Deutschmann

your ticket to all buses

Instruction Manual
Universal Fieldbus-Gateway
UNIGATE® CM - CANopen

Deutschmann Automation GmbH & Co. KG
www.deutschmann.com | wiki.deutschmann.de

Manual Art.-No.: V4010E

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.

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.

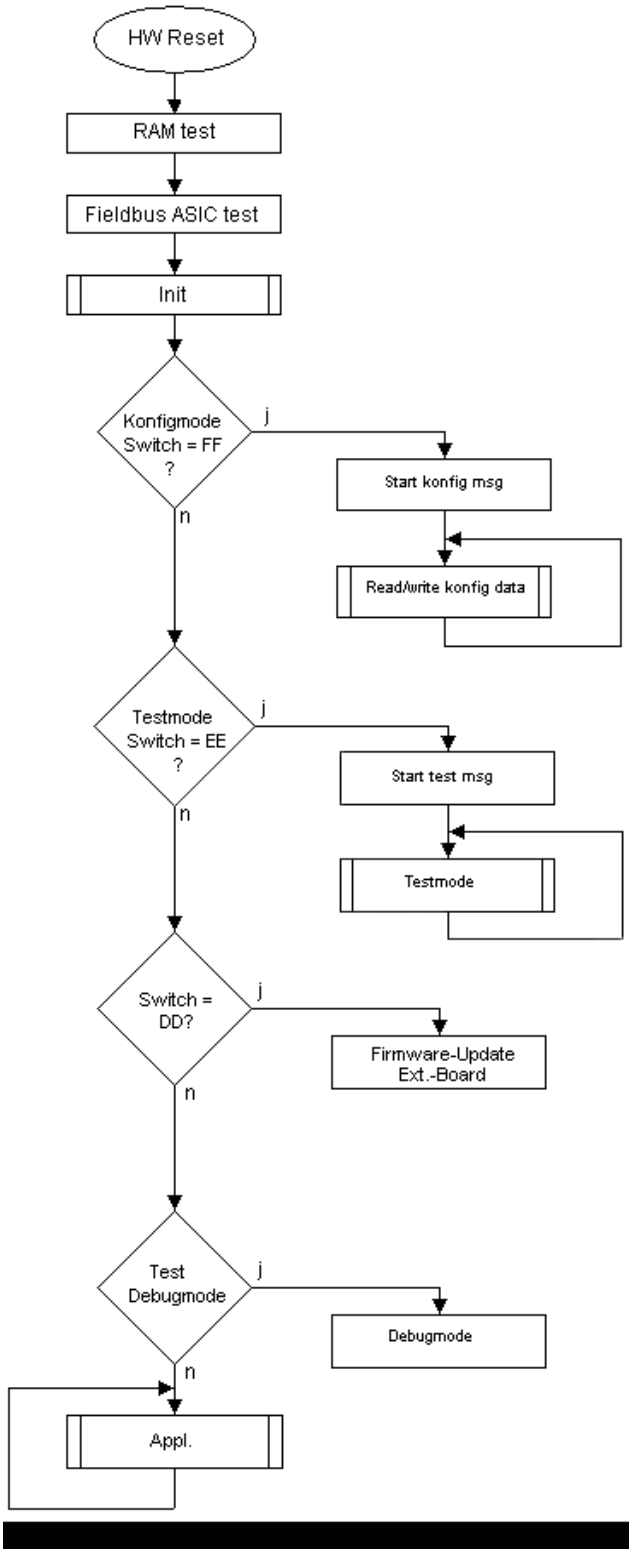
3 Introduction

The UNIGATE® CM-CANopen module serves to adapt a serial port to CANopen. In this application, it functions as a Gateway and operates as CANopen Slave. It can be operated by any standard-compliant Master.

