 1: Terminal labeling and termination



Picture 2: Front panel: Rotary switches, LEDs and termination PB



In case the front panel should pop out it does not affect the device's function or quality. It can be put in again without problems.

### 9.2 Connectors

#### 9.2.1 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+".

### 9.2.2 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



#### Attention:

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.

### 9.2.3 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

### 9.2.4 PROFIBUS DP-connector

The plug (labeled: PROFIBUS) for the connection to PROFIBUS is available on the front side of the device.

Pin assignment (9-pole D-SUB, socket)

Pin No.	Name	Function
1	Shield	
2		
3	B	Not inverting input-/output-signal from PROFIBUS
4	CNTR-P	Control signal / repeater
5	M5	DGND-data reference potential
6	P5	5 V supply voltage
7		
8	A	Inverting input-/output-signal from PROFIBUS
9		

### 9.2.5 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.

## 9.3 LEDs

The Gateway UNIGATE<sup>®</sup> CM - PROFIBUS features 9 LEDs with the following significance:

LED Power/State	red/green	Interface state Ethernet
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 Bus Power	green	Supply voltage PROFIBUS
LED Bus	red	PROFIBUS error
LED Bus State	red/green	Interface state PROFIBUS

### 9.3.1 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 28.1.1)
red bright	CL basis stopped, PC connection with Ext.-Board active (Firmware update, see chapter 12)

### 9.3.2 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".

### 9.3.3 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

### 9.3.4 LED "Bus Power"

This LED is connected directly to the electrically isolated supply voltage of the PROFIBUS-side.

### 9.3.5 LED "Bus"

This LED is directly controlled by the PROFIBUS ASIC and goes out in the state "DATA EXCHANGE".

### 9.3.6 LED "Bus State"

Lights green	PROFIBUS in the state data exchange
Flashes green	Gateway waits for PROFIBUS communication data
Flashes green/red	Gateway waits for PROFIBUS parameter data
Lights red	General PROFIBUS error

## 9.4 Switches

The Gateway features 8 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
Termination (PROFIBUS)	switchable PROFIBUS DP-terminating resistor
Rotary coding switch High	PROFIBUS-ID (high byte)
Rotary coding switch Low	PROFIBUS-ID (low byte)
CANopen-Termination	switchable CANopen terminating resistor

### 9.4.1 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.**

### 9.4.2 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 "EE" (testmode) and "FF" (config mode) are not possible for RS422- or RS485-operation.



**The switch position "DD" (ie, S4 and S5 in position "D") is reserved for internal purposes, Firmware-Update extension. The Gateway should only be switched into this mode for a Firmware-Update. Otherwise the firmware of the extension will be deleted and there is no access, respectively no function of the extension anymore. For details see chapter 12.**

## Switch positions

Switch positions S4	Switch positions S5	Function	Description
D	D	Firmware-Update Ext.-Board	(Description see chapter 12)
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.

### 9.4.3 Termination (PROFIBUS)

If the Gateway is operated as the first or last physical device in the PROFIBUS DP, 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 (220 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 PROFIBUS literature for further information on the subject of bus termination.

Note: To activate or deactivate the bus termination, please remove the bus-connector and carefully set the switch to the desired position.

### 9.4.4 Rotary coding switches High + Low (PROFIBUS-ID)

With these two switches the Gateway's PROFIBUS-ID (00... 7D) is set in hexadecimal notation. Please refer to the Annex for a conversion table from decimal to hexadecimal. This value is read in only once when the Gateway is activated. The value can also be read-out or analyzed through the Script command "Get (FieldbusID, LongTemp)".

See also chapter 16.6.

### 9.4.5 CANopen termination

If the Gateway is operated as the first or last physical device in the CANopen<sup>®</sup>, 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.

## 10 Error handling

### 10.1 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**).

### 10.1.1 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

## 11 Installation guidelines

### 11.1 Installation of the module

The module with the dimensions 23 x 117 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).

#### 11.1.1 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>.

#### 11.1.2 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°.

### 11.2 Wiring

#### 11.2.1 Connection systems

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

- Standard screw-type/plug connection (power supply + RS)
- 9-pin D-SUB plug connector (PROFIBUS DP)

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.

c) The 9-pin D-SUB plug connector is secured with two screws with "4-40-UNC" thread. It is best to use a screwdriver with a blade width of 3.5 mm to screw the screw tight.  
Tightening torque: 0.2... 0.4 Nm

##### 11.2.1.1 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 labeling on the device.

### 11.2.1.2 Equipotential bonding connection

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

## 11.2.2 PROFIBUS DP communication interface

### 11.2.2.1 Bus line with copper cable

This interface is located on the module in the form of a 9-pin D-SUB socket on the front side of the housing.

- Plug the PROFIBUS connector onto the SUB-D socket labelled "PROFIBUS DP".
- Firmly screw the securing screws of the plug connector tight using a screwdriver.
- If the module is located at the start or end of the PROFIBUS line, you must connect the bus terminating resistor integrated in the gateway. In order to do this, slide the slide switch to the position labelled ...on...
- If the module is not located at the start or at the end, you must set the slide switch to position "off".

### 11.2.3 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.

### 11.2.4 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 PROFIBUS DP, 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.

