BERGHOF
AUTOMATIONSTECHNIK
Unternehmen der ZUNDEL Holding

CEDISP29-16/8

Dialog Controller

V.1.11

User Handbook

Automation System

CANtrol®

Disclaimer

The content of this publication was checked for compliance with the hardware and software described. However, discrepancies may arise, therefore no liability is assumed regarding complete compliance. The information in this document will be checked regularly and all necessary corrections will be included in subsequent editions. Suggestions for improvements are always welcome.

Subject to technical changes.

Trademark

CANtrol® // is a registered trademark of BERGHOFAutomationstechnik GmbH

General Information on this Manual

Content:
This manual describes the Dialog Controller CEDISP29-16/8 and its modifications. The product-related information contained herein was up to date at the time of publication of this manual.

Completeness:
This manual is complete only in conjunction with the user manual entitled

'Introduction
to CANtrol Automation System'

and the product-related hardware or software user manuals required for the particular application.

Standards:
The CANtrol automation system, its components and its use are based on International Standard IEC 61131 Parts 1 to 4 (EN 61131 Parts 1 to 3 and Supplementary Sheet 1).
Supplementary Sheet 1 of EN 61131 (IEC 61131-4) entitled 'User Guidelines' is of particular importance for the user.

Order numbers:
Please see the relevant product overview in the 'Introduction to CANtrol Automation System' manual for a list of available products and their order numbers.

Ident. No.: 2806220

You can reach us at:

BERGHOFAutomationstechnik GmbH
Harretstr. 1
D-72800 Eningen / Germany
Phone: +49 7121 / 894-0
Telefax: +49 7121 / 894-100
e-mail: info@berghof-automation.de
www.berghof-automation.de

BERGHOFAutomationstechnik GmbH works in accordance with DIN EN ISO 9001:2000

Update

Version	Date	Subject
1.0	27.01.04	First version
1.1	18.05.04	Valid from HW-Version 0002 CAN TERM included. Display: A new setup menue, default value via button F2
1.11	28.07.05	New warnings in chapter 'Replacing the battery'

blank page

Contents

1. General Instructions

1.1. Hazard Categories and Indications

The indications described below are used in connection with safety instructions you will need to observe for your own personal safety and the avoidance of damage to property.

These instructions are emphasised by bordering and/or shading and a bold-printed indication, their meaning being as follows:

DANGER !	means that death, severe physical injury or substantial damage to property will occur on failure to take the appropriate precautions.

Warning !	means that death, severe physical injury or substantial damage to property may occur on failure to take the appropriate precautions.

Caution	means that minor physical injury or damage to property may occur on failure to take the appropriate precautions.

Note:
provides important information on the product or refers to a section of the documentation which is to be particularly noted.

1.2. Qualified users

Qualified users within the meaning of the safety instructions in this documentation are trained specialists who are authorised to commission, earth and mark equipment, systems and circuits in accordance with safety engineering standards and who as project planners and designers are familiar with the safety concepts of automation engineering.

1.3. Use as Prescribed

This is a modular automation system based on the CANbus, intended for industrial control applications within the medium to high performance range.

The automation system is designed for use within Overvoltage Category I (IEC 364-4-443) for the controlling and regulating of machinery and industrial processes in low-voltage installations in which the rated supply voltage does not exceed 1,000 VAC (50/60 Hz) or 1,500 VDC.

Qualified project planning and design, proper transport, storage, installation, use and careful maintenance are essential to the flawless and safe operation of the automation system.

The automation system may only be used within the scope of the data and applications specified in the present documentation and associated user manuals.

The automation system is to be used only as follows:

- as prescribed,
- in technically flawless condition,
- without arbitrary or unauthorised changes and
- exclusively by qualified users

The regulations of the German professional and trade associations, the German technical supervisory board (TÜV), the VDE (Association of German electricians) or other corresponding national bodies are to be observed.

Safety-oriented (fail-safe) systems

Particular measures are required in connection with the use of SPC in safety-oriented systems. If an SPC is to be used in a safety-oriented system, the user ought to seek the full advice of the SPC manufacturer in addition to observing any standards or guidelines on safety installations which may be available.

Warning ! As with any electronic control system, the failure of particular components may result in uncontrolled and/or unpredictable operation. All types of failure and the associated fuse systems are to be taken into account at system level.
The advice of the SPC manufacturer should be sought if necessary

2. Dialog Controller CEDISP29-16/8

2.1. Overview

Order number

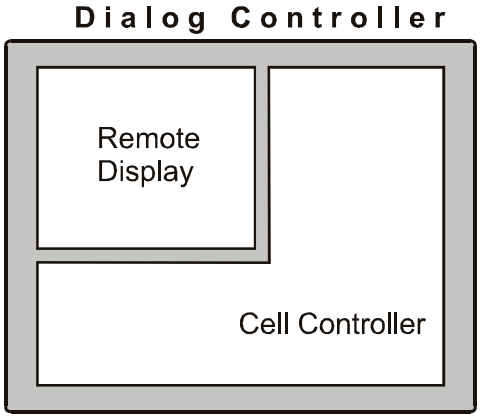
The order/item No. required for acquiring a replacement is to be found on the nameplate of the module.

Structur

The Dialog Controller integrates to 2 different devices in one casing. Essentially the 2 devices are combinated via the CAN bus.

The Dialog Controller comprises an

- real-time-capable Cell Controller and an
- Remote Display with Operator controls.



2VF100134DG00.cdr

Cell Controller

The Cell Controller degreed digital I/O, analog input and interfaces for Ethernet, CAN bus and serial communication. One of 2 CAN interfaces can also used for data communication in contact-conductor-fed systems
The module is prepared for expansion via E-bus. The Cell Controller is programmable to IEC 61131-3 standard.

Features Cell Controller

- MC 68360 CPU / 33 MHz
- 702 kB program and data memory (RAM)
- 1024 kB program memory (flash)
- 1 Ethernet 10 MBit/s
- 2 CAN interfaces ISO 11898
- 1 RS232 interface
- 1 RS485 interface
- 1 serial data interface RS232 for configuration / application
- 16 digital inputs
- 8 digital outputs
- 2 analog inputs

Remote Display

Remote Display as CANopen Slave module with fixed functionality.
The display, the LEDs and the keyboard are controlled exclusively by the CAN bus. Display texts and menu structures can be freely configured. A CP1131 library supports this function. The base parameters are set on the module itself by means of a configuration menu.

Features Remote Display

- CANopen device
- Graphical LCD display, 240 x 64 pixels with back bias
- 29 membrane keys separated into blocks for special keys, numeric keys and function keys, 9 bi-coloured LEDs, 5 single-coloured LEDs
- Real-time clock
- Customised versions possible

Display

Graphical, backlit LCD display with a resolution of 240 x 64 pixels and adjustable contrast. The character sizes that can be displayed are 8 x 8 or 16 x 16 pixels. The characters can be displayed in reverse colour and underlined. In graphics mode each pixel can be activated individually.

Membrane layout

29 membrane keys and 9 bi-coloured and 5 single-coloured LEDs, arranged in blocks:

Topmost block:	4 status display LEDs
Middle block :	5 function keys
Lower block, left:	15 keys, numeric keys block with 1 LED
Lower block, right:	9 keys, block of special keys each having 1 LED (bi-coloured).

Customised version

The Dialog Controller is also available in customised versions.
Individualised options include logo, special colours, device name and key labelling as well as special arrangement of keys and LEDs.
The maximum number of keys possible is 35 and the maximum number of LEDs is 36.

Material supplied

The material supplied comprises:

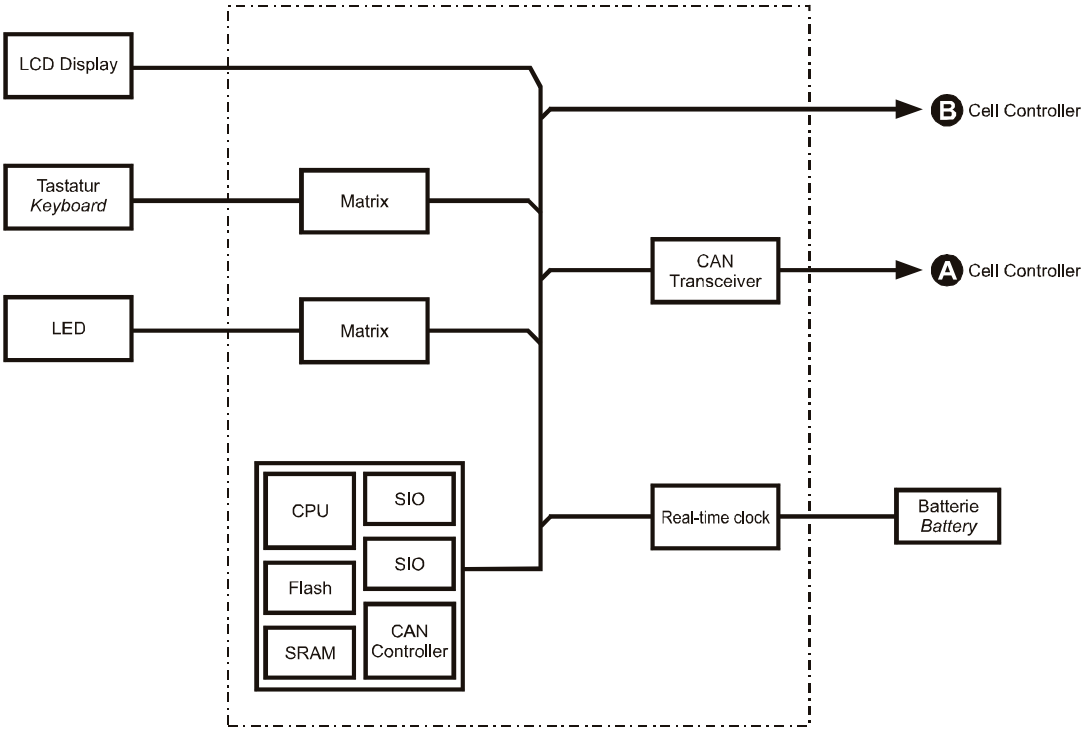
- Dialog Controller with assembly materials

2.2. Technical Data

Dialog Controller	**CEDISP29 16/8-1131**
Item No.	22 0900 000
Real-time clock	yes, by the Display
Development environment	CP1131
CPU	MC 68360 / 33 MHz
Expansion	E-bus module with 7-segment display and infrared remote control lockable (not supplied)
User memories	
Program memory (flash)	1024kB (contains pre-initialised variables, route tables, etc.)
Program and data memory (RAM)	702 kB
Dimensions	
Dimensions B x H x D [mm]	204 x 175 x 55 (without connector)
Port	as per dimensioned sketch
Weight	approx. 1500 g
Mounting	with mounting frame for front mounting
Operating temperature range	5° C to 50° C (no moisture condensation)
EMC, safety class, insulation test, system of protection	
Emitted interference	EN 50081-2, industrial range
Noise immunity	EN 50082-2, industrial range
Safety class	III
Insulation strength	EN 61131-2; DC 500 V testing voltage
System of protection	IP65 front / IP20 on the back
Supply voltage, current consumption	
Power supply module electronics (supply voltage)	SELV DC +24 V, < 1 A (EN 61131-2) / according to display brightness
Power supply - digital I/Os	DC +24 V (EN 61131-2) distributed into 3 groups
Power consumption	at U_e = DC +24 V running at no load max. 1 A, fuse protection according to load on I/Os, max. 8 A
Power-supply reverse voltage protection	yes
Electrical isolation	yes, between CAN bus and digital I/O
Digital inputs/outputs (DIO)	
Number of inputs	16
Number of outputs	8
Output current	0,5 A
Short-circuit protection	yes
Connection method	vertical three-wire front wiring with push-on terminal strips for screw, spring or crimp connection
Analog inputs	
Number of inputs	2 inputs (X7 / X8)
Input characteristics	Either as 4..20 mA with +24 V / 50 mA sensor supply or as unipolar voltage input 0..+10 V applicable
Resolution	10 bit
Connection method	vertical front wiring with push-on terminal strips for screw, spring or crimp connection