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

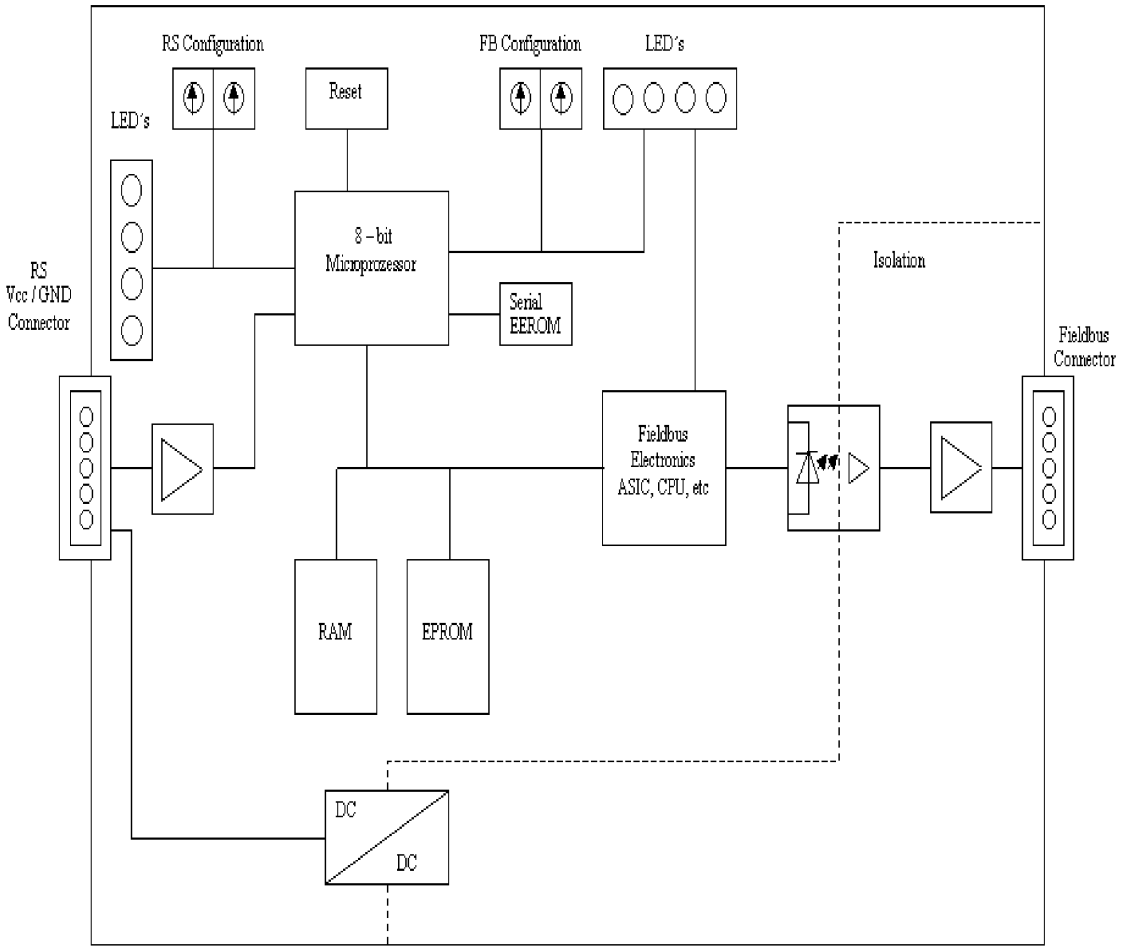
- Electrically isolated interface to CANopen
- CAN-controller SJA 1000
- Microprocessor 89C51RD2
- RAM and EPROM
- Serial interface (RS232, RS485 and RS422) to the device connected externally
- Ext.-Board with additional CANopen interface on the application side

3.1 UNIGATE® CM software flow-chart



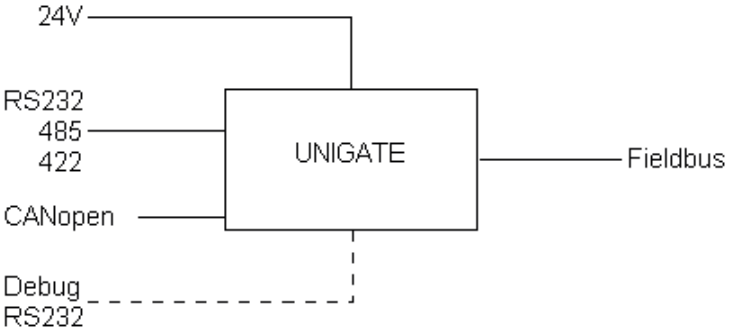
3.2 UNIGATE® block diagram

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

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-COV4-CL (RS+Ext.Board) V2.0 [29] (c)dA Switch=0xC1FF Script(8k)="Leer"
Author="Deutschmann Automation GmbH" Version="1.0" Date=21.08.2001 SN=47110001
Ext-Board: CL-Erweiterung(CANopen-IO-DICNET) V0.74 (c)dA SN=4294967295
CAN: ID=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 additional CANopen interface (application side):
1 byte: Echo of the first received byte via CAN
2 byte: Read back byte IO8
3. byte: value of the DIP-switch (currently not used, fixed value is "FF")
4. byte: Output to IO8

4.3 Data exchange mode

The Gateway has to be in the data exchange mode, so that a data exchange between the RS-side of the Gateway and the fieldbus is possible. As long as the Gateway is not in the configuration-, test-, firmware-update-, or debug mode, the data exchange mode is active. In the data exchange mode the Gateway will execute the downloaded Script.

5 RS-interface

5.1 RS-interfaces at the UNIGATE® CM

The UNIGATE® CM - CANopen has the interfaces RS232, RS422 and RS485 available. The hardware always features a DEBUG-interface, see chapter 7.