#### 11.2.4.1 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!



## 12 Firmware Update

The UNIGATE<sup>®</sup> 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)

## 13 PROFIBUS DP

### 13.1 Description of the DPV1-/DPV2-functions

#### 13.1.1 DPV1

Die DPV1-Erweiterung besteht aus folgenden Funktionen:

*1. Acyclic data exchange with Class1-Master (e. g.: PLC)*

This function is optional for a DPV1-Slave. Our gateways support this function as a default. By means of this function the Class1-Master can read and write data from the slave acyclically. This data is processed by the script in the gateway. The channel for the acyclic data exchange is set up firmly during the parameterization.

*2. Acyclic data exchange with Class2-Master (e. g.: control unit)*

This option is optional for a DPV1-Slave as well. Our gateways support this function as a default. By means of this function the Class2-Master can read and write data from the slave acyclically. This data is processed by the script in the gateway. The channel for the acyclic data exchange is set up prior to every data exchange and closed again afterwards.

*3. Alarm handling*

Also the alarms are optional. If they are activated, they replace the device-specific diagnosis. At present our gateway does not support alarms.

Every DPV1-Slave must support the extended parameterization since it is determined in Octet 8 of the parameterization telegram whether it is a DPV0- or a DPV1-Slave.

A DPV1-Slave can also be operated at a DPV0-Master if the DPV1-functions remain turned off.

#### 13.1.2 DPV2

Die DPV2-enlargement consists of the following functions:

*1. Isochron Mode (IsoM)*

It means the clock-synchronous behavior of a bus system. This function is optional for a DPV2-Slave and is activated via the GSD-file. At present our gateway does not support this mode.

*2. Data Exchange Broadcast (DxB)*

It means the communication between Slaves (inter-communication). This function is optional for a DPV2-Slave and is activated via the GSD-file. At present our gateway only supports the function of the „Publisher“ (sending data to other Slaves). The function „Subscriber“ (receiving data from other Slaves) is not supported at present.

*3. Up- And Download*

This function is also optional for a DPV2-Slave and at present it is not supported by our gateway.

*4. Time-synchronization (Time stamp)*

This function is also optional for a DPV2-Slave and at present it is not supported by our gateway.

*5. Redundancy concept*

This function is also optional for a DPV2-Slave and at present it is not supported by our gateway.

### 13.2 Representation of the data in PROFIBUS DP

Any standard-compliant PROFIBUS DP Master can exchange data with the gateway. It is also possible to use very "simple" Master connections owing to the data structure.

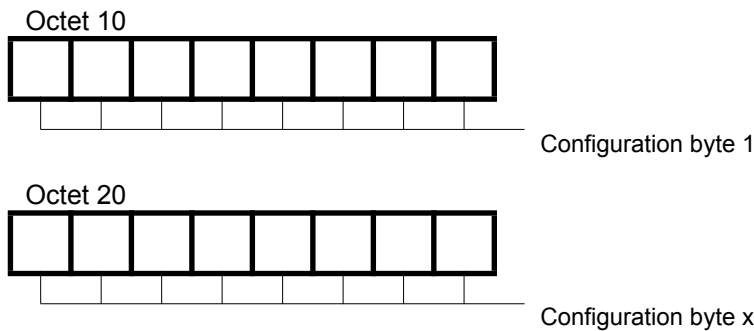
#### 13.2.1 Configuration telegram

After programming, the Master must send a configuration telegram to the corresponding Slave. The configuration telegram provides the Slave with information on the length of the input/output data. If the user has set the 'Length byte' flag, this means the maximum data lengths. Otherwise, it means the actual lengths.

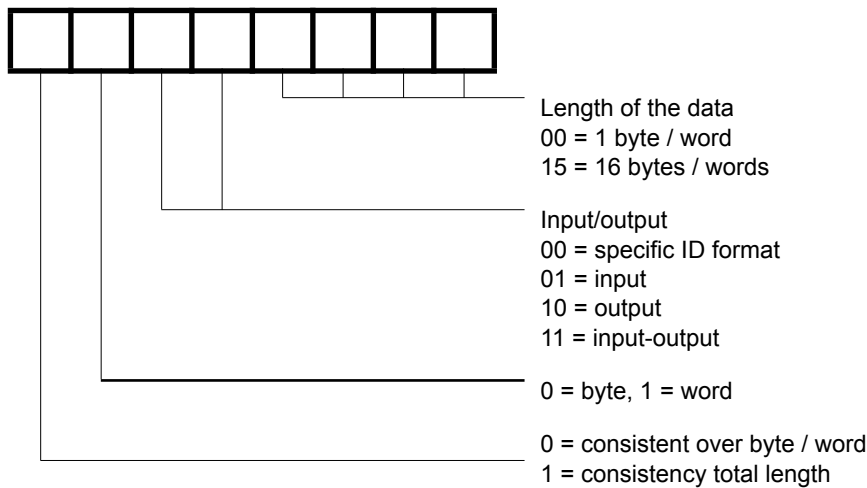
The user normally also configures the configuration telegram in the project planning tool where he may also, if necessary, enter the address range in which the useful data is stored.

You can write up to 16 bytes or words in one octet of the DataUnit (DU). Inputs and outputs having the same format can be combined in one octet. Otherwise, you must use as many octets as the number of different bytes/words you wish to use and which cannot be combined in one octet. If the module detects, during the check, that the maximum permitted input/output data lengths have been exceeded, it signals incorrect configuration to the Master during a subsequent diagnostic scan. It is then not ready for useful data communication.