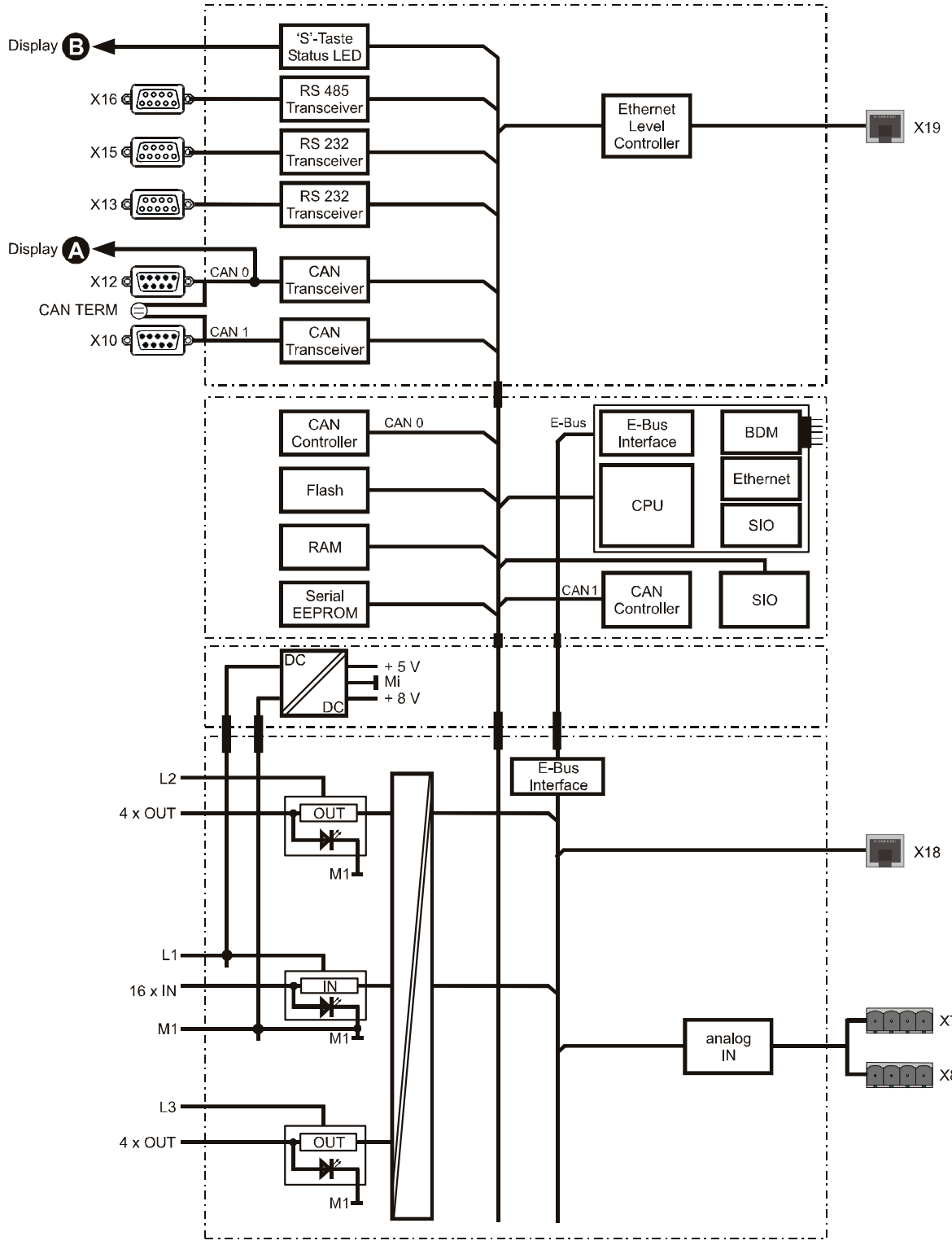
Dialog Controller	CEDISP29 16/8-1131
Serial data interfaces	
Number and type of interfaces	1 RS232 for programming / application (X13) 1 serial interface RS232 (X15) 1 serial interface RS485 (X16) (current-loop configuration is not an option)
Ethernet interface	
Number and type of interfaces	1 Ethernet interface, 10 MBit/s (X19)
Protocols	TCP/IP and UDP/IP
Connection method	RJ45
CAN interfaces	
Number and type of interfaces	2 standard CAN ISO11898 (X10 / X12)
CAN bus terminator	Each of both CAN bus interfaces (ISO11898) are equipped with a switchable CAN bus termination.
Display connection (internal)	
CAN Bus	by CAN channel 0 as CANopen slave (X12)
Display elements	
Display	LCD, 240 x 64 pixels, backlit
Character size	8 x 8, 16 x 16 pixels
Character set	Standard ASCII, other character sets can be loaded
LED’s display application	5 single-coloured LEDs and 9 bi-coloured LEDs
LED's / status	5 status LED’s; therefrom 2 for CAN status (CAN0 / CAN1) 1 status -LED per input / output
Operator controls	
Number of keys	29
Key fields	Numeric and cursor keys block, special keys, function keys
S-key	Yes, on the back (module reset)
Software	
Control application	to IEC 61131-3
Display application	via library (control- and display application are on one Controller)
Interface tools	via Ethernet, CAN bus or RS232 interface

2.3. Block Diagram Remote Display



2VF100123DG00.cdr

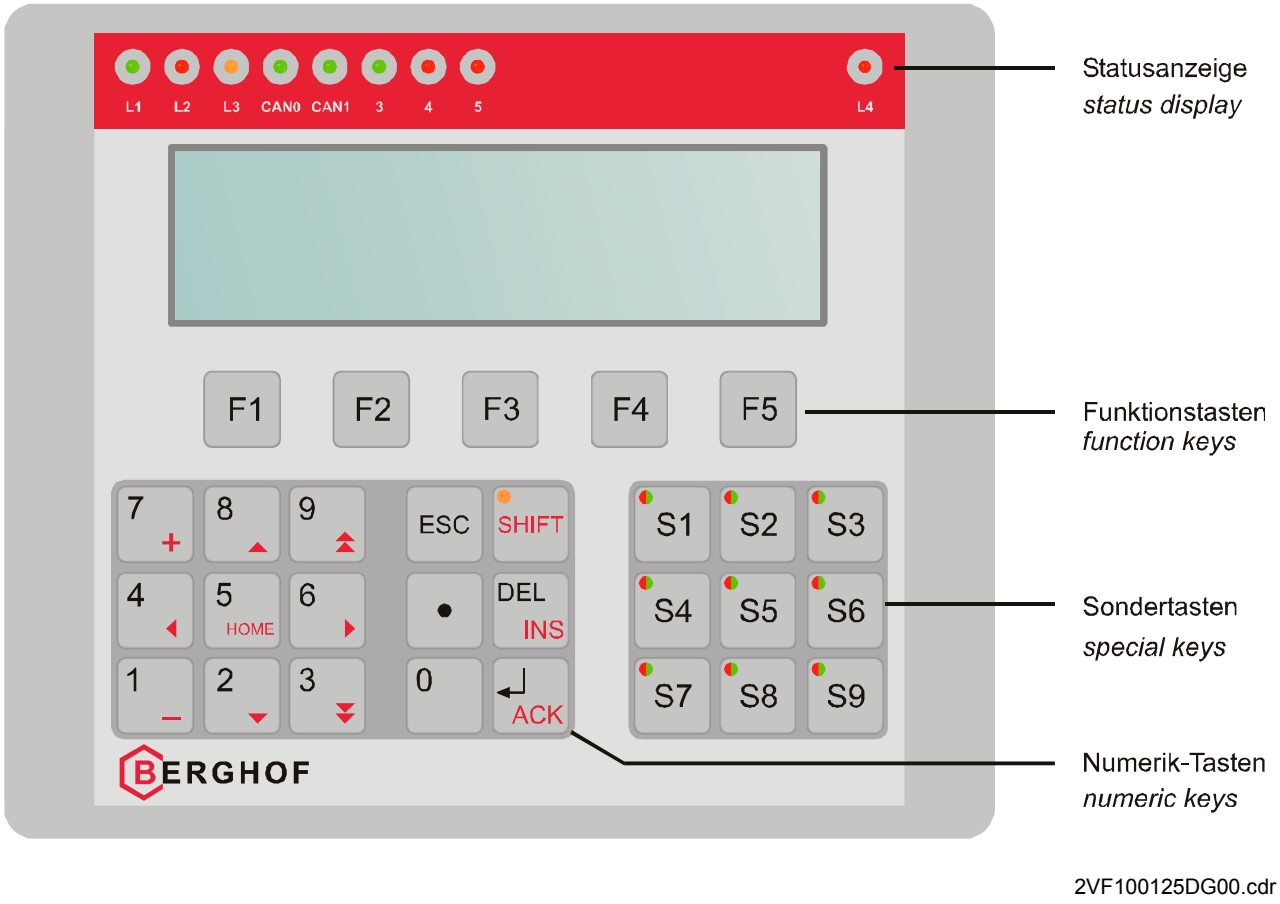
2.4. Block Diagram Cell Controller



2VF100144DG01.cdr

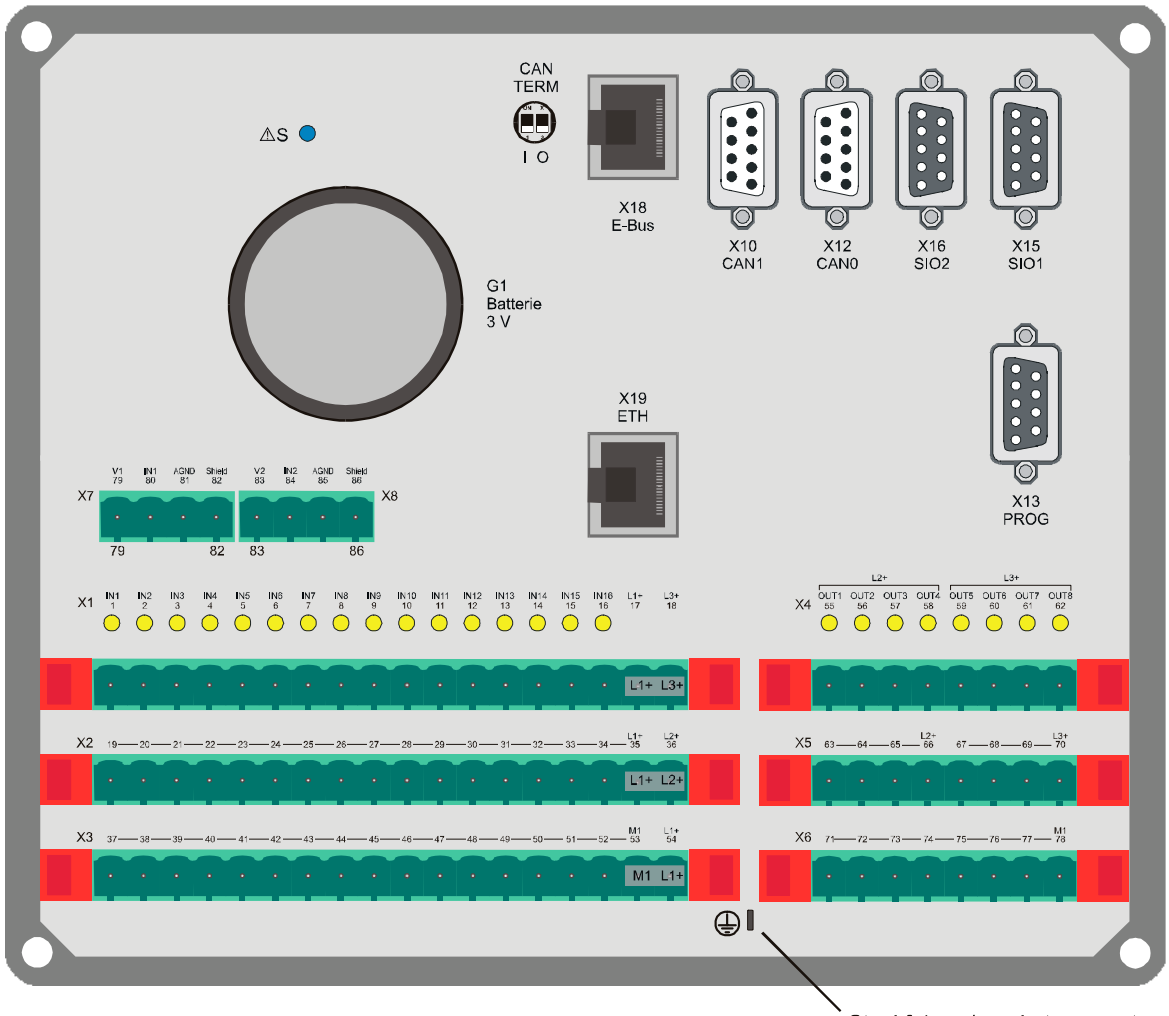
2.5. Device Views and Connections

2.5.1. Front view



2VF100125DG00.cdr

2.5.2. Rear view



2.5.3. Control configuration of digital inputs/outputs

The digital inputs/outputs are located in DIO module [0] and DIO module [1] of the control configuration in the programming tool CP1131.