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 SSI-interface

The UNIGATE® also supports the connection of applications or products, that communicate via SSI.

6.1 Initiation of the SSI-interface

The required Script (`example_SSI`), the firmware- (Cust0023) and PROTOCOL DEVELPER-extension (`Cust_ssi.xml`) are available free of charge from our website at www.deutschmann.de, as well as the softwaretool Protocol Developer and the configuration software WINGATE.

- In the Protocol Developer (see chapter 7, The Debug-interface) the ConfigFile "`Cust_ssi.xml`" has to be added. At Options -> Settings -> ConfigFiles.
- Load the Script "example_SSI.dss" into the Protocol Developer.
- The encoder type and the clock frequency has to be defined in the Script itself under "`Set number of bits`" and "`Set type and clock stretch value`" (default = 12-Bit-Single-Turn-Gray, max. clock stretch):

```
// Set number of bits
//   1..16 = Single Turn
//  17..32 = Multi Turn
moveconst (bNumBits, 12);      // i.e. 12 bit single turn
// MT SSI 4096 x 4096 = 16777216 = 0b100000000000000000000000 => 24 bit

//-------------------------------------------------------------------------
// Set type and clock stretch value
//  Type (low nibble):
//    0 = Reserved
//    1 = output value as is (i.e. binary encoder)
//    2 = convert Gray encoded output value to binary (i.e. Gray encoder)
//   >2 = Reserved
//
//  Clock stretch value (high nibble):
//  Please note that the given frequency values are only a rough estimate. The
//  exact frequency varies depending on the devices underlying architecture.
//    0 = No Stretch --> ~300 kHz
//    1 = ~185 kHz
//    2 = ~150 kHz
//    3 = ~125 kHz
//    4 = ~110 kHz
//    5 = ~100 kHz
//    6 = ~ 88 kHz
//    7 = ~ 80 kHz
//    8 = ~ 72 kHz
```

```
//    9 = ~ 67 kHz
//    A = ~ 62 kHz
//    B = ~ 58 kHz
//    C = ~ 54 kHz
//    D = ~ 50 kHz
//    E = ~ 48 kHz
//    F = ~ 45 kHz

//moveconst ( wTyp, 0x02); // i.e. Gray encoder, no clock stretch (High-Nibble=0)
  moveconst ( wTyp, 0xF2); // i.e. Gray encoder, max clock stretch (High-Nibble=F)
```

- Load the Script into the device. Open WINGATE and activate the device in the configuation mode (see chapter 4.1, Configuration mode (config mode)) - an actuation message appears, that looks in line with the following (example CL-PB):
  Special Firmware (23) not loaded
  RS-PBV1-CL (232/422/485) V7.31[30] (c)dA Switch=0x02FF Script(8k)="SSI"
  Author="Deutschmann Automation" Version="V 1.0" Date=20.03.2008 SN=47110002 ID=2
  Konfigmode...
  The note "Special Firmware (23) not loaded" means that the firmware-extension is not yet loaded. The extension is loaded through Extras -> Firmware Script Extension. Select the file "`Cust0023 (Cmd 23 + 24 for SSI).hex`" and choose "`write extension`".
- Re-start the device → now only the device's actual actuation message appears and not the note any more.
- Bring the device into the data exchange mode (see chapter 4.3, Data exchange mode) → DONE!

6.2 Hardware-wiring

The clock wires of the SSI-interface are placed onto the Tx-wires of the RS422-interface and the data wires onto the Rx-wires at the UNIGATE® CM.

X1 (3-pin + 4-pin screw-plug-connector):

Pin no.	Name	Function at SSI
1	Rx 232	n. c.
2	Tx 232	n. c.
3	AP-GND	n. c.
4	Rx 422+	SSI DAT+
5	Rx 422-	SSI DAT-
6	Tx 422+	SSI CLK+
7	Tx 422-	SSI CLK-

7 The Debug-interface

7.1 Overview of the Debug-interface

The UNIGATE® IC 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.

7.2 Starting in the Debug-mode

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

7.3 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.

7.4 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’.

7.5 Commands of the Debug-interface

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

8 Mode of operation of the system

8.1 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 8.3.

8.2 Interfaces

The Gateway features the RS232-, RS422- and RS485-interfaces.

8.3 Data exchange CANopen V3

All data is transferred by the Gateway in dependence of the downloaded Script.
The following three objects are existing in the CANopen-gateway for the data exchange on the CANopen-side:

- Default setting as long as the Script command CO_Init_Channel is not carried out.
  - Adr. 2000H (Type DOMAIN):Data received by the gateway
  - Adr. 2001H (Type DOMAIN):Data sent by the gateway
  - Adr. 2002H (Type BYTE): Length of the data sent

The length of the receiving- and transmitting buffer (Obj. 2000 + 2001) is configured through WINGATE®.