### 13.2.2 Configuration telegram



Structure of an octet in the configuration telegram:



### 13.2.3 Diagnosis

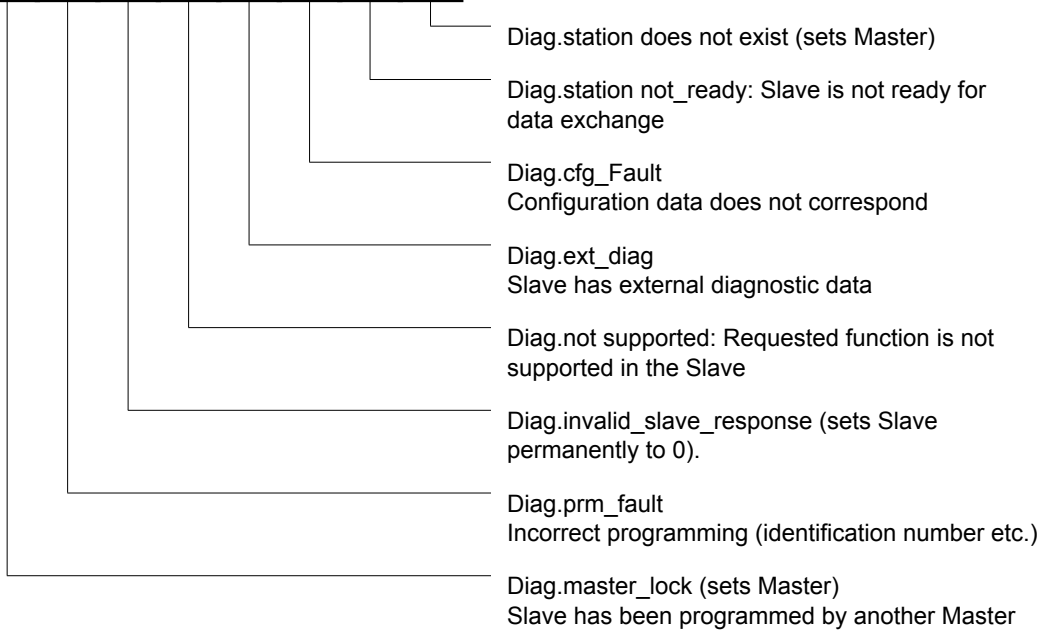
Diagnostic data is high-priority data. The gateway runs an external diagnostic if it detects an internal error.

Representation of the information in the external diagnostic byte:

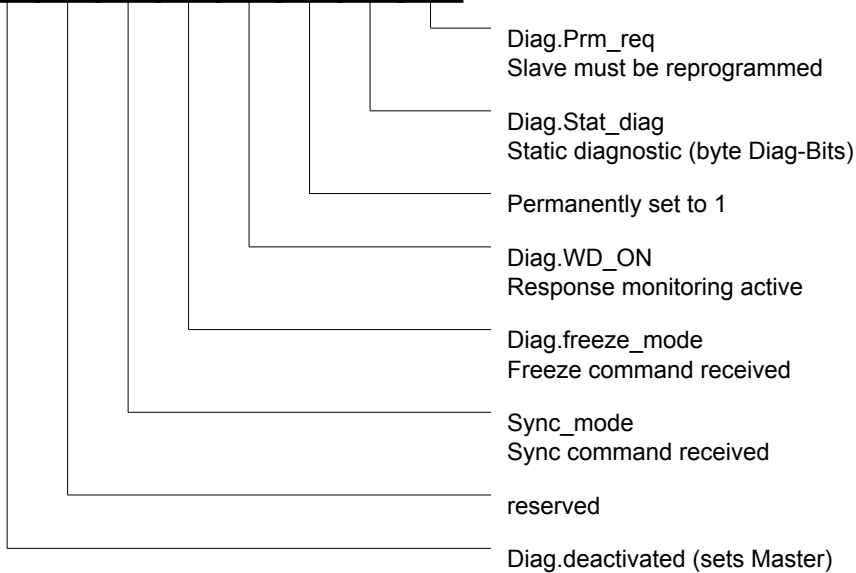
The diagnostic information of a DP Slave consists of standard diagnostic information items (6 bytes) and a user-specific diagnostic information item. (Error number)