Control configuration CP1131 - DIO module [0]

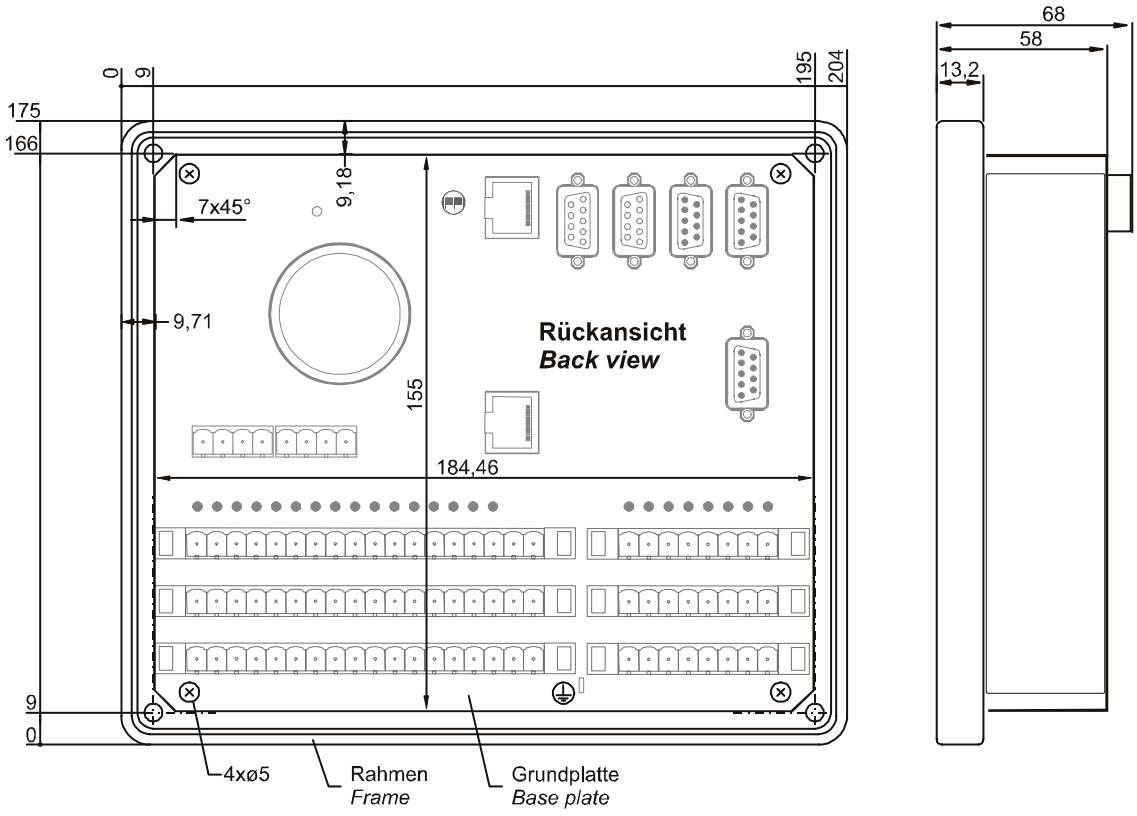
Type	Control configuration CP1131	
Signal	**OUT**	**IN**
OUT 0	QX0.0	-
OUT 1	QX0.1	-
OUT 2	QX0.2	-
OUT 3	QX0.3	-
OUT 4	QX0.4	-
OUT 5	QX0.5	-
OUT 6	QX0.6	-
OUT 7	QX0.7	-

Control configuration CP1131 - DIO module [1]

Type	Control configuration CP1131	
Signal	**OUT**	**IN**
IN 0	-	IX1.0
IN 1	-	IX1.1
IN 2	-	IX1.2
IN 3	-	IX1.3
IN 4	-	IX1.4
IN 5	-	IX1.5
IN 6	-	IX1.6
IN 7	-	IX1.7
IN 8	-	IX1.8
IN 9	-	IX1.9
IN 10	-	IX1.10
IN 11	-	IX1.11
IN 12	-	IX1.12
IN 13	-	IX1.13
IN 14	-	IX1.14
IN 15	-	IX1.15

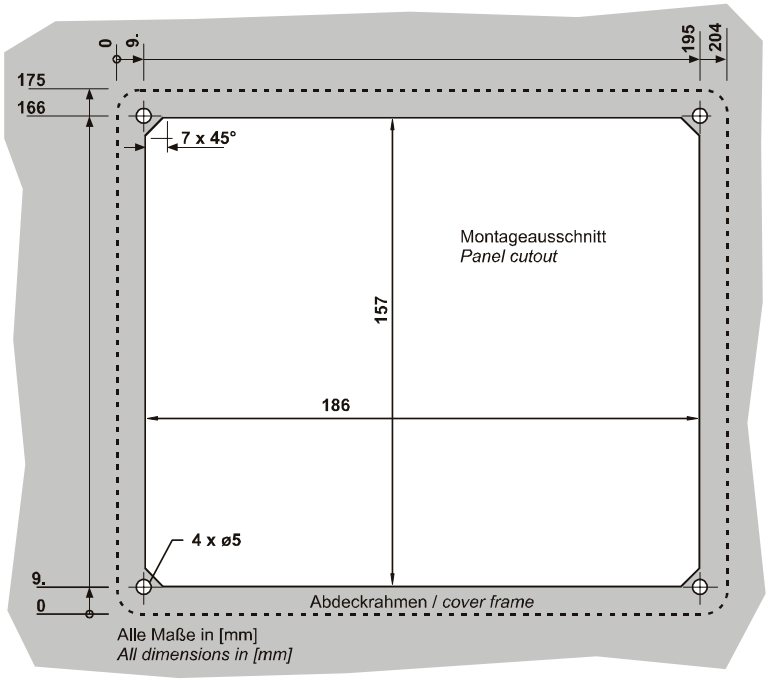
2.6. Mounting

2.6.1. Housing dimensions



2VF100146DG01.cdr

2.6.2. Mounting cut-out



2VF100128DG00.cdr

2.7. Component Operation

Warning ! Do not insert, connect, detach or touch connections when in operation!
Destruction or malfunctioning may otherwise occur.
Disconnect all power supply before working on modules.

2.7.1. Commissioning

Re-examine all connections for correct wiring and polarity before applying the supply voltage; when this is done, switch on the supply voltage. The Dialog Controller has a common connector for the supply unit of the Remote Display and the Cell Controller.

After power up the Cell Controller re-activates the state which was active before power down.

Boot UP

The Remote Display executes a boot up process and goes directly into CANopen operational state.
It can then be operated straight away by means of CANopen SDO/PDO accesses.
The following text appears on the display:

Display / 240x64 DD.MM.YYYY
(DD.MM.YYYY indicates the creation date of the system software)

When the supply voltage is disconnected, the display turns off and all LEDs are set to 0 (low). After restarting the LEDs remain in this state.

CANopen

The Remote Display is accessed in accordance with the CANopen communication profile defined in the CiA Draft Standard DS301.
The device profile is completely application-specific.

PDO/SDO telegrams

Depending on function, the Remote Display is controlled by means of SDO (*Service Data Object*) or PDO (*Process Data Object*) telegrams.
Texts or pixel graphics for the display are transmitted solely via SDO. The LEDs are switched via SDO, also.
All keyboard operation is transmitted via PDO on an event-driven basis.
The Remote Display supports one PDO telegram in the transmit direction and one SDO channel.

Note:
Please refer to the section '*Object Dictionary' (ff)* for more information.

2.7.2. Functions Selection, Displays, Diagnostics

Warning ! Do not touch 'S' button during normal operation.
Program sequence could otherwise be put into an undefined state.
Risk of uncontrolled system and machine states!
Put system/machine into a safe initial state ('maintenance' mode for example) before actuating the 'S' button.

'S' button

The 'S' button used to switch between modes and to re-start the module.
The function of the 'S' button is software-dependent.
Each actuation resets the Remote Display.

I/O status

Each input and output has its own yellow I/O-status LED which indicates the logic state of the input or output in question.

Operating status

Module mode and other functions are indicated by 5 operating-status LEDs.
Error messages are also displayed by these status LEDs.

I/O status

LED status	Logical status
input LED yellow ON	1 (HIGH, activated)
input LED yellow OFF	0 (LOW)
output LED yellow ON	1 (HIGH, activated)
output LED yellow OFF	0 (LOW)

Operating status

LED		Logical status
	CAN 0	ON = CAN 0 communication active
	CAN 1	EIN = CAN 1 communication active
3	CAN status 3 (green)	see Software Manual
4	CAN status 4 (red)	see Software Manual
5	CAN status 5 (red)	see Software Manual

CAN TERM

Integrating the module at the beginning or end of a CAN bus network (ISO11898), is supported by a DIP switch for the CAN bus termination. Each of both CAN bus interfaces (CAN Channel 0, CAN Channel 1) are equipped with this feature.

Switch status

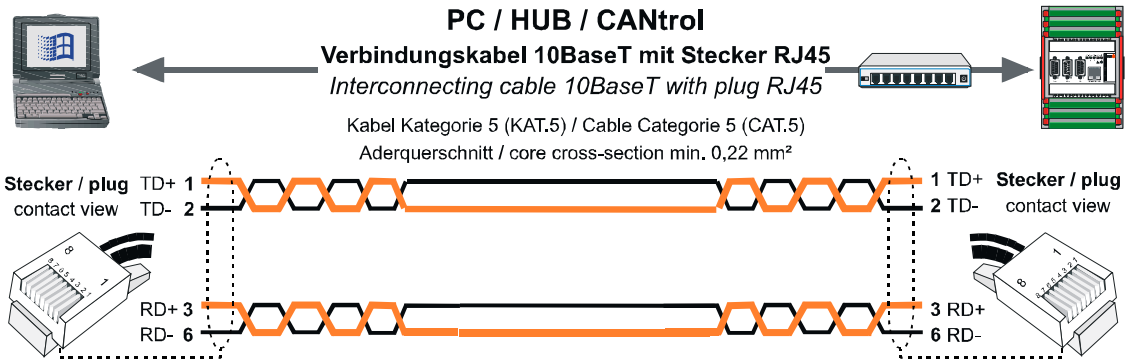
ON	CAN Bus Termination active
OFF	CAN Bus Termination not active

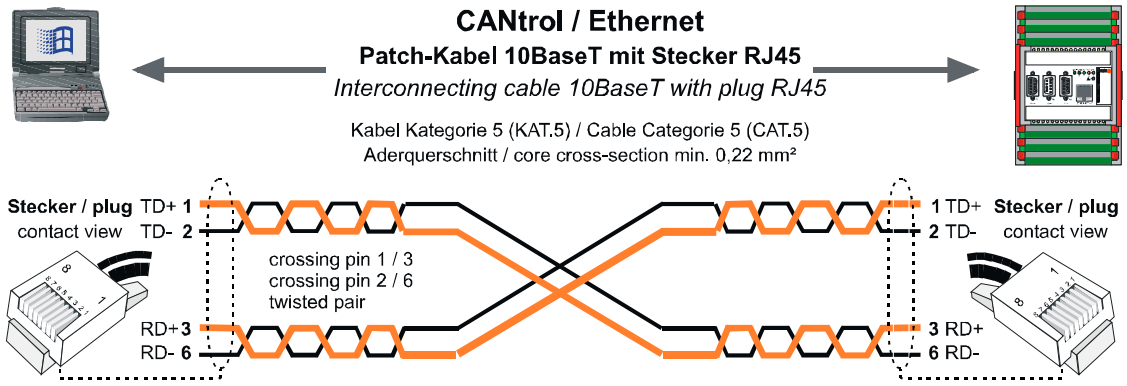
blank page

3. Communication Interfaces CEDISP29-16/8

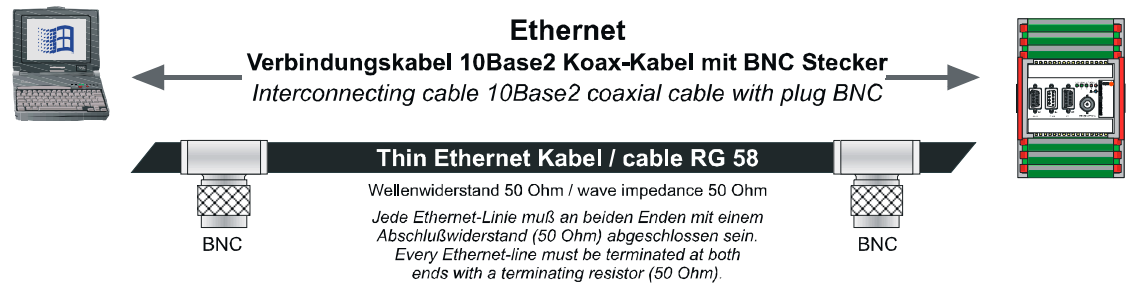
3.1. Ethernet Interface

The Ethernet interface (X19 / ETH) is on the back of the module.
The module is supplied with a RJ45 connector.





Pin	Signal	Description
1	**TD+**	
2	**TD-**	
3	**RD+**	
4	NC	reserved
5	NC	reserved
6	**RD-**	
7	NC	reserved
8	NC	reserved



Um den Anforderungen in industrieller Umgebung gerecht zu werden, sollten nur gesicherte Protokolle gefahren werden.
For correct data transfer in industrial atmosphere only safty protocolls should be used.

2VF100006DG00.cdr

3.2. Serial Interfaces

The module has 3 serial communication interfaces.
1 x RS232 (X13 / PROG)
1 x RS232 (X15)
1 x RS485 (X16)

Note:
The RS485 interface is unsuitable to operate absolute encoder with the CP1131 library CM485LIB.LIB .