8.3.1 SDO-access

Generally the data can always be exchanged through SDOs (Obj. 2000 - 2002).
Likewise an access to all Mandatory-objects according to CiA® DS 301 is possible through SDOs.

8.3.2 PDO-access

PDOs are supported according to the following table depending on the configured length and the PDO-length is set dynamically to the correct value:

Gateway receiving-data	**Gateway transmitting-data**	**Receiv.-PDO1 (Adr = 512 + ID)**	**Transm.-PDO1 (Adr = 384 + ID)**
Max. 8 Byte	Max. 8 Byte	Receiv. data	Transm. data
Max. 8 Byte	>8 Byte	Receiv. data	Length transmitting data
>8 Byte	Max. 8 Byte	-	Transm. data
>8 Byte	>8 Byte	-	Length transmitting data

8.4 Possible data lengths

The table below shows the maximum transferable data in CANopen:.

Input data	max. 255 bytes	Variable: maximum value in this case
Output data	max. 255 bytes	Variable: maximum value in this case
Emergency data	1 byte	See chapter Error handling

9 Generating a Script

Note: All commands relating to the extension do not work in the debug mode!

9.1 What is a Script?

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

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

9.2 Memory efficiency of the programs

A Script command can carry out e. g. a complex checksum like a CRC-16 calculation via data. For the coding of this command only 9 byte are required as memory space (for the command itself). This is only possible when these complex commands are contained in a library.
A further advantage of this library is, that the underlying functions have been in practical use for a couple of years and therefore can be described as 'void of errors'. As these commands are also present in the native code for the controller, at this point also the runtime performance of the Script is favorable.

9.3 What can you do with a Script device?

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

9.4 Independence of buses

Basically the Scripts do not depend on the bus, they are supposed to operate on. It means that a Script which was developed on a PROFIBUS Gateway can also be operated on an Interbus without changes, since the functioning of these buses is very similar. In order to also process this Script on an Ethernet Gateway, perhaps further adjustments have to be made in the Script, so that the Script can be executed reasonably.
There are no fixed rules how which Scripts have to operate properly. When writing a Script you should take into account on which target hardware the Script is to be executed, so the necessary settings for the respective buses can be made.

9.5 Further settings at the Gateway

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

9.6 The use of the Protocol Developer

The Protocol Developer is a tool for an easy generation of a Script for our Script Gateways. Its operation is exactly aimed at this use. After starting the program the Script that was loaded the last time is loaded again, provided that it is not the first start.
Typical for Windows Script commands can be added by means of the mouse or the keyboard. As far as defined and required for the corresponding command, the dialog to the corresponding command is displayed, and after entering the values the right text is automatically added to the Script. The insertion of new commands by the Protocol Developer is carried out in a way that existing commands will not be overwritten. Generally a new command is inserted in front of the one where the cursor is positioned. Of course the commands can also be written by means of the keyboard or already written commands can also be modified.

9.7 Accuracies of the baud rates

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:

BaudIst= (F32 / K)
F32 = Crystal frequency [Hz] / 32
K = Round (F32 / BaudSoll);
Round () is a commercial roundoff

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

F32 = 40000000 / 32 = 1250000
K = Round(1250000 / 9600) = Round(130.208) = 130
BaudIst= 1250000 / 130 = 9615.38

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:

Error[%] = (abs(BaudIst - BaudSoll) / BaudSoll) * 100
In our example the following error results:
Error = (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%

9.8 Script processing times

The Script is translated by the Protocol Developer and the consequently generated code is loaded into the Gateway. Now the processor in the Gateway interprets this code. In this case, there are commands that can be processed very fast (e. g. "`Set Parameter`"). There are also commands, however, that take longer (e. g. copying 1000 bytes). Consequently, for one thing the processing time differs due to the kind of Sript 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.

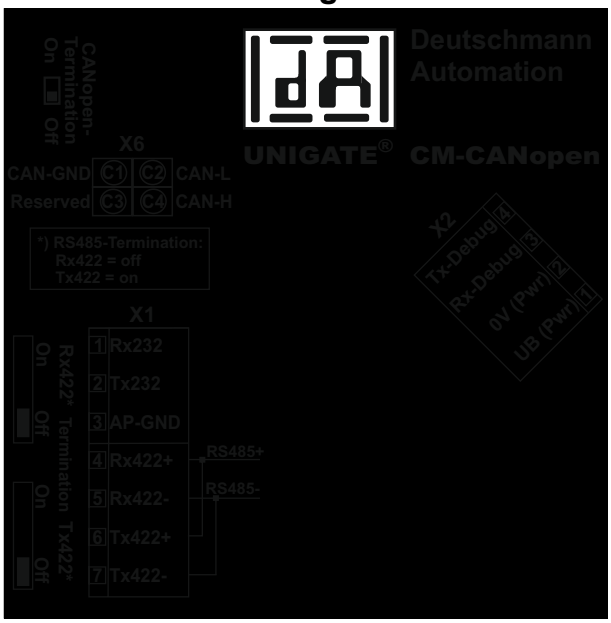
By means of the tasks mentioned above, the following recommendation can be formulated in order to receive a rather fast Script processing:

- 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 transfered per second.
- Do not unnecessarily extend the data length at the Fieldbus side. Especially at acyclical bus data, if possible do only send them when changes were made. The data length at buses that are configured to a fixed length (e. g. PROFIBUS) should not be longer than absolutely necessary.

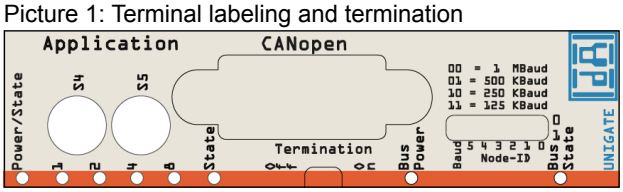
If the processing time should be too large in spite of these measures, there is the possibility to generate a customized Script command, that executes several tasks in one Script command. Please contact our support department for this purpose.

10 Hardware ports, switches and LEDs

10.1 Device labeling



Picture 1: Terminal labeling and termination



Picture 2: Front panel: Rotary switches, DIP-switch, LEDs and termination CO

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.

10.2 Connectors

10.2.1 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+".

10.2.2 Connector supply voltage and DEBUG-interface

Pin assignment X2 (4-pin 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.

10.2.3 CANopen interface connector (application side)

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

10.2.4 CANopen-connector

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

Pin assignment (9-pin D-SUB, plug)

Pin No.	Name	Function
1		
2	CAN-L	Dominant Low
3	CAN-GND	CAN Ground
4		
5		
6		
7	CAH-H	Dominant High
8		
9		

10.2.5 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.

10.3 LEDs

The Gateway UNIGATE® CM - CANopen features 8 LEDs with the following significance:

LED Bus Power	green	Supply voltage CANopen
LED Bus State	red/green	Interface state CANopen
LED Power / State	red/green	Interface state of additional CANopen interface (application side)
LED State	red/green	User-defined / general Gateway error
LEDs 1 / 2 / 4 / 8 (Error No. / Select ID)	green	User-defined / general Gateway error

10.3.1 LED "Bus Power"

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

10.3.2 LED "Bus State"

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 (approximately 1000 ms).

LED triple flash 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 indiates 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.

10.3.3 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: (see Chapter 10.3.2)

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

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

10.3.4 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

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

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

10.4 Switches

The Gateway features 7 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 (CANopen)	switchable CANopen-terminating resistor
DIP-switch	Node-ID and baud rate
CANopen-Termination	switchable CANopen-terminating resistor (application side)

10.4.1 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 Ω) 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 Ω) to ground and a pull-up resistor (390 Ω) to VCC.

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

10.4.2 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 Ext.-Board. 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.

Switch positions

Switch postions S4	Switch postions S5	Function	Description
D	D	Firmware-Update extension	(Description see chapter 13)
E	E	Test mode	(Description see chapter 4.2) **Note:** This mode can only be terminated by a reboot.
F	F	Config mode	(Description see chapter 4.1) **Note:** This mode can only be terminated by a reboot.

10.4.3 Termination (CANopen)

If the Gateway is operated as the first or last physical device in the CANopen, there must be a bus termination at this Gateway. In order to do this, either a bus terminating resistor must be activated in the connector or the resistor (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 Fieldbus 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.

10.4.4 CANopen termination (application side)

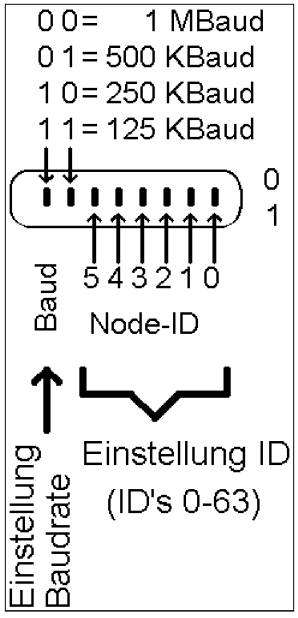
If the Gateway is operated as the first or last physical device in the CANopen, there must be a bus termination at this Gateway. In order to do this, either a bus terminating resistor must be activated in the connector or the resistor (120 Ω) integrated in the Gateway must be activated. In order to do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

10.4.5 DIP-switch

The DIP-switch is used to set the Node-ID and Baud rate according to picture 3.