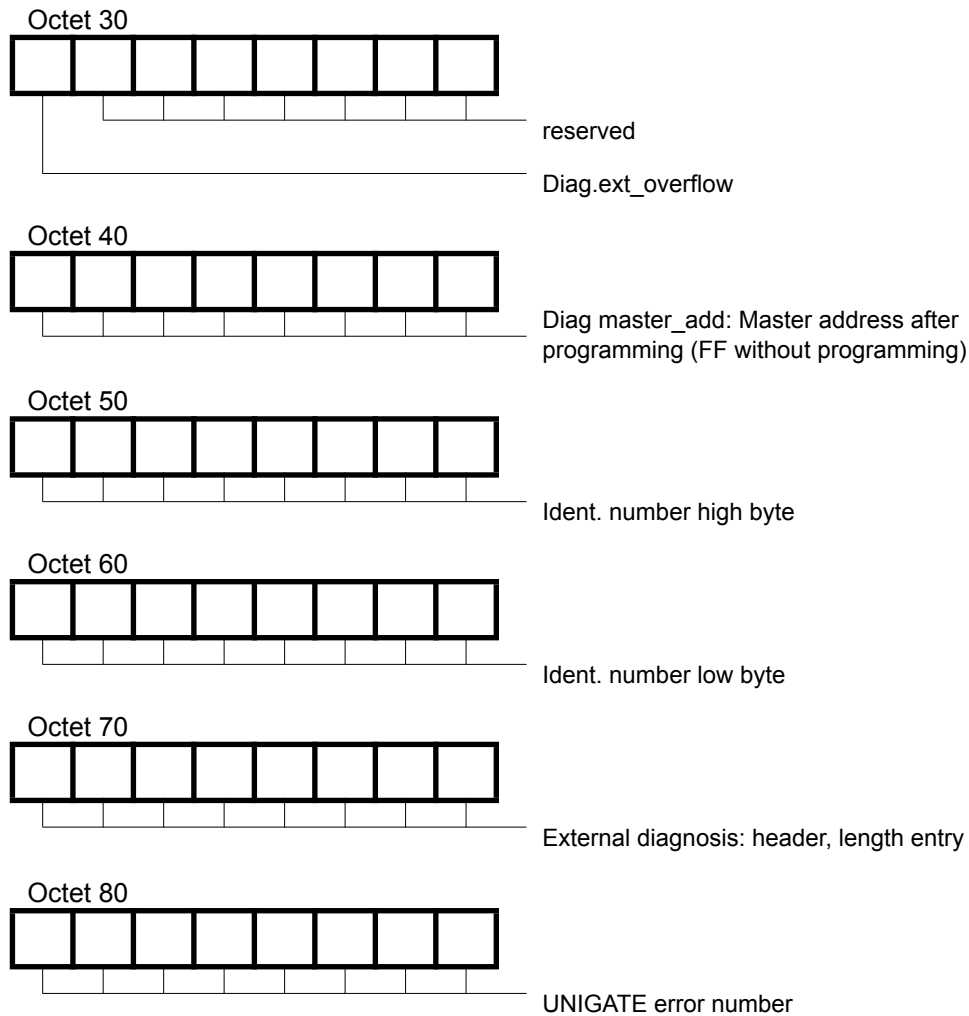
Telegram for diagnostic request:

Octet 10



Octet 20





### 13.2.3.1 Diagnosis in DPV1

The "external diagnosis" of DPV0 (old PB) is used differently at DPV1. At DPV1 the alarms and the status messages are transferred there. Since the gateway's error numbers have been transferred in the external diagnosis it became necessary then to carry out an adaptation at DPV1. Now the 3 bytes "0x81, 0x00, 0x00" have been inserted in front of the actual error message in order to be compatible with DPV1-masters. With it a DPV1-master identifies our gateway's error as status message now.

This results in a different representation of our gateway error in PROFIBUS. At DPV0 the error number is transmitted in unmodified form as 1 byte external diagnosis. In case DPV1 has been activated in the gateway through the GSD-file, the error number occurs as 1 byte status message.

Supposing the DPV1 is activated and a master (which does not support the alarms and status messages) is connected, then our gateway error number is presented as "external diagnosis" with 4 bytes (see above), at which point the 4th byte contains the error number and the preceding 3 bytes (0x81, 0x00, 0x00) can be ignored.

### **13.2.4 Data exchange**

After the Master, in the diagnostic, detects that the Slave is ready for data exchange, it sends data exchange telegrams. Either the Master stores the data in the input/output direction in the address area specified during project planning or the control program must fetch or retrieve the data using specific function blocks.

## 14 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.



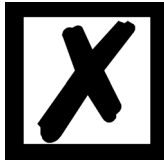
**Attention:**

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\.

### 14.1 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 PROFIBUS is in „Data-Exchange“.



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

## 14.2 Configuration options for the individual transport protocol

### 14.2.1 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 Fieldbus ID	✓						
(CM) COB-ID used(1/16)				✓		✓	
,	,	,	,	,	,	,	,
,	,	,	,	,	,	,	,
,	,	,	,	,	,	,	,
,	,	,	,	,	,	,	,
,	,	,	,	,	,	,	,
(CM) COB-ID used(16/16)				✓		✓	

### 14.2.2 Possibilities on the fieldbus side

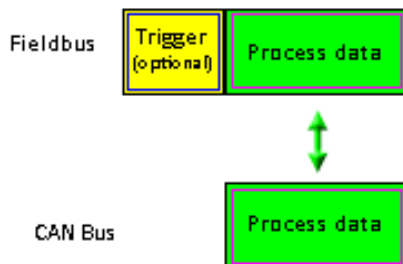
(CM) Transport protocol Fieldbus Parameter	CANopen Mapping	Layer 2 11Bit	Universal (L2 11Bit)	Universal (L2 11Bit) COB-ID	Universal (L2 11/ 29Bit)	Universal (L2 11/29Bit) COB-ID	L2 11Bit (Tgl+ FBlen)
Fieldbus ID	✓	✓	✓	✓	✓	✓	✓
Fieldbus Data exchange	✓		✓	✓	✓	✓	
Fieldbus length-byte			✓	✓	✓	✓	
Ident Number (0x2862)	✓						
Ext. Diag Off	✓		✓	✓	✓	✓	
Swap word	✓		✓	✓	✓	✓	



## 14.3 Operation of the transport protocols

### 14.3.1 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 **PROFIBUS DP** and **CANopen®**.

Here the Gateway on the CANopen® side acts as a standard-compliant master and on the ProfibusDP side as a standard-compliant slave so that the problems that appear again and again (to couple the CANopen® Slave participant to a Siemens PLC with ProfibusDP) 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:

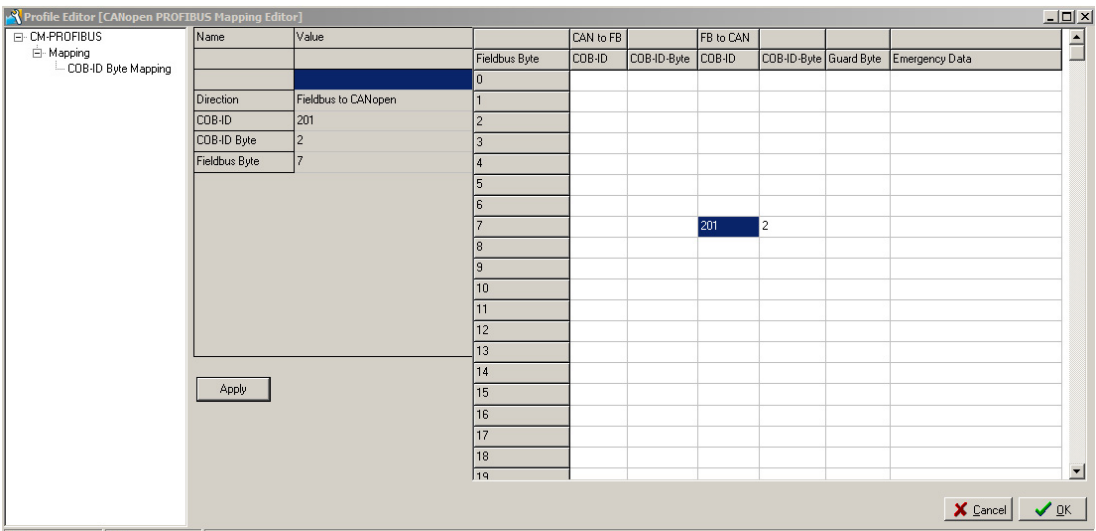
That way it can be defined that the 2nd byte of the COB-ID201 (CANopen®-participant) is transferred into the Profibus data as 7th byte.

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 14.3.1.10)

#### 14.3.1.1 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):



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®).**

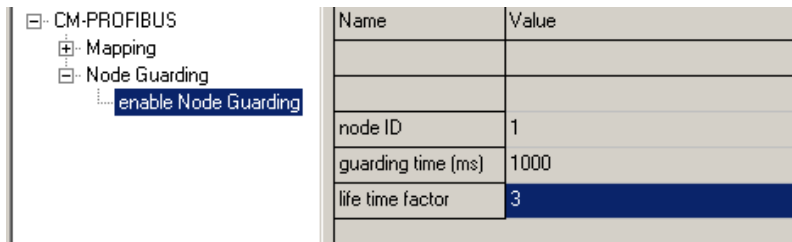
### 14.3.1.2 Node guarding

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**

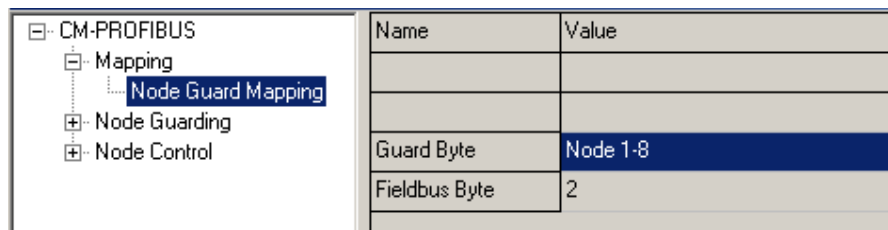
The parameter node ID indicates the ID of the CANopen® node on which the control is supposed to happen. With the guarding time the interval in which the the master calls on the slave to send its guarding answer is to be indicated in milliseconds.

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.

**14.3.1.3 Displaying the node guarding states**

The node guard mapping is used to represent the node guard states of the single slaves to the PROFIBUS. 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 Profibus data of the Gateway. So they are shown in the input area of the PROFIBUS master for this Gateway.



**Display of the node guarding information**

**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 was 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't answer to the node guarding and slave ID 2 works regularly. A mapping of information is entered into the PROFIBUS in the 2nd byte.

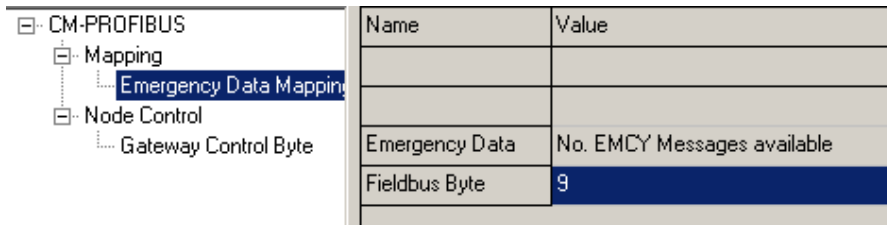
**Fieldbus data of the Gateway:**

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.