SIO X13 (RS232 / PROG) assignment

Pin	Signal	Description
1	reserved	Do not connect
2	RXD	RS232 / received data
3	TXD	RS232 / transmitted data
4	reserved	Do not connect
5	GND	Signal ground
6	CAN1_H	CAN_H bus line (dominant high)
7	reserved	Do not connect
8	reserved	Do not connect
9	CAN1_L	CAN_L bus line (dominant low)

SIO2 X15 (RS232) assignment

Pin	Signal	Description
1	reserved	Do not connect
2	RXD	received data
3	TXD	transmitted data
4	reserved	Do not connect
5	GND	Signal ground
6	reserved	Do not connect
7	RTS	request to send
8	CTS	clear to send
9	reserved	Do not connect

SIO1 X16 (RS485) assignment

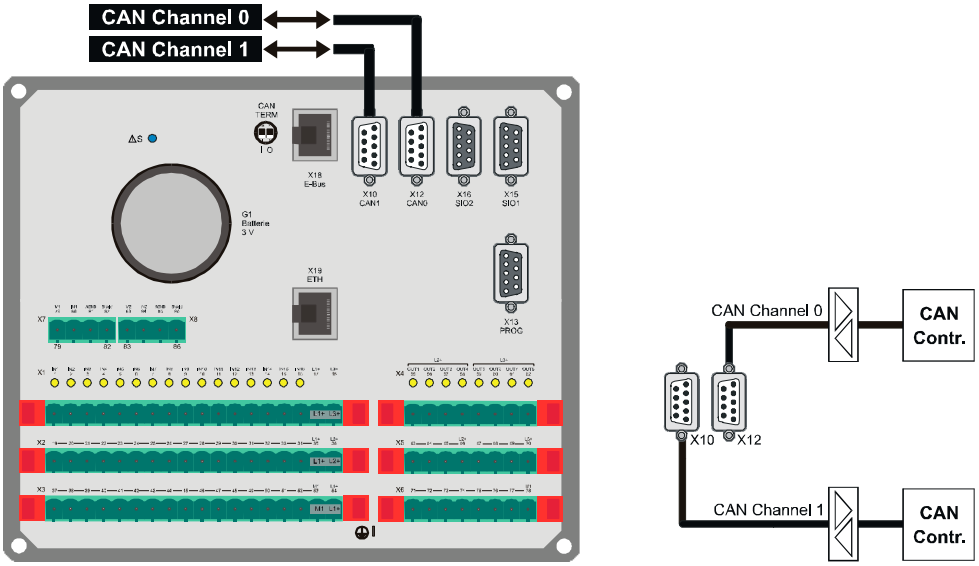
Pin	Signal	Description
1	RTXD-	RS485 received / transmitted data low
2	reserved	Do not connect
3	reserved	Do not connect
4	RTXD+	RS485 received / transmitted data high
5	GND	Signal ground
6	reserved	Do not connect
7	reserved	Do not connect
8	reserved	Do not connect
9	reserved	Do not connect

3.3. CAN Interfaces

The CPU module is equipped with 2 CAN interfaces, named with
CAN Channel 0 and CAN Channel 1.
CAN Channel 0 and CAN Channel 1 are standard interfaces to ISO11898.

For both standard interfaces is a switchable CAN TERM available. Either of the two CAN interfaces can be used to program and configure the module. Either of the two CAN interfaces can be used to program and configure the module.

Note:
The Remote Display is internal connected via CAN channel 0



2VF100147DG01.cdr

CAN channel 0

Terminal X12 on the back; ISO11898

CAN Channel 0 operates as application-specific communication interface that can be used for the connection of peripheral equipment, e.g. by CANopen protocol. The baud rate reaches a maximum of 1 MBit and can be adjusted by software.

CAN channel 1

Terminal X10 on the back; ISO11898
The baud rate reaches a maximum of 125 kBit and can be adjusted by software.
CAN Channel 1 operates as application-specific communication interface that can be used for the connection of peripheral equipment, e.g. by CANopen protocol.

blank page

4. Digital Inputs/Outputs 16/8-0,5

4.1. Grouping of Inputs/Outputs

The grouping facility permits formation of groups, separate power circuits, emergency off circuits, etc. as and when required.
Inputs/outputs can be supplied in groups as
• 1 input group and
• 2 output / input groups.

The **modular electronic circuit** for C modules is supplied together with input group 1 (Group 1) over connection terminals L1+ and M1.
The modular electronic circuit must be supplied with power in **any** cases, otherwise the modules will be inoperable. The supply of the modular electronic circuit must be provided directly (unswitched) from the supply unit.

Inputs

Inputs (sensors) must be supplied directly from the supply unit.
Do not conduct the sensor supply through switched circuits.

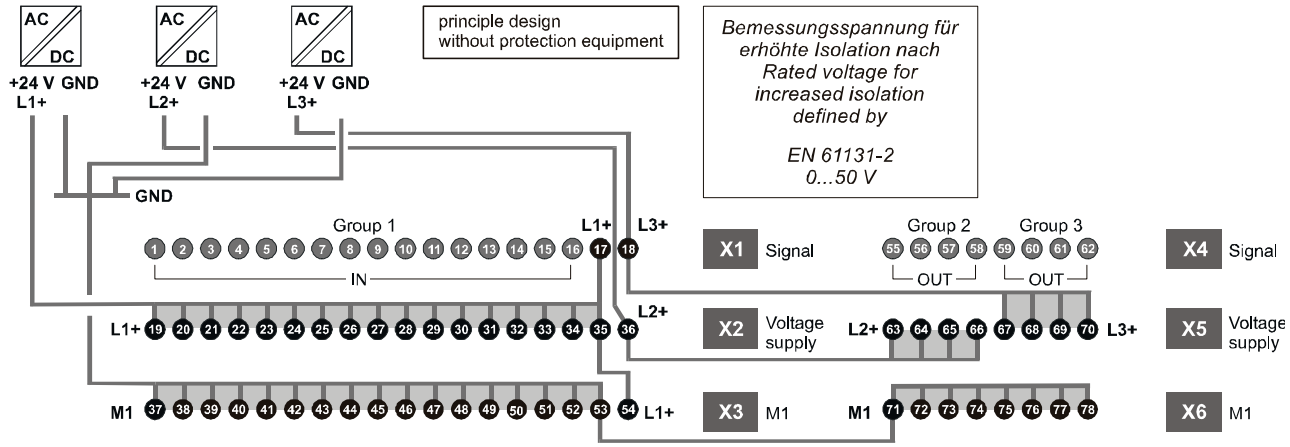
Outputs

Output groups may be supplied through upstream switch elements (emergency off, manual switches, etc.).

Warning !

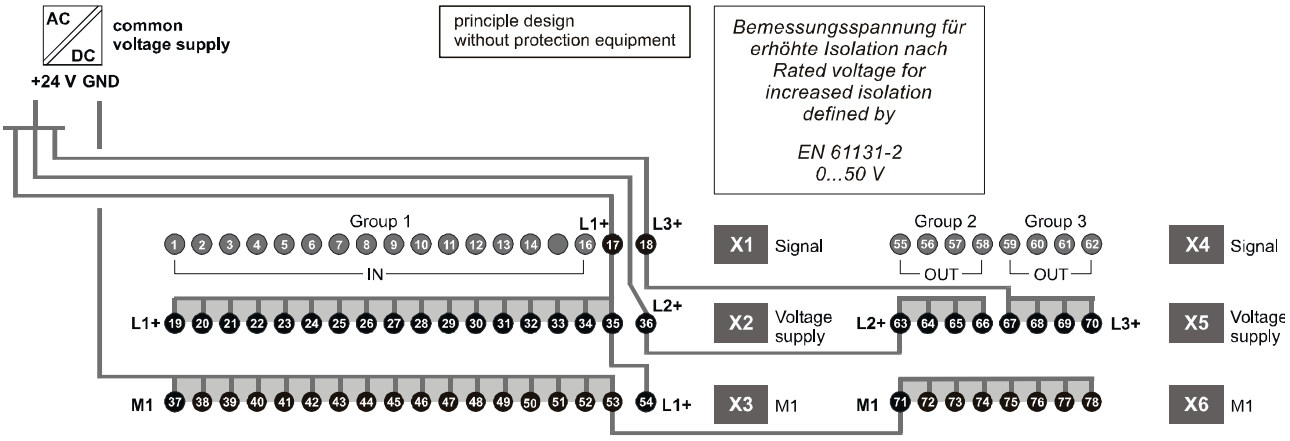
Feedback could destroy the module and/or the sensors!
Otherwise, when group power supply is disconnected, connected sensors could produce a feedback over the output transistors.
Always make sure the sensors are each supplied from the same power source as the module's associated I/O group.

4.1.1. Schematic Diagram of Input/Output Grouping



2VF100131DG00.cdr

4.1.2. Without Grouping



2VF100130DG00.cdr

4.2. Digital Inputs, high side switching

The digital inputs are high side switching type 1 inputs for 3-conductor sensors. They are designed for input voltages of 24 V nominal. The inputs are transmitted cyclically to the CPU. An open input is interpreted as static 0 (LOW).

Pulse recognition and interference suppression

Inputs are read cyclically. Pulses < 100 µs are hardware suppressed. The sampling interval can be parameterised by software. The shortest possible sampling interval is 250 µs.

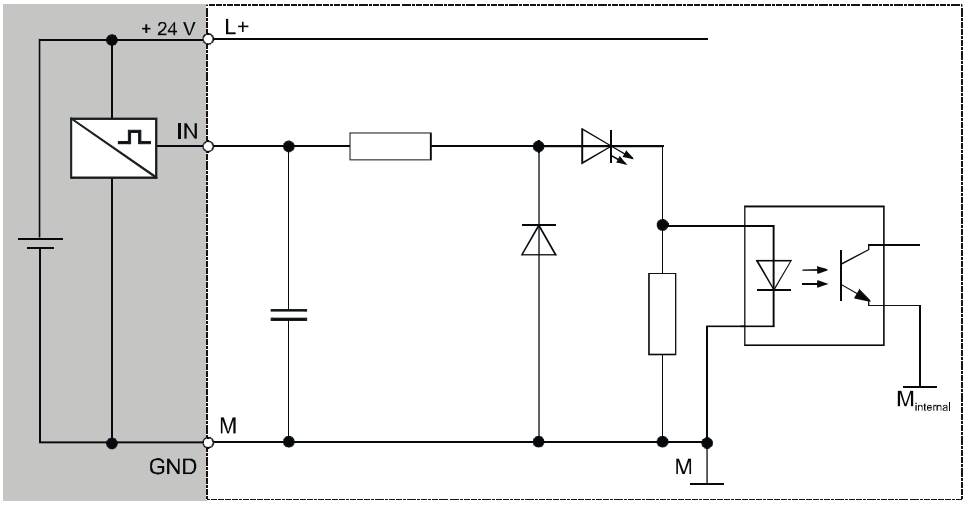
If pulses are to be detected reliably they must be longer than the sampling interval stipulated by software.
Multiple sampling can be programmed in order to suppress spurious pulses.
Sampling interval and multiple sampling (filtering) can be activated in groups of 32 inputs each.

Note:
This function is available only for C applications at present. Using IEC 61131-3 the sampling rate is pre-set.

Operating status

The status of each input is indicated by a yellow operating status LED on the front panel of the module. The LEDs are spatially assigned to the supply terminals.
An LED lights when its associated input is activated (logical 1 / HIGH).