If the Node-ID 0 (which is not allowed in CANopen) is set on the DIP-Switch, then the Node-ID, which is stored in the EEROM via the Script or WINGATE is used in this case. That way it is also possible to set Node-IDs > 63 and there is no unclarity if the Node-ID of the DIP-Switch or the Node-ID of the Script is valid.

In the initial state the Node-ID 1 is stored in the EEROM.



Picture 3: DIP-switch

11 Error handling

11.1 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. In the default state this error numer is additionally sent as emergency telegram via CANopen®. The code 61xx (hex), that indicates an internal Firmware error according to DS301 is used for it. The current error code is entered for “xx“ in accordance with the below table. A detail error, that is used for internal purposes may still be included in byte 3 and 4 of the emergency message.
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`".
Error 10 is additionally activated in case of the heartbeat-error.

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
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
0	1	0	0	4	Fieldbus hardware error or wrong Fieldbus-ID
0	1	0	1	5	Script error
0	1	1	0	6	Reserved
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

11.1.1 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

12 Installation guidelines

12.1 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).

12.1.1 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 mm².

12.1.2 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°.

12.2 Wiring

12.2.1 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 (CANopen)

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 mm²
- Solid conductor: 1 x 0.25 ... 1.5 mm²
- 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

12.2.1.1 Power supply

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

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

12.2.1.2 Equipotential bonding connection

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

12.2.2 CANopen communication interface

12.2.2.1 Bus line with copper cable

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

- Plug the CANopen® connector onto the SUB-D plug labelled "CANopen".
- 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 CANopen line, you must connect the bus terminating resistor integrated in the gateway. In order to do this, slide the slide switch to the position labeled ...on...
- If the module is not located at the start or at the end, you must set the slide switch to position "off".

12.2.3 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.

12.2.4 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 ≥ 60 V
- unshielded lines for AC voltage ≥ 25 V
- coaxial lines for monitors

⇒ Group B:
- unshielded lines for DC voltages ≥ 60 V and ≥ 400 V
- unshielded lines for AC voltage ≥ 24 V and ≥ 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.

12.2.4.1 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!

13 Firmware CL-extension with CANopen interface

The firmware version is output in the configuration mode (see chapter 4.1). The actual start-up message appears, following the message of the extension that looks similar to the following:

Ext-Board: CL-Erweiterung(CANopen-IO-DICNET) V0.74 (c)dA SN=4294967295
CAN: ID=1 Baud=500000 LSS-ID=1, LSS-BaudIdx=2

The switch position "DD" (meaning both, S4 and S5 are both in position "D") serves for the Firmware-Update. This position is transmitted to the CL-extension. The CPU of the CL-module goes mute and the CL-extension is set fixed in the boot mode, which means now a Firmware NEEDS to be loaded into the extension. The Firmware-Update can be started via the Firmware-Download-Tool FDT (Resume Download). The LED "Powe/State" is bright red.

14 CANopen

14.1 Description CANopen

This specification is based on the CiA® Draft Standard 301 (DS301).
CANopen supports the Standard CAN-frame with 11-bit Identifier.
It is not required to support the extended frame with 29-bit Identifier.

14.1.1 CANopen V3

Only for used products. Currently we provide CANopen V4.

A CANopen V3 script can be adapted to CANopen V4 with only minor changes.

Syntax
CO_InitChannel (vw_Channel , Direction , vw_len , vw_Obj_Nr , vw_COBId)

Description
From Script rev. 25 on and higher the CAN Firmware allows the definition of user objects and the mapping of up to 16 Rx and 16 Tx PDOs. For Script rev. between 22 and 25 only up to 5 Rx and 5 Tx PDOs can be used.

Predefined Communication
For some applications one Rx and one Tx PDO is sufficient.
It is possible to use CANopen without the definition of communication channels. In this case the data is mapped as follows:

Data width	Direction	Object	Mapping
1..8 byte	Rx	2000	Default Rx-PDO1 (COB-ID 200 + Node ID)
1..8 byte	Tx	2001	Default Tx-PDO1 (COB-ID 180 + Node ID)
9..255 byte	Rx	2000	Data not mapped (could be read by SDO) no Rx-PDO available
9..255	Tx	2001 (Tx-Data) 2002 (Tx-Length)	Data not mapped (write by SDO) Data width in Tx PDO 1 (COB-ID 180 + Node ID)

It is possible to use ReadBus and WriteBus for the data exchange. This standard behavior is no longer active if you call CO_InitChannel at least once.

User-defined communication
This mode is necessary if you want to use more sophisticated CANopen functions. You have to initialize a CANopen channel for every PDO or object.