**14.3.1.4 Emergency Messages**

The gateway can receive 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 message 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 PROFIBUS is already visible with the toggle of the data

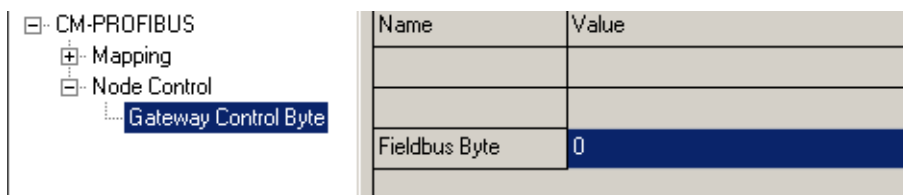


**Setting of the emergency messages**

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

**14.3.1.5 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**

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

#### 14.3.1.6 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.

#### 14.3.1.7 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.

#### 14.3.1.8 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.

#### 14.3.1.9 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.

CM-PROFIBUS	Name	Value
Mapping		
Node Control		
Gateway Control Byte		
Gateway Error Code	Fieldbus Byte	10

##### Setting of the error byte

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

#### 14.3.1.10 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:

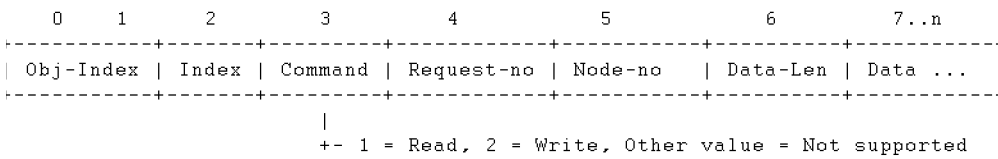
FB-Idx (In) = Data from the FB-Master (PLC) to UNIGATE

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

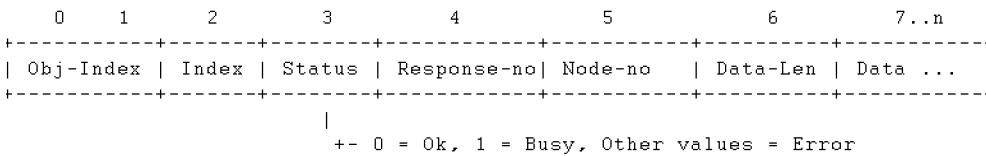
The user has to see to it that enough space is reserved for the longest data record in the FB-data.

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.

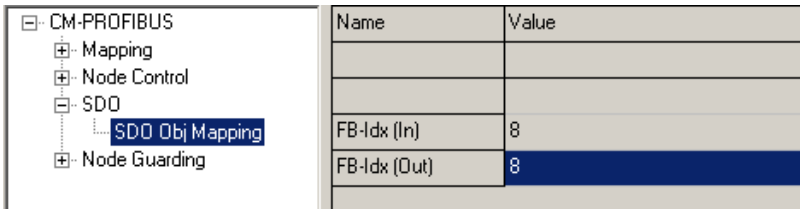
Data record structure (FB-input = data from the FB-Master):



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)

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)	14	02	Data Len(data length = 2)
15	...	Data...	15	47	Data... (47h = hardw.vers. „G“)
16	...	17. byte	16	...	17. byte

**14.3.1.11 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	map ...
(CM) Slave Mode	<b>enabled</b>
(CM) CAN Baudrate	<b>125 kbs</b>
(CM) CAN Fieldbus ID	<b>0</b>

**14.3.1.12 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	map ...
(CM) Slave Mode	<b>enabled</b>
(CM) CAN Baudrate	<b>125 kbs</b>
(CM) CAN Fieldbus ID	<b>0</b>

**14.3.1.13 PROFIBUS Ident Nr. (Gerätetausch V2862)**

-----FIELDBUS-----	
Fieldbus ID	<b>126</b>
Fieldbus Data exchange	<b>On Change</b>
Fieldbus lengthbyte	<b>inactive</b>
Ident Number (0x2862)	<b>disabled</b>
Ext. Diag Off	<b>Ident Number (0x2862)</b>
Swap word	<b>Ident Number (0x2862)</b>
----- Diagnose -----	<input checked="" type="radio"/> enabled <input type="radio"/> disabled
Diag-Monitor Application port	<input type="button" value="OK"/> <input type="button" value="Cancel"/>
8167016A1A5C0029010002CE	

The Ident Number is = 3218h. If you want to replace a UNIGATE CM device in an existing project with a V2862 UNIGATE® CO (M)-ProfibusDP, you can change back the Ident Number to the old Version 2862h via this parameter. As a result you can keep the old GSD file and you don't have to change the project.

Note: The configuration file (\*.gwc) of the UNIGATE® CO (M)-ProfibusDP cannot be loaded in the UNIGATE® CM.

There may be differences in the temporal data exchange.

#### 14.3.1.14 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".

#### 14.3.1.15 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 Profibus-configuration time of 244 byte I/O and 488 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-configuration as short as possible

Use of all 8 bytes in the COB-IDs if possible => Use of the least possible amount of COB-IDs.

### 14.3.2 Layer 2 11Bit

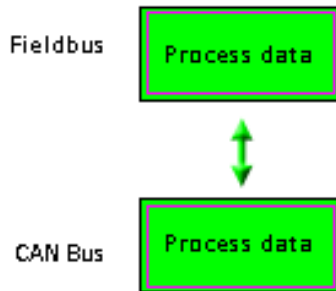
Transparent data exchange with Fieldbus.

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.