4.2.1. Block diagram of input

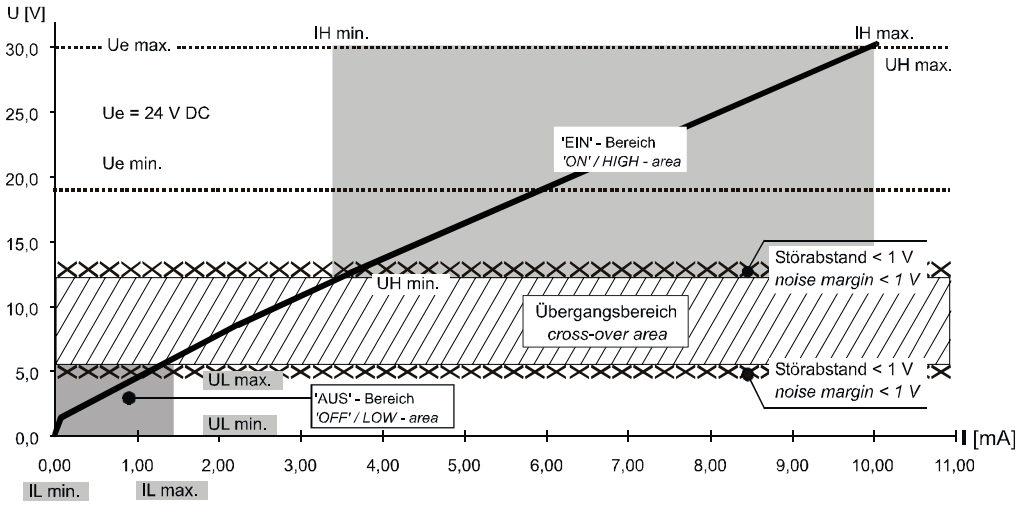


2VF100009DG01.cdr

4.2.2. Digital Inputs Data

Module data	
Number of inputs	16
Line lengths: in switchgear cabinet dedicated l.v. wiring	 Allow for voltage drop when choosing conductor cross-section, otherwise no restrictions in practice. Observe all relevant local regulations and the requirements of EN 61131-3. Please consult manufacturer regarding lightning hazard
Rated load voltage L+ Reverse voltage protection	24 VDC (SELV) yes
Electrical isolation	yes (optical isolator) in groups
Status display	yes, yellow LED for each input
Alarms	definable according to software
Input delay	parameterisable by software
Input capacitance	< 10 nF

Digital-input operating areas



Eingangsspannung (DC) der externen Stromversorgung
Input voltage (DC) of extern power supply

Ue	24 V	Bemessungsspannung / *rated voltage*
Ue max.	30 V	oberer Grenzwert / upper *limit*
Ue min.	19,2 V	unterer Grenzwert / *lower limit*

Grenzwerte für '1' Signal für die 'EIN'-Bedingung
Limit for '1' signal for the 'ON'-condition

$UH_{max.}$	30,0 V	obere Spannungsgrenze / upper *voltage limit*
$IH_{max.}$	10,0 mA	obere Stromgrenze / upper *current limit*
$UH_{min.}$	13,5 V	untere Spannungsgrenze / *lower voltage limit*
$IH_{min.}$	3,5 mA	untere Stromgrenze / *lower current limit*

Grenzwerte für '0' Signal für die 'AUS'-Bedingung
Limit for '0' signal of the 'OUT'-condition

$UL_{max.}$	5,5 V	obere Spannungsgrenze / upper *voltage limit*
$IL_{max.}$	1,5 mA	obere Stromgrenze / upper *current limit*
$UL_{min.}$	0 V	untere Spannungsgrenze / *lower voltage limit*
$IL_{min.}$	0 mA	untere Stromgrenze / *lower current limit*

2VF100010DG00.cdr

4.3. Digital Outputs, high side switching

Warning !	The module can be destroyed by overvoltages > 32 V and / or feedback. Risk of **fire!**

Outputs

The outputs are of high side switching 24 volt type (two-conductor). Maximum output current per output is 500 mA. The outputs have a common earth (GND) when operating in groups. Power is supplied separately from the supply for the modular electronic circuit (see 'Connection Assignment').
The outputs switch automatically to '0' (LOW) if there is no available data link to the CPU or if the module's internal supply is insufficient.

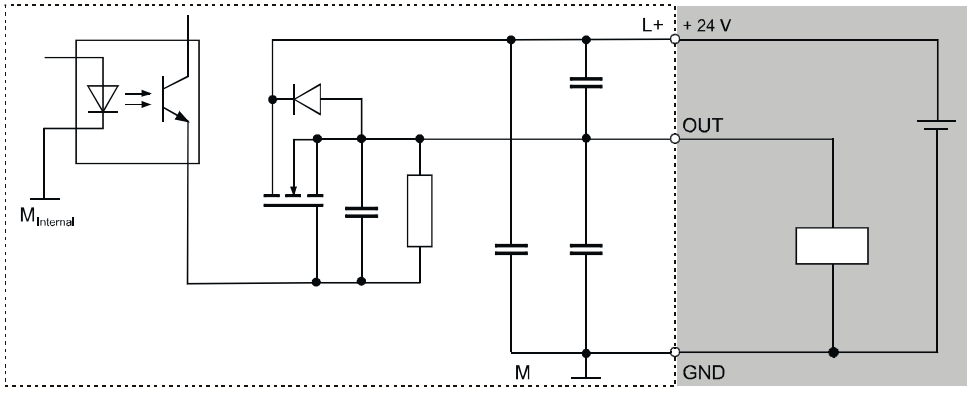
Protected output

All outputs are protected by an incorporated current-limiting circuit and a thermal overload protection circuit. If overloaded, the affected output switches off.
The output can be re-activated by program on elimination of the overload and thermal cooling. A high-speed de-excitation feature having a terminal voltage of 50 V, related to L+, protects all outputs against induced voltage peaks under inductive loads.
The overload protection of non-involved outputs may also respond prematurely if feedback or high-speed de-excitation give rise to thermal loads.

Operating status

The status of each output is indicated by a yellow operating status LED on the front panel of the module. The LEDs are spatially assigned to the supply terminals. A LED lights when its associated output is activated, logical '1' (HIGH).

4.3.1. Block diagram of output



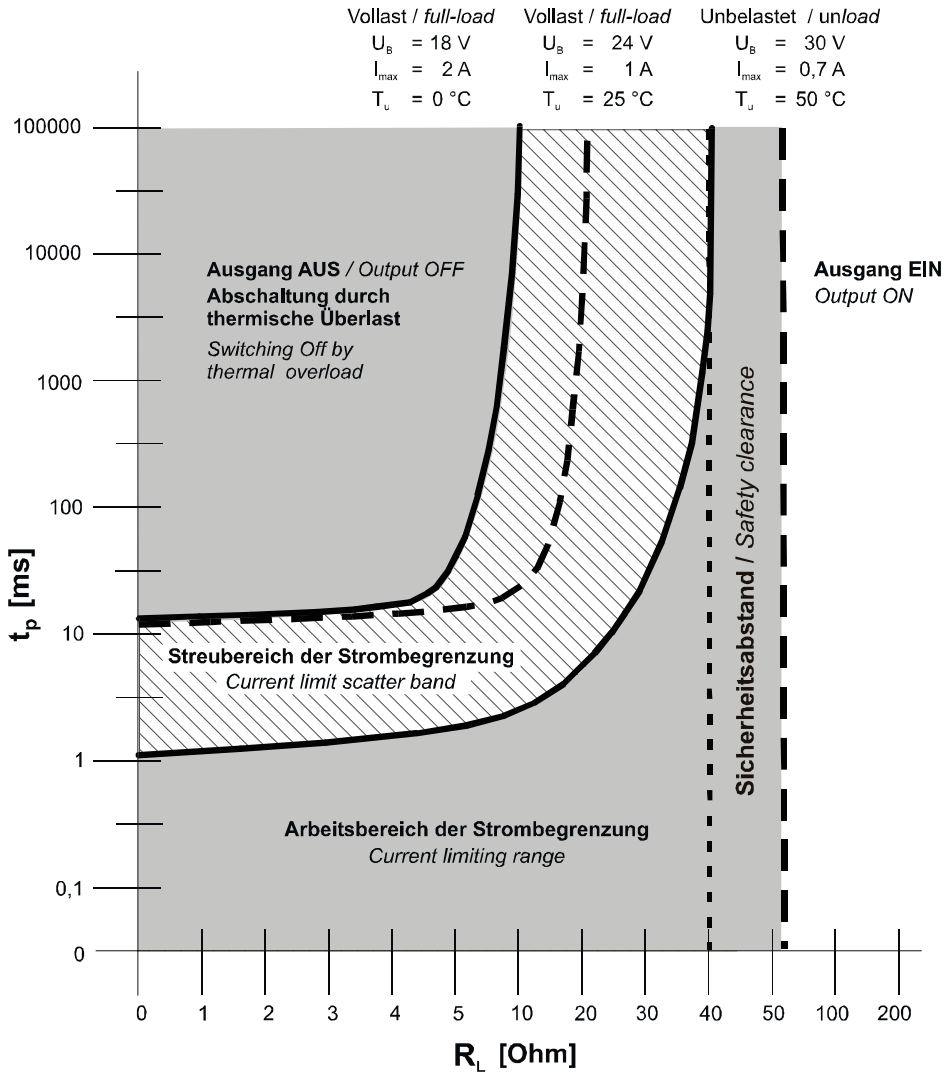
2VF100090DG00.cdr

4.3.2. Digital Outputs Data

Module data	
Number of outputs	8 semiconductor outputs in 2 groups
Type of outputs	semiconductor, non-holding
Suppressor circuit for inductive loads	high-speed de-excitation 50 V terminal voltage (typical) to + 24 V
Power loss due to de-excitation	max. 0.5 watts per output max. 4 watts per module
Status display	yes, yellow LED for each output
Diagnostic function	yes, switching state can be read back at pin
Load connection	
Total loading (100%)	4 A (8 x 0,5 A)
Overload protection	yes, in event of thermal overload Responding of thermal overload protection may influence adjoining outputs
Short-circuit protection [1)] response threshold	yes, electronic current-limiting feature, min. 0.5 A, typically 0.9 A
1) Current is limited electronically. Responding of the short-circuit protection feature produces thermal overload and trips the thermal overload protection circuit.	
Output delay for ‘0’ to ‘1’ for ‘1’ to ‘0’	 max. 0,5 ms max. 0,5 ms
Output capacitance	< 20 nF
Rated voltage	+24 VDC
Voltage drop (at rated current)	< 0,5 V
Rated current for ‘1’ signal	 0,5 A
Leakage current for ‘0’ signal	 max. 0,1 mA
Total current of all outputs	max. 4 A (8 x 0,5)
Total current per group (horizontal mounting on vertical mounting plate)	max. 2 A (4 x 0,5)
Lamp load (+24 VDC)	max. 6 watts
Connection of two outputs in parallel to provide logic operation to increase performance	 allowed not allowed
Insulation resistance	
Rated voltage	0 V <Ue <50 V
Test voltage up to 2,000 m altitude	500 VDC

Overload Reaction of Digital Outputs

Überlast-Verhalten der digitalen Ausgänge
Overload-reaction of digital output



Innerhalb des Streubereichs der Strombegrenzung ist das Verhalten der Strombegrenzung undefiniert.
Within the current-limit scatter band the reaction of current limiting is undefined.

2VF100021DG00.cdr

Note:
It is not possible to know for certain within the current limit scatter band whether the response will be to disconnect or to return to the working range. As a result, this state should be avoided!
The output is ready for operation by elimination of the overload and thermal cooling.

5. Analog Inputs

The Dialog Controller is equipped with 2 analog, multiplexed inputs, protected by diodes. Depending on a software switch the inputs can be operated as unipolar voltage or current inputs. The current sensors can be supplied with +24 V / 50 mA.

The sensors have to be supplied via the module pin Vx.
Vx is electrically isolated from the module supply.

Reading the analog values can be done with the CP1131 library 'AnalogIN.LIB'.

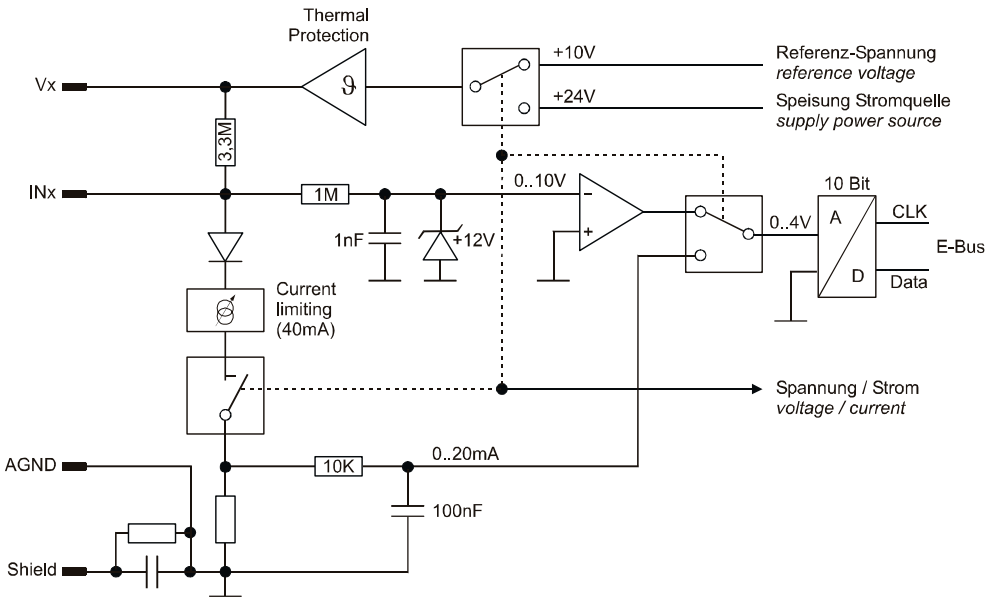
Input channel data

Voltage measurement	
Input voltage rated value	0..10 V
Input voltage, maximum rating	+ 12 V
Input resistance	10 MOhm, typical
Current measurement	
Input voltage rated value	4..20 mA
Input voltage, maximum rating	40 mA (U_e = +24 V)
General data	
Input filter, 1st degree	τ = 1 ms
Conversion method	successive approximation, no error codes
Resolution voltage current	10 bit; 1 LSB = 9,8 mV 10 bit; 1 LSB = 19,6 µA
Precision in temp. range 0-50°C voltage current	 +/- 1 LSB; +/- 1 % +/- 1 LSB; +/- 1 %
Reverse voltage protection	yes
Sampling Rate	150 kHz

Note:
The burden of the current measurement mode is non linear, because of it's internal protect in circuits.