Syntax
CO_InitChannel (vw_Channel , Direction , vw_len , vw_ObjAddress , vw_COBId)

Use the following values for the parameters:

<table>
<tr><th>Parameter</th><th>Type</th><th>Meaning</th></tr>
<tr><td>vw_Channel</td><td>Word</td><td>
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>0</td><td>Using CAN Layer2, we have no PDO and SDO data access</td></tr>
<tr><td>1..8</td><td>Define PDO 1..8</td></tr>
</table>
</td></tr>
<tr><td>Direction</td><td>-</td><td>RX or Tx depending of the data direction. It is seen from the devices view, this means Rx-data is incoming data.</td></tr>
<tr><td>vw_ByteLen</td><td>Byte</td><td>Length of the object to use. If the length is > 8 byte only the first 8 bytes are used to transmit by the PDO</td></tr>
<tr><td>vw_ObjAddress</td><td>Word</td><td>Allowed values are 0x2000 to 0x5FFF, which is the range of objects to be defined by the user</td></tr>
<tr><td>vw_COB_ID</td><td>Word</td><td>
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>0</td><td>PDO is not active, data is defined to be used by a SDO transfer only.</td></tr>
<tr><td>0x181..0x57F</td><td>Allowed range for normal Rx and Tx - PDO's</td></tr>
<tr><td>0xFFFE</td><td>Is to be used, if the master defines the COB-ID's when the CANopen network is started by the master (no predefined COB-IDs).
The resulting objects used by the device are 0x1800 + (PDO-Nr - 1). Sub-Index 1 of this object contains the COB-ID. After writing a valid value to this object with subindex the requested PDO becomes active. Valid values must be in the range from 0x181 to 0x57F.</td></tr>
<tr><td>0xFFFF</td><td>Is to be used for PDO1 and PDO2. The COB-ID is as defined by the predefined connection set
Tx-PDO1: 0x180 + Node-ID
Rx-PDO1: 0x200 + Node-ID
Tx-PDO2: 0x280 + Node-ID
Rx-PDO2: 0x300 + Node-ID</td></tr>
</table>
</td></tr>
</table>

CAN Layer 2

If you want to use CAN Layer 2, you can set a special Script initialization to access every CAN message without any Filter. Please note that the data format for ReadBus and WriteBus differs from other functionalities in this case. From now on the COB-ID of the message is to be read or sent in the data area's first 2 bytes.

The following examples can be found in the file folder "example" after the installation of the software "Protocol Developer":

- Example CANopen 2 PDOs
- Example CAN Layer 2

With it please take a look at the followlwing Script commands from the "Protocol Developer" as well:

- CO_Read PDO
- CO_WriteEmergency
- CO_WritePDO

14.1.2 CANopen V4

Additional supported functions:
- Heartbeat
- Dynamic mapping
- Onswich message

The following example can be found in the file folder “example“ after the installation of the software “Protocol Developer“ (this example gives a detailed description of the initialization):
- Example_CO_V4.dss

At CANopen V4 the following fieldbus-specific Scripts are supported:
- Init Object Table
- Create Object
- Set PDO Communiction
- Set PDO Mapping
- Write Object
- Read New CANopen Object Data
- Emergency Message

The software does not support default objects, as described at CANopen V3.

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)
3	Service life	10 years	
4	Housing size	23 x 117 x 111 mm (screw-plug-connector included) 23 x 117 x 100 mm (screw-plug connector not included)	W x D x H
5	Installation position	Any	
6	Weight	130 g	
7	Operating temperature	-40 ºC ... +85 ºC	The negative temperatures are only valid for the usual conditions (not condensing)
8	Storage/transport temperature	-40 ºC ... +85 ºC	
9	Atmospheric pressure during operation during transport	 795 hPa ... 1080 hPa 660 hPa ... 1080 hPa	
10	Installation altitude	2000 m 4000 m	Unrestricted Restricted - Ambient temperature ≤ 40ºC
11	Relative humidity	Max. 80 %	No condensation, no corrosive atmosphere
12	External power supply	10..33 V DC	Standard power supply unit to DIN 19240
13	Current consumption at 24 VDC	Typ. 120 mA max 150 mA	At 10.8V. typ. 350 mA
14	Reverse voltage protection	Yes	But does not function!
15	Short-circuit protection	Yes	
16	Overload protection	Poly-switch	Thermal fuse
17	Undervoltage detection (USP)	≤ 9 V DC	
18	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	CANopen RS485	RS232-C RS232-C	RS485/RS422 RS485/RS422
1	Standard	CiA® DS 102	DIN 66020	EIA Standard
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
3	Transmission method	Master / slave	Master / slave	Master / slave
4	Number of users : - Transmitters - Receivers	 32 32	 1 1	 32 32
5	Cable length: - Maximum - Baud rate-dependent	 1300 m 50 kBd → 1300 m 100 kBd → 640 m 200 kBd → 310 m 500 kBd → 112 m 1 MBd → 40 m	 15 m no	 1200 m <93.75 kBd → 1200 m 312, kBd → 500 m 625 kBd → 250 m
6	Bus topology	Line	Point-to-point	Line
7	Data rate: - Maximum - Standard values	 1Mbit/s 125 kB 250 kB 500 kB 1 MB	 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
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
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
10	Transmit range (SPACE): - Voltage level - Logic level	 - 0.5 ... + 0.05 V 0	 + 3 ... + 15 V 0	 - 0.2 ... + 0.2 V 0
11	Transmit pause (MARK): - Voltage level - Logic level	 + 1.5 ... +3 V 1	 - 3 ... –15 V 1	 + 1.5 ... +5 V 1