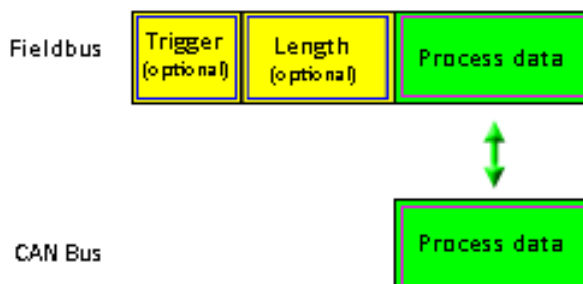
Example for sending a message from fieldbus to CAN:  
 COB-ID should be 182, data content „0x01 ... 0x04“  
 SPS Out (in Hex): 01 82 01 02 03 04

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.

**14.3.3 Universal (L2 11Bit)**

**Data structure**



Identical with protocol "Layer 2 11Bit" (see chapter 14.3.2) 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.

### 14.3.4 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.

### 14.3.5 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

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

Identical with protocol Universal (L2 11/29Bit), see chapter 14.3.5, 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 14.3.4.

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

See protocol Universal L2 11Bit, chapter 14.3.3, but with fixed trigger and fieldbus length byte (processing time/script run time optimized).

## 14.4 Delivery status (factory setting)

- Layer 2 11Bit, protocol description see chapter 14.3.2.

## 14.5 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.

## 14.6 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.

## 14.7 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.

## 15 Technical data

### 15.1 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 117 x 117 mm (screw-plug-connector included) 23 x 117 x 100 mm (screw-plug connector not included)	W x D x H
6	Installation position	Any	
7	Weight	160 g	
8	Operating temperature	-40°C ... +85°C	The negative temperatures are only valid for the usual conditions (not condensing)
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	At 10.8V. typ. 350 mA
16	Supply at the Profibus-interface	5 V DC / max. 50 mA	(Max. 50 mA at < 30°C ambient temperature)
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

### 15.1.1 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	ProfibusDP RS485	RS232-C RS232-C	RS485/RS422 RS485/RS422	CANopen® RS485
1	Standard	EIA Standard	DIN 66020	EIA Standard	CiA® DS 102
2	Transmission mode	Symmetrical asynchronous serial half-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	Master / Slave	Master / Slave	Master / Slave	Master / slave
4	Number of users : - Transmitters - Receivers	32 32	1 1	32 32	32 32
5	Cable length: - Maximum  - Baud rate-dependent	1200 m  93.75 kBd → 1200 m 187.5 kBd → 1000 m 500 kBd → 400 m 1.5 MBd → 200 m >1.5 MBd → 100 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	Line	Point-to-point	Line	Line
7	Data rate: - Maximum  - Standard values	12 Mbit/s  9.6 kBit/s 19.2 kBit/s 93.75 kBit/s 187.5 kBit/s 500 kBit/s 1.5 Mbit/s 3 MBit/s 6 MBit/s 12 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	625 kBaud 2.4 kBit/s 4.8 kBit/s 9.6 kBit/s 19.2 kBit/s 57.6 kB 312.5 kB 625 kB	1Mbit/s  125 kB 250 kB 500 kB 1 MB
8	Transmitter: - Load - Maximum voltage - Signal, unloaded - Signal, loaded	54 Ω - 7 V ... 12 V ± 5 V ± 1.5 V	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 - Max. input signal - Sensitivity	12 Ω - 7 V ... 12 V ± 0.2 V	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

<b>10</b>	Transmit range (SPACE):				
	- Voltage level	- 0.2 ... + 0.2 V	+ 3 ... + 15 V	- 0.2 ... + 0.2 V	- 0.5 ... + 0.05 V
	- Logic level	0	0	0	0
<b>11</b>	Transmit pause (MARK):				
	- Voltage level	+ 1.5 ... +5 V	- 3 ... -15 V	+ 1.5 ... +5 V	+ 1.5 ... +3 V
	- Logic level	1	1	1	1

Table: Technical data of the interfaces at the module

## 16 Commissioning guide

### 16.1 Note

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

### 16.2 Components

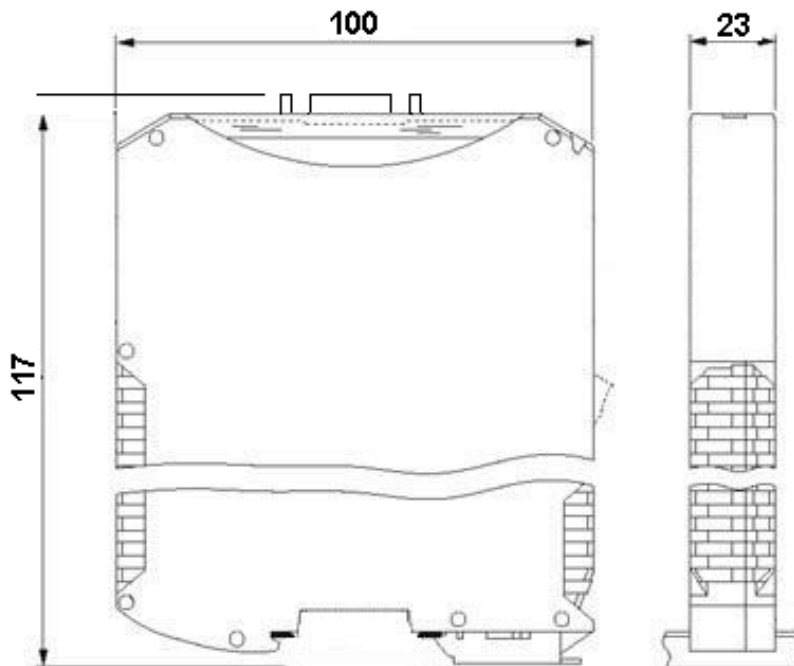
You will require the following components to commission the UNIGATE®:

- UNIGATE®
- Connection cable from gateway to the process
- Connection cable for CANopen connector
- Connector for Profibus connection to the Gateway
- PROFIBUS cable (this cable is generally already installed on site!)
- 10..33 V DC power supply (DIN 19240)
- Type file or GSD file and user manual (a sample 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)).