5.1.1. Analog Input (block diagram, 1 channel)

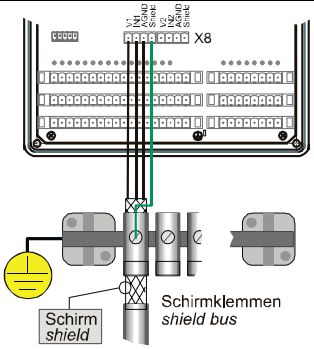
Note:
After *'Power Up'*, the mode *'voltage measurement'* (+10 V) is preset, as diagrammed on the block diagram.



5.1.2. Sensor connection, examples

Note:
The measuring line shield can, where required, be connected to SHIELD.

Where better HF contact of the shields is possible at another point in the casing, there should be no additional connection to SHIELD on the module.



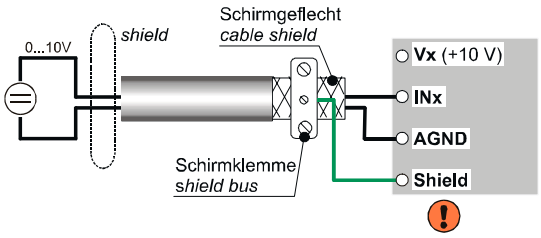
2VF100136DG00.cdr

Voltage measurement

Floating sensors

Connection via shielded cable.

The sensor is connected in the required polarity.



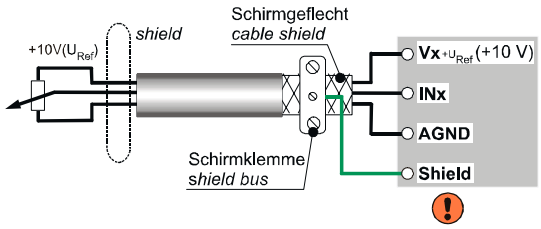
2VF100137DG00.cdr

Sensor connection via internal reference voltage (+10 V)

Connection via shielded cable.

The sensor is connected in the required polarity.

It must be ensured that the reference voltage is not overloaded and that the current taken from the reference source can flow back again via the AGND connections.



2VF100138DG00.cdr

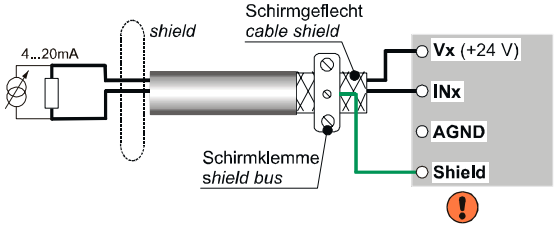
Current measurement

Sensors with auxiliary power connection (4..20 mA / +24 V)

Connection via shielded cable. Sensors with auxiliary power supply.

The sensor is connected in the required polarity. The AGND connection of the module remains open.

It is to be ensured that the potential equalisation via the common supply and sensors actually occurs.



2VF100139DG00.cdr

blank page

6. Configuration of the Remote Display

A Remote Display will have to be configured before it can be used in an application for the first time. This configuration is done using the Remote Display's operator level (keyboard/display).
It is not possible to perform the configuration using the CNW Tool (**C**ANtrol **N**ode **W**izard) provided for CANtrol automation systems.

Warning !

In configuration operation, the Remote Display cannot carry out any control functions – it interprets all input as configuration data.

Make sure that you shut down all machines, equipment or parts thereof that are operated by the Remote Display during configuration!

Start configuration

When the supply voltage is switched on, key No. 07 *(key S9 on the device)* must be pressed to start configuration mode.
The display shows the configuration menu. The picture shows the pre-set parameters. The parameters required for the CANopen communication >*Node ID*< (ID No.) and >*CAN Baudrate*< are entered using the keyboard.

2VF100132DG01.cdr

Quit configuration

To quit configuration mode, switch off the Remote Display's supply voltage.

Start application mode

Restart the device to start up application mode.
Turn the supply voltage on again without touching any keys. Application mode will now be active.

6.1. Settings

Node ID

The individual CAN bus subscribers are identified within the CANopen communication with a unique node ID.
Node IDs can be numbered between 1 and 127. They are shown on the display as hexadecimal digits (01 to 7F). The node ID numbers can be freely allocated, but they do have an influence on the priorities of the individual CAN subscribers (*not to be confused with CAN message identifiers*).

Note:
Node ID numbers must always be unambiguously allocated among the subscribers of the same physical CAN line.

CAN baud rate

Successful CAN communication depends on a single baud rate setting for all bus subscribers. The values chosen should correspond to the maximum line length, but should not be unnecessarily high (additional safety factor).

Contrast

Valid contrast values range between 0_{hex} and 50_{hex}.
(pre-set factory settings 25_{hex} standard)

Real-time clock

The optional real-time clock is controlled by the CAN bus.
For more information, please refer to the section *'Object Dictionary'* under *'0x2110 Set Clock'* (ff.).

Operation

Parameters are set using the numeric keys by pressing one of the keys shown.

Keys	Description
7 + 8 ▲ 9 ▲▲	Parameter is set to the next lowest value
1 – 2 ▼ 3 ▼▼	Parameter is set to the next highest value
0	The pre-set values are displayed
↵ ACK	Value is confirmed and saved

Note:
Only values that were confirmed by pressing the ↵ ACK key will actually be saved!

Dark display

If the display looks dark after switching the device on, it is possible that an unsuitable contrast setting has been chosen.
To change the contrast setting, switch off the Remote Display's supply voltage and follow the steps described under *"Start configuration"*.

With the key F2 the pre-set factory settings (09_{hex}) are active. If the display remains dark, there is a failure.

6.2. Real-time Clock with Buffer Battery

The Remote Display is equipped with a real-time clock.

Set clock

Please refer to the section *'Object Dictionary'* under *'0x2110 Set Clock'* (ff.) for information on setting the clock.

Power supply

A battery is required to power this clock.

Ladezustand

The buffer battery's charge state is monitored by the Remote Display.
It is possible to see the battery's current charge state by means of the CANopen object '*0x2100 Battery Charger State*'.

Caution

Buffer battery voltages which
exceed 3.2 V or
fall below 2.0 V jeopardise the function of the real-time clock.
When it exceeds level: Check battery type and hardware.
When it falls below level: Replace battery.

The following battery charge states should be noted:

3,2 Volt	Typical voltage for a new battery. **This value must not be exceeded**!
3,0 Volt	Battery's nominal voltage
2,5 Volt	Battery must be replaced
2,0 Volt	Battery must be replaced immediately to ensure continued function of real-time clock.

Warning !

Explosion hazard!
Do not dispose of new or discharged batteries in fire, do not solder cell body and do not recharge. Do not dismantle battery.
Always replace battery with a battery of the same type.
Check for correct polarity when inserting battery.

6.2.1. Replacing the battery

The buffer battery must be replaced at least every 3 years, regardless of the charge state.

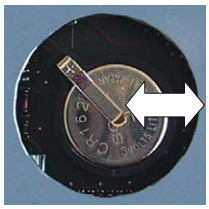
Warning ! Replace Battery With Type CR1620 (Lithium Battery 3V), manufactured by SONY only **(or equivalent manufacturer, abnormal charging current is 2.5 mA or higher)**. Use of Another Battery May Present A Risk Of Fire Or Explosion.

It can be replaced without losing any data in the real-time clock, provided that the Remote Display's supply voltage is switched on. If the supply voltage is not switched on, a maximum time of 20 seconds is available for the battery replacement. After this time, the clock is reset.

Battery replacement:

Note:
The battery is held by a clip. Do not lift this clip, because it can be destroyed.

- Lift up Battery case cover.
- Remove used battery slide out sidewards.
- Insert new battery slide in sidewards, check for correct polarity and battery type.
- Attach battery case cover on the previous place.



2VF100133DG00.cdr

6.3. Membrane Keyboard and Matrix Coding

The keys and LEDs are identified through matrix coding.

Keyboard matrix

The keys are coded in a 5 x 7 matrix.
The coding is done using 5 bytes; within one byte each key is assigned to one bit.

Key actuated = Bit set (1)
Key not actuated = Bit reset (0)

Caution

The matrix circuit recognises a maximum of 2 actuated keys simultaneously.
An incorrect interpretation can occur if more than 2 keys are actuated simultaneously.
Consider this when using the application in order to ensure safe operation.

Keyboard codes

The keyboard codes are transmitted directly via the CAN bus.
The CAN telegram is 5 bytes long and is transmitted with each active state change of a key (press/release).

Bit No. / HexCode	D7 80	D6 40	D5 20	D4 10	D3 08	D2 04	D1 02	D0 01	
Byte 0	X	07	06	05	04	03	02	01	Key
		S9	not assigned	not assigned	not assigned	not assigned	not assigned	not assigned	Assignment
Byte 1	X	14	13	12	11	10	09	08	Key
		S6	S3	F5	F4	F3	F2	F1	Assignment
Byte 2	X	21	20	19	18	17	16	15	Key
		S2	S1	SHIFT	ESC	9	8	7 +	Assignment
Byte 3	X	28	27	26	25	24	23	22	Key
		S5	S4	DEL INS	•	6	5 HOME	4	Assignment
Byte 4	X	35	34	33	32	31	30	29	Key
		S8	S7	ACK	0	3	2	1 _	Assignment

LED matrix

The LEDs are coded in a 6 x 6 matrix.
The coding is done using 6 bytes; within one byte each LED is assigned to one bit.

Bit set (1) = LED lights up
Bit reset (0) = LED unlit
Each LED can also be operated in flash mode.

Bit No. *HexCode*	***D7*** *80*	***D6*** *40*	***D5*** *20*	***D4*** *10*	***D3*** *08*	***D2*** *04*	***D1*** *02*	***D0*** *01*	
Byte 0	06	05	04	03	02	01	X	X	LED
	not assigned	not assigned	green S6	green S1	red S6	red S1			Assignment
Byte 1	12	11	10	09	08	07	X	X	LED
	not assigned	not as-signed	green S7	green S2	red S7	red S2			Assignment
Byte 2	18	17	16	15	14	13	X	X	LED
	not assigned	not assigned	green S8	green S3	red S8	red S3			Assignment
Byte 3	24	23	22	21	20	19	X	X	LED
	not assigned	not assigned	green S9	green S4	red S9	red S4			Assignment
Byte 4	30	29	28	27	26	25	X	X	LED
	not assigned	not assigned	L1	green S5	not assigned	red S5			Assignment
Byte 5	36	35	34	33	32	31	X	X	LED
	L3	orange SHIFT	not assigned	not assigned	L2	L4			Assignment

7. Object Dictionary

7.1. Overview

The module is a CANopen Slave device with a manufacturer-specific device profile and a communication profile in accordance with Draft Standard DS 301 of the CAN user's organisation 'CAN in Automation' (CiA). All device profile and communication profile parameters are stored in an object dictionary.
The communication profile's basic utilities comprise the following functions.

Boot Up

When the supply voltage is applied, the module executes a boot up process and goes directly into CANopen operational state.
It can then be operated straight away by means of CANopen SDO/PDO accesses.
The following text appears on the display:

Display / 240x64 DD.MM.YYYY
(DD.MM.YYYY indicates the creation date of the system software)

When the supply voltage is disconnected, the display turns off and all LEDs are set to 0 (low). After restarting, the LEDs remain in this state.

SDO Telegrams

Service Data Objects (SDOs):
Full access to the object dictionary entries is possible via the SDO channel.
When accessing the object dictionary by SDO, the individual objects are selected using an index and subordinate subindex.
The index is displayed as a 16-bit value and the subindex as an 8-bit value.
Only manufacturer-specific entries are available for LED and display control.

PDO Telegrams

Process Data Objects (PDOs):
The module supports 1 transmit event PDO.
The transmit PDO is active without further configuration, once the remote module is operative. The PDO cannot be changed or deactivated.
Transmit PDO: Keyboard reports

7.2. Process Data Objects (PDOs) for Keyboard and LEDs

Process Data Objects (PDOs) are defined as individual, unconfirmed CAN telegrams with a maximum data length of 8 bytes. PDOs are suitable for fast data transfer without any protocol overhead and can be transmitted by all users in the network. A PDO therefore always has one transmitter and it can be processed by one or more receiver(s).
All PDOs supported by the module are exclusively defined as asynchronous event PDOs. They are not suitable for cyclic data exchange. The data format of the individual PDOs is preset and unchangeable. The PDO is available immediately after boot up.