Table: Technical data of the interfaces and ports on the module

16 Commissioning guide

16.1 Note

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

16.2 Components

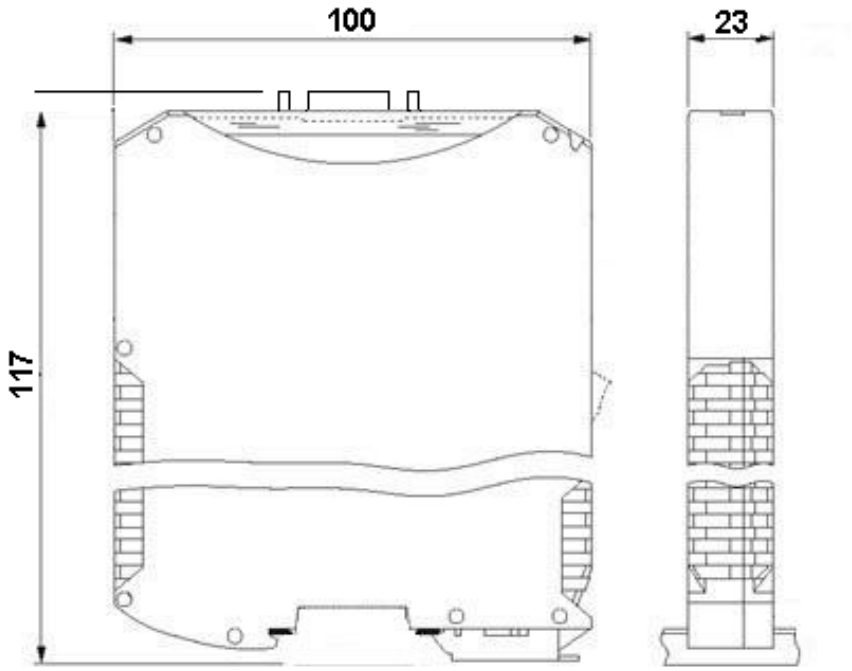
You will require the following components to commission the UNIGATE®:

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

16.3 Installation

The UNIGATE® CM - CO 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 - CANopen



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 CANopen address and baud rate

Set the CANopen-Node-ID and the baud rate at the fieldbus end of the module on the DIP-switch (see chapter 10.4.5).

Attention:
The CANopen address set must correspond to the planned address!
All users in the CANopen have to use the same Baud rate!
These values are read in only on power-up of the gateway!

16.7 CANopen connection

Connect the device to the CANopen at the interface labelled "CANopen".

16.8 Additional CANopen interface connection (application side)

Connect the device to the CANopen at the interface X6.

16.9 Setting the address and baudrate (additional CANopen interface, application side)

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!

16.10 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 EDS file was not supplied with your project planning tool, a sample file can be found on the Internet (www.deutschmann.de).

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 or directly in our Wiki on www.wiki.deutschmann.de

If your questions are still unanswered please contact us directly.

Please note down the following information before calling:
- Device designation
- Serial number (S/N)
- Article number
- Error number and error description

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

Technical Support hours are as follows:
Monday to Thursday from 8 am to midday and from 1 pm to 4 pm, Friday from 8 am to midday (CET).

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

Central office and sales department	+49 6434 9433-0
Technical Support	+49 6434 9433-33
Fax sales department	+49 6434 9433-40
Fax Technical Support	+49 6434 9433-44
E-mail Technical Support	support@deutschmann.de

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 (±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® IC for the respective product)
DN	=	DeviceNet
EC	=	EtherCAT
EI	=	Ethernet/IP
FE	=	Ethernet 10/100 MBit
FEX	=	Ethernet 10/100 MBit-version X (see comparison table UNIGATE® IC for the respective product)
IB	=	Interbus
IBL	=	Interbus
LN62	=	LONWorks62
LN512	=	LONWorks512
ModTCP	=	ModbusTCP
MPI	=	Siemens MPI®
PL	=	Powerlink

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

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