### 16.3 Installation

The UNIGATE® CM - PROFIBUS DP 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.

### 16.4 Dimensional drawing UNIGATE® CM - PROFIBUS DP



### 16.5 Commissioning

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



## 16.6 Setting the PROFIBUS address

Set the PROFIBUS address at the fieldbus end of the module on the two rotary switches designated "PROFIBUS-ID High" and "PROFIBUS-ID Low" This adjustment is carried out in a hexadecimal way.

Example:

The PROFIBUS-ID is 26 decimal = 1A hexadecimal

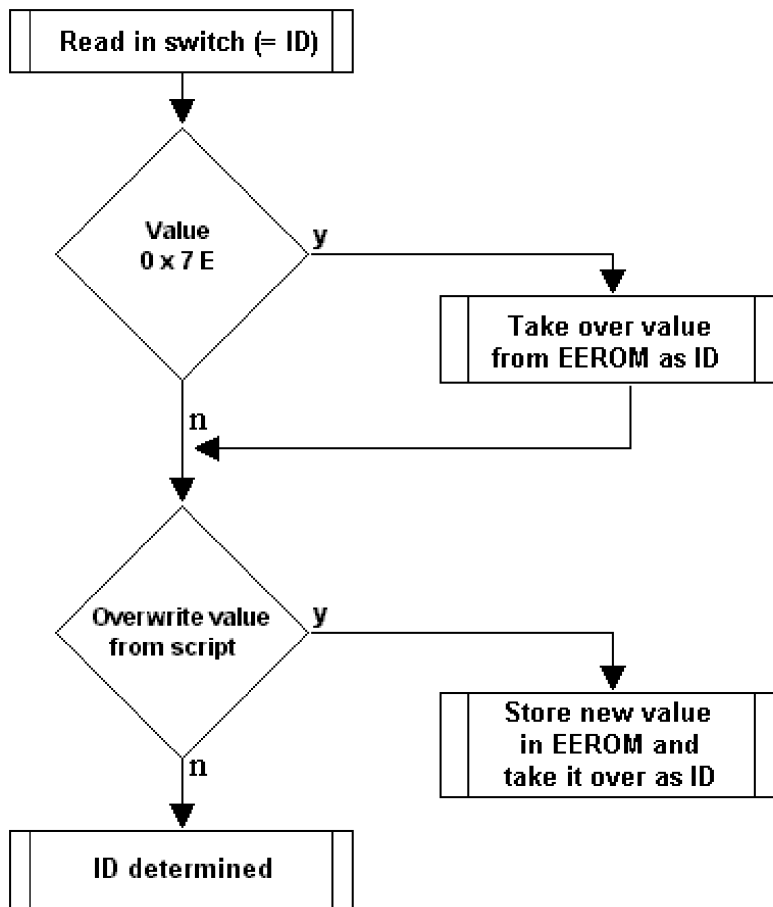
The switch "PROFIBUS-ID High" has to be set to 1 and the switch "PROFIBUS-ID Low" has to be set to A.

If the rotary switch is set to "7E" (=126) on the PROFIBUS-side (PROFIBUS-ID), the gateway operates with a PROFIBUS-address, that is stored in the EEROM. In the state of delivery this address is set to 126 and can only be changed by a PROFIBUS-master via the PROFIBUS itself. The address 126 is reserved in the PROFIBUS for this purpose; that means that a slave with this address can never carry out a data exchange, but it can only be configured with a new ID.

In case the rotary switch is set to a value between 0...125, the gateway operates - with this PROFIBUS-ID and a change via the master is not possible.

The ID can also be set through the Script command "Set (Fieldbus ID, xx)" (xx = PROFIBUS-ID in decimal form).

### Flow diagram



**Attention:**

The PROFIBUS address set must correspond to the planned address under COM PROFIBUS!

It is read in only on power-up of the gateway!

## 16.7 PROFIBUS connection

Connect the device to the Profibus at the interface labelled "PROFIBUS".

## 16.8 Setting the CANopen address and baudrate

The setting of the CANopen-Node-ID and the baudrate is carried out via 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!

## 16.9 CANopen connection

Connect the device to the CANopen at the interface X6.

## 16.10 Connection to the process device

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

## 16.11 Connecting the supply voltage

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

## 16.12 Shield connection

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

## 16.13 Project planning

Use any project planning tool for project planning.

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

## 16.14 Literature

We recommend that you read book "Schnelleinstieg in PROFIBUS DP", author M. Popp, to help you quickly get to grips with the subject of the PROFIBUS DP and the mode of operation of the available ASICs. The book can be ordered from the PROFIBUS User Organisation, Order No. 4.071. (German-language).

**Address:**

PROFIBUS Nutzerorganisation e.V.

Haid-und-Neu-Str. 7

D-76131 Karlsruhe

Germany

Tel: 0721 9658 590

---

## 17 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)

### 17.1 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.

### 17.2 Downloading PC software

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

## 18 Annex

### 18.1 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

## 18.2 Hexadecimal table

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