Identifier

Just like each CAN telegram, the PDOs also have identifiers.
This identifier allocation is structured in accordance with CANopen in such a way that the CANopen Slave device involved can be recognised using the node ID. In addition, the PDO's transmit/receive feature is coded from the viewpoint of the CANopen Slave device. With 11 bits, 2048 identifiers are available. A separate starting point is defined for each of the various receive and transmit PDOs, to which the node ID is added.

Transmit PDO: 384 + node ID

Example of calculation of a transmit PDO:
The preset value in the module is node ID 64 (40_{hex})

Transmit PDO => 384 (180_{hex}) + 64 (40_{hex}) = 448 ($1C0_{hex}$)

Transmit PDO

The transmit PDO contains 5 bytes of user data. It is only transmitted when there is an event. A telegram is transmitted when a key is pressed and released. Keeping a key pressed will not cause multiple telegrams to be sent. The complete keyboard matrix is mapped in telegram format (PDO mapping), as described in the section *"Membrane Keyboard and Matrix Coding"*.

Name	**Comment** (number of mapped objects)
Byte 0 Keyboard Matrix	1st object to be mapped
Byte 1 Keyboard Matrix	2nd object to be mapped
Byte 2 Keyboard Matrix	3rd object to be mapped
Byte 3 Keyboard Matrix	4th object to be mapped
Byte 4 Keyboard Matrix	5th object to be mapped

7.3. Service Data Objects (SDOs)

All object dictionary entries can be accessed via SDO telegrams. These objects are identified by means of a 16-bit index and an 8-bit subindex. Therefore, one telegram format is sufficient in order to be able to read or write various data.
An SDO data transfer always includes one request telegram and one response telegram. These telegrams can be recognised by the CAN identifier.

Identifier

The identifier allocation is structured in accordance with CANopen in such a way that the CANopen Slave device involved can be recognised using the node ID. In addition, the SDO's transmit/receive feature is coded.
With 11 bits a field of 2048 identifiers is available. A starting point is defined for the SDO, to which the node ID is added.

Request telegram (Master > Slave): 1536 + node ID
Response telegram (Slave > Master): 1408 + node ID

Example of calculation of SDO identifiers:
The preset value in the module is node ID 64 (40_{hex})

Request telegram: => 1536 (600_{hex}) + 64 (40_{hex}) = 1600 (640_{hex})
Response telegram: => 1408 (580_{hex}) + 64 (40_{hex}) = 1472 ($5C0_{hex}$)

SDO Protocol

The module only supports write access, that is the downloading of data from the CANopen Master to the CANopen Slave (Remote Display).
Expedited and segmented download is implemented.
Expedited download <= 4 bytes of user data
Segmented download > 4 bytes of user data

Note:
For a detailed description of SDO telegrams please refer to the section *'Telegram Formats for CANopen SDO Telegrams'*.

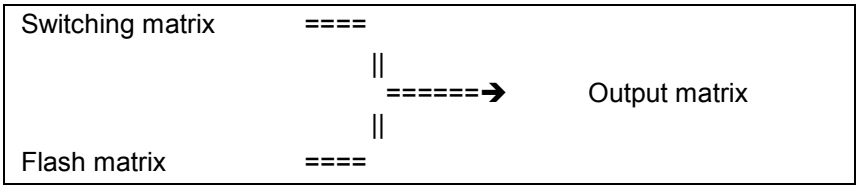
7.4. Switching/Flashing of LEDs

Each LED can be operated in two different modes:

- Switch on/switch off mode
- Flash mode

With the objects available, up to 8 LEDs are simultaneously activated via SDO. This is the result of the byte-oriented structure of the LED matrix.

The user has one switching object and one flashing object available. These objects are represented internally in a switching matrix and a flash matrix. The combination of these two matrixes results in the output matrix, which contains the complete output image for controlling the LEDs. This makes it possible to implement flashing and switching LEDs simultaneously via one byte of the output matrix.



The switching matrix and flash matrix values are transferred to the output matrix under observance of the following rules:

- With the switching of an LED via the switching matrix, the rhythm of adjacent LEDs, which are just then active in the flash function, is not disturbed.
- The LED will be switched immediately in the first flash cycle. Only afterwards will it be synchronised with the flashing rate of all flashing LEDs.
  This means that there is an immediate optical command acknowledgement available.
  The initial flashing rate after switching on a flash function can therefore be up to twice as long as normal.
- If a flashing LED is switched off via the flash matrix, the switching-off does not occur until the next switch-off phase. In this way even brief flash commands are visible.
  The LED always illuminates for at least one complete duty cycle.
- Bi-coloured LEDs can flash alternately in both colours.
  For this, one of the two LEDs is switched on by switch command and the other activated by flash command. In this way, the first LED flashes inversely to the cycle of the second, flashing LED.

7.5. Manufacturer-Specific Device Profile (Object Dictionary)

This section describes all objects supported by the Remote Display.

7.5.1. Overview of Object Dictionary

LED control

Index	Object	Name	Type	Acc
0x2030	Array	LED On / Off	unsigned 8	wo
0x2040	Array	LED flash	unsigned 8	wo
0x2050	Var	LED flash ratio	octet string	wo

Text display functions

0x2000	Var	Clear Screen	unsigned 8	wo
0x2010	Var	Set Text Cursor Position	unsigned 8	wo
0x2800	Array	Write Text 8x8	string	wo
0x2810	Array	Write Text 16x16	string	wo

Graphical display functions

0x2020	Var	Set Graphic Cursor Position	unsigned 8	wo
0x2820	Var	Write Graphic Pixel	octet string	wo

Real-time clock functions (optional)

0x2100	Var	Battery Charger State	unsigned 8	ro
0x2110	Array	Set Clock	unsigned 8	wo
0x2120	Array	Read Clock	unsigned 8	ro
0x2130	Record	Show Clock	unsigned 8	wo

7.5.2. Notes on Object Dictionary

0x2030	**LED ON/OFF**	
	Used for writing the LED switching matrix. The bit distribution corresponds to the matrix wiring, as described in the section *"Membrane Keyboard and Matrix Coding"*.	
OBJECT DESCRIPTION		
	Index	0x2030
	Name	LED ON/OFF
	Object Code	Array
VALUE DESCRIPTION		
	Subindex	**0**
	Description	1st Matrix Row
	Access Level	User
	Object Class	Optional
	PDO Mapping	No
	Data Type	Unsigned 8
	Length	1
	Value Range	No
	Mandatory Range	No
	Subindex	**1**
	Description	2nd Matrix Row
	Access Level	User
	Object Class	Optional
	PDO Mapping	No
	Data Type	Unsigned 8
	Length	1
	Value Range	No
	Mandatory Range	No
	Subindex	**2**
	Description	3rd Matrix Row
	Access Level	User
	Object Class	Optional
	PDO Mapping	No
	Data Type	Unsigned 8
	Length	1
	Value Range	No
	Mandatory Range	No

VALUE DESCRIPTION		
	Subindex	**3**
	Description	4th Matrix Row
	Access Level	User
	Object Class	Optional
	PDO Mapping	No
	Data Type	Unsigned 8
	Length	1
	Value Range	No
	Mandatory Range	No
	Subindex	**4**
	Description	5th Matrix Row
	Access Level	User
	Object Class	Optional
	PDO Mapping	No
	Data Type	Unsigned 8
	Length	1
	Value Range	No
	Mandatory Range	No
	Subindex	**5**
	Description	6th Matrix Row
	Access Level	User
	Object Class	Optional
	PDO Mapping	No
	Data Type	Unsigned 8
	Length	1
	Value Range	No
	Mandatory Range	No

User data bytes in “expedited transfer”

Byte 4	Byte 5	Byte 6	Byte 7
Select LED	-	-	-

Note:
Please refer to the section *“Switching and Flashing of LEDs”* for explanations.

0x2040	**LED Flash**	
	Used for writing the LED flash matrix. The bit distribution corresponds to the matrix wiring, as described in the section *"Membrane Keyboard and Matrix Coding"*.	
OBJECT DESCRIPTION		
	Index	0x2040
	Name	LED flash
	Object Code	Array
VALUE DESCRIPTION		
	Subindex	**0**
	Description	1st Matrix Row
	Access Level	User
	Object Class	Optional
	PDO Mapping	No
	Data Type	Unsigned 8
	Length	1
	Value Range	No
	Mandatory Range	No
	Subindex	**1**
	Description	2nd Matrix Row
	Access Level	User
	Object Class	Optional
	PDO Mapping	No
	Data Type	Unsigned 8
	Length	1
	Value Range	No
	Mandatory Range	No
	Subindex	**2**
	Description	3rd Matrix Row
	Access Level	User
	Object Class	Optional
	PDO Mapping	No
	Data Type	Unsigned 8
	Length	1
	Value Range	No
	Mandatory Range	No

VALUE DESCRIPTION		
	Subindex	**3**
	Description	4th Matrix Row
	Access Level	User
	Object Class	Optional
	PDO Mapping	No
	Data Type	Unsigned 8
	Length	1
	Value Range	No
	Mandatory Range	No
	Subindex	**4**
	Description	5th Matrix Row
	Access Level	User
	Object Class	Optional
	PDO Mapping	No
	Data Type	Unsigned 8
	Length	1
	Value Range	No
	Mandatory Range	No
	Subindex	**5**
	Description	6th Matrix Row
	Access Level	User
	Object Class	Optional
	PDO Mapping	No
	Data Type	Unsigned 8
	Length	1
	Value Range	No
	Mandatory Range	No

User data bytes in “expedited transfer”

Byte 4	Byte 5	Byte 6	Byte 7
Select Flash LED	-	-	-

Note:
Please refer to the section *“Switching and Flashing of LEDs”* for explanations.

0x2050	LED Flash Ratio	
	Set LED flash frequency.	
OBJECT DESCRIPTION		
	Index	0x2050
	Name	LED flash ratio
	Object Code	Var
VALUE DESCRIPTION		
	Subindex	**0**
	Description	LED flash ratio
	Access Level	User
	Object Class	Optional
	PDO Mapping	No
	Data Type	Octet string
	Length	2
	Value Range	01..7F
	Mandatory Range	No

User data bytes in “expedited transfer”

Byte 4	Byte 5	Byte 6	Byte 7
‘On’ duration	‘Off duration’	-	-

Note:
The frequency is defined through the ‘on’ duration/‘off’ duration ratio. It applies for all LEDs and is automatically synchronised regardless of the moment of switching on. ‘On’ duration and ‘off’ duration can be set independently of each other.
The value range goes from 01_{hex} to $7F_{hex}$ and is divided into stages of approximately 10 ms.

0x2000	**Clear Screen** The object clears the screen and places the text and graphic cursor at top left *(row 1/column 1)*.	
OBJECT DESCRIPTION		
	Index	0x2000
	Name	Clear Screen
	Object Code	Var
VALUE DESCRIPTION		
	Subindex	**0**
	Description	Clear Screen
	Access Level	User
	Object Class	Optional
	PDO Mapping	No
	Data Type	Unsigned 8
	Length	1
	Value Range	No
	Mandatory Range	No

Note:
Subindex and user data are not then evaluated and can stand as they are.

0x2010	**Set Text Cursor Position**	
	Position text cursor.	
OBJECT DESCRIPTION		
	Index	0x2010
	Name	Set Text Cursor Position
	Object Code	Var
VALUE DESCRIPTION		
	Subindex	**0**
	Description	Set Text Cursor Position
	Access Level	User
	Object Class	Optional
	PDO Mapping	No
	Data Type	Array of Unsigned 8
	Length	2
	Value Range	Z_POS: 1..8 (max. 8 lines) S_POS: 1..30 (max. 30 columns/characters)
	Mandatory Range	No

User data bytes in “expedited transfer”

Byte 4	Byte 5	Byte 6	Byte 7
Z_POS *(line position)*	S_POS *(column position)*	-	-

The text cursor is positioned on the graphical display under consideration of the character size (8 x 8 pixels). This corresponds to a maximum of 240 characters in 8 lines. The cursor supports the character-exact, but not the pixel-exact positioning of fonts.

Note:
The cursor is always invisible and marks the position at which the next ASCII character is displayed.
If the application shows a visible cursor, then for example the special character with the ASCII code 0xDB (219_{dec}) can be displayed as a string with a character.
A flashing cursor or different cursor shapes are not possible.

0x2800	**Write Text 8x8** Write text in 8 x 8 pixel format.	
OBJECT DESCRIPTION		
	Index	0x2800
	Name	Write Text 8x8
	Object Code	Array
VALUE DESCRIPTION		
	Subindex	**0**
	Description	Write Text 8x8
	Access Level	User
	Object Class	Optional
	PDO Mapping	No
	Data Type	Octet string
	Length	1 ..30
	Value Range	20 .. 7A (hex)
	Mandatory Range	No

The subindex can be used to implement attributes such as underlining, strike-through, inversion and other cursor shapes.
Each bit in the subindex is inverted when the pixel row of a character is output. The subindex contains with one byte exactly 8 bits. In this way, all 8 rows of a 8 x 8 character pattern can be addressed.

Examples of subindex-controlled functions:

Subindex	Description
0	Outputs non-inverted text
0x80	Outputs inverted characters
0xFF	Outputs underlined characters

ASCII Codes

All ASCII codes are implemented.

Note:
The text output does not include line breaks!
The output of several text lines or a complete screen page is implemented on a line-by-line basis. If more characters are output than can be displayed in a line, the text is output at the start of a line again, but is shifted one pixel row underneath.

0x2810	**Write Text 16x16**	
	Writing of text data in 16 x 16 pixel format.	
OBJECT DESCRIPTION		
	Index	0x2800
	Name	Write Text 16x16
	Object Code	Array
VALUE DESCRIPTION		
	Subindex	**0**
	Description	Write Text 16x16
	Access Level	User
	Object Class	Optional
	PDO Mapping	No
	Data Type	Octet string
	Length	1 ..15
	Value Range	20 .. 7A (hex)
	Mandatory Range	No

The subindex can be used to implement attributes such as underlining, strike-through, inversion and other cursor shapes.
Each bit in the subindex is inverted when the double pixel row of a character is output. The subindex contains with one byte exactly 8 bits. In this way all double rows of the 16 x 16 character pattern are addressed.
Examples of subindex-controlled functions:

Subindex	Description
0	Outputs non-inverted text
0x80	Outputs inverted characters
0xFF	Outputs underlined characters

ASCII Codes All ASCII codes are implemented.

Note:
The text output does not include line breaks!
The output of several text lines or a complete screen page is implemented on a line-by-line basis. If more characters are output than can be displayed in a line, the text is output at the start of a line again, but is shifted one pixel row underneath.
If the full line width is used, ensure that with all 16 x 16 pixel text output that the text cursor is located at the start at an odd number (1; 3 ; 5; ...). Only in this way can a character be fully displayed at the end of a line.

0x2020	Set Graphic Cursor Position	
	Set graphic cursors at a defined position.	
OBJECT DESCRIPTION		
	Index	0x2020
	Name	Set Graphic Cursor Position
	Object Code	Var
VALUE DESCRIPTION		
	Subindex	**0**
	Description	Set Graphic Cursor Position
	Access Level	User
	Object Class	Optional
	PDO Mapping	No
	Data Type	Unsigned16
	Length	2
	Value Range	0..1919
	Mandatory Range	No

User data bytes in “expedited transfer”

Byte 4	Byte 5	Byte 6	Byte 7
LSB	MSB	-	-

The graphical display has a resolution of 240 x 64 pixels. The 240 pixels of a line have 8 bits combined in one byte. In this way, 30 bytes each with 8 bits depict a line with 240 pixels. The graphic cursor is controlled via the byte positioning.

Line 1	Byte Position 0	»	Byte Position 29
	Byte 0		Byte 29
	Graphic Cursor Pos. 0		**Graphic Cursor Pos. 29**
Line 2	Byte Position 30	»	Byte Position 59
	Byte 0		Byte 29
	Graphic Cursor Pos. 30		**Graphic Cursor Pos. 59**
Line 3	Byte Position 60	»	Byte Position 89
	Byte 0		Byte 29
	Graphic Cursor Pos. 60		**Graphic Cursor Pos. 89**
	•		•
	•		•
	•		•
Line 64	Byte Position 1890	»	Byte Position 1919
	Byte 0		Byte 29
	Graphic Cursor Pos. 1890		**Graphic Cursor Pos. 1919**

Note:
Further information is contained in the object *‘Write Graphic Pixel’*.

0x2820	**Write Graphic Pixel**	
	Write graphic data.	
OBJECT DESCRIPTION		
	Index	0x2820
	Name	Write Graphic Pixel
	Object Code	Var
VALUE DESCRIPTION		
	Subindex	**0**
	Description	Set Graphic Cursor Position
	Access Level	User
	Object Class	Optional
	PDO Mapping	No
	Data Type	Octet string
	Length	1 ..240
	Value Range	No
	Mandatory Range	No

Note:
Before graphic data are written, the graphic cursor must be correctly positioned over the object *'Set Graphic Cursor Position'*.

The graphical display has a resolution of 240 x 64 pixels. The 240 pixels of a line have 8 bits combined in one byte. In this way, 30 bytes each with 8 bits depict a line with 240 pixels. The graphic cursor is controlled via the byte positions. The pixels are controlled via the bits within a byte.

Line 1	**Byte 0**								»	**Byte 29**							
	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0		2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	80	40	20	10	08	04	02	01		80	40	20	10	08	04	02	01
Line 2	**Byte 30**								»	**Byte 59**							
	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0		2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	80	40	20	10	08	04	02	01		80	40	20	10	08	04	02	01
Line 3	**Byte 60**								»	**Byte 89**							
	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0		2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	80	40	20	10	08	04	02	01		80	40	20	10	08	04	02	01
	•									•							
	•									•							
	•									•							
Line 64	**Byte 1890**								»	**Byte 1919**							
	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0		2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	80	40	20	10	08	04	02	01		80	40	20	10	08	04	02	01

Example:
In the graphical display, the content 1000 0001_{bin} (81_{hex}) sets the first and last pixel of the byte, whose start position was defined by the graphic cursor.

Segmented Transfer Up to 240 bytes of graphics data can be transferred in a telegram sequence *(segmented transfer)*. A complete graphics page with 1920 bytes requires a maximum of 8 telegrams in *'Segmented Transfer Mode'*.

The graphic cursor and line breaks are automatically controlled by the data output.

Expedited Transfer Graphics data of up to 4 bytes in length use the *'expedited transfer'*.

0x2100	Battery Charger State	
	Read the battery charger state of the real-time clock.	
OBJECT DESCRIPTION		
	Index	0x2100
	Name	Battery Charger State
	Object Code	Var
VALUE DESCRIPTION		
	Subindex	**0**
	Description	Battery Charger State
	Access Level	User
	Object Class	Optional
	PDO Mapping	No
	Data Type	Unsigned 8
	Length	1
	Value Range	No
	Mandatory Range	No

User data bytes in “expedited transfer”

Byte 4	Byte 5	Byte 6	Byte 7
Battery Status	-	-	-

Battery Charger States

Battery State (hex)	Voltage	Description
FF	**4.2**	‘Overflow’ - Battery not inserted or incorrectly inserted; or max. permissible voltage limit exceeded
F3	4.0	Measuring range limit
D5	3.5	
C2	**3.2**	Typical voltage for a new battery. **This value must not be exceeded**!
BD	3.1	
B7	3.0	Battery’s nominal voltage
98	2.5	Warning level for battery replacement
78	**2.0**	Battery must be replaced immediately to ensure continued function of real-time clock.
5B	1.5	
3D	1.0	
20	0.5	
00	0.0	

Caution

All buffer battery voltage levels which
exceed **3.2 V** ($C2_{hex}$) or
fall below **2.0 V** (78_{hex}) jeopardise the function of the real-time clock.
When it exceeds level: Check battery type and hardware
When it falls below level: Replace battery.

For more information, please refer to the section *‘Real-time Clock and Buffer Battery’*.

0x2110	**Set Clock** Write time and date of real-time clock	
OBJECT DESCRIPTION		
	Index	0x2110
	Name	Set Clock
	Object Code	Array
VALUE DESCRIPTION		
	Subindex	**0**
	Description	Sec
	Access Level	User
	Object Class	Optional
	PDO Mapping	No
	Data Type	Unsigned 8
	Length	1
	Value Range	No
	Mandatory Range	No
	Subindex	**1**
	Description	Min
	Access Level	User
	Object Class	Optional
	PDO Mapping	No
	Data Type	Unsigned 8
	Length	1
	Value Range	No
	Mandatory Range	No
	Subindex	**2**
	Description	Hour
	Access Level	User
	Object Class	Optional
	PDO Mapping	No
	Data Type	Unsigned 8
	Length	1
	Value Range	No
	Mandatory Range	No
	Subindex	**3**
	Description	Day of Week
	Access Level	User
	Object Class	Optional
	PDO Mapping	No
	Data Type	Unsigned 8
	Length	1
	Value Range	No
	Mandatory Range	No

VALUE DESCRIPTION		
	Subindex	**4**
	Description	Day
	Access Level	User
	Object Class	Optional
	PDO Mapping	No
	Data Type	Unsigned 8
	Length	1
	Value Range	No
	Mandatory Range	No
	Subindex	**5**
	Description	Month
	Access Level	User
	Object Class	Optional
	PDO Mapping	No
	Data Type	Unsigned 8
	Length	1
	Value Range	No
	Mandatory Range	No
	Subindex	**6**
	Description	Year
	Access Level	User
	Object Class	Optional
	PDO Mapping	No
	Data Type	Unsigned 8
	Length	1
	Value Range	No
	Mandatory Range	No

User data bytes in “expedited transfer”

Byte 4	Byte 5	Byte 6	Byte 7
Clock	-	-	-

Note:
The clock data are transferred in hexadecimal format.
The value 19_{hex} is evaluated in the real-time clock as a BCD number and therefore corresponds to the decimal value of 19.00 hours.

0x2120	**Read Clock** Read time and date of real-time clock	
OBJECT DESCRIPTION		
	Index	0x2120
	Name	Read Clock
	Object Code	Array
VALUE DESCRIPTION		
	Subindex	**0**
	Description	Sec
	Access Level	User
	Object Class	Optional
	PDO Mapping	No
	Data Type	Unsigned 8
	Length	1
	Value Range	No
	Mandatory Range	No
	Subindex	**1**
	Description	Min
	Access Level	User
	Object Class	Optional
	PDO Mapping	No
	Data Type	Unsigned 8
	Length	1
	Value Range	No
	Mandatory Range	No
	Subindex	**2**
	Description	Hour
	Access Level	User
	Object Class	Optional
	PDO Mapping	No
	Data Type	Unsigned 8
	Length	1
	Value Range	No
	Mandatory Range	No
	Subindex	**3**
	Description	Day of Week
	Access Level	User
	Object Class	Optional
	PDO Mapping	No
	Data Type	Unsigned 8
	Length	1
	Value Range	No
	Mandatory Range	No

VALUE DESCRIPTION		
	Subindex	**4**
	Description	Day
	Access Level	User
	Object Class	Optional
	PDO Mapping	No
	Data Type	Unsigned 8
	Length	1
	Value Range	No
	Mandatory Range	No
	Subindex	**5**
	Description	Month
	Access Level	User
	Object Class	Optional
	PDO Mapping	No
	Data Type	Unsigned 8
	Length	1
	Value Range	No
	Mandatory Range	No
	Subindex	**6**
	Description	Year
	Access Level	User
	Object Class	Optional
	PDO Mapping	No
	Data Type	Unsigned 8
	Length	1
	Value Range	No
	Mandatory Range	No

User data bytes in “expedited transfer”

Byte 4	Byte 5	Byte 6	Byte 7
Clock	-	-	-

Note:
The clock data are transferred in hexadecimal format.
The value 19_{hex} is evaluated in the real-time clock as a BCD number and therefore corresponds to the decimal value of 19.00 hours.

0x2130	**Show Clock** Show time on the display.	
OBJECT DESCRIPTION		
	Index	0x2130
	Name	Show Clock
	Object Code	Array
VALUE DESCRIPTION		
	Subindex	**0**
	Description	Stop Clock
	Access Level	User
	Object Class	Optional
	PDO Mapping	No
	Data Type	Unsigned 8
	Length	1
	Value Range	No
	Mandatory Range	No

Stops the time output and frees up the display area for overwriting. The last valid time is kept, but the clock continues to work internally.

Note:
The user data are not then evaluated and can stand as they are.

	Subindex	**1**
	Description	Show Clock at Text Cursor Position
	Access Level	User
	Object Class	Optional
	PDO Mapping	No
	Data Type	Array of Unsigned 8
	Length	2
	Value Range	Z_POS: 1..8 (max. 8 lines) S_POS: 1..30 (max. 30 columns/characters)
	Mandatory Range	No

User data bytes in “expedited transfer”

Byte 4	Byte 5	Byte 6	Byte 7
Z_POS *(line position)*	S_POS *(column position)*	-	-

Note:
The clock is positioned via the text cursor.
You can find more detailed information on cursor positioning in text mode in the object *‘Set Text Cursor Position’*.

Blank page

8. Service Data Object - SDO Protocol

Apart from general functions, such as the activation of a PDO, CANopen modules also have a range of device-specific functions, for example:
- switching of outputs
- referencing of drives
- display of texts

In order to be able to control these functions via the CAN bus, both of the following requirements will need to be fulfilled for CANopen:

- The functions have to be numerically coded.
  For this CANopen uses a 16-bit index and an 8-bit subindex. All available device-specific and general functions are stored in an object dictionary. This object dictionary is an elementary part of a CANopen technical manual.
- A protocol is required to transfer the different functions from the object dictionary.
  In CANopen the protocol is known as Service Data Object (SDO).
  It must be able to read or write data. The protocol must be able to deal with the CAN bus feature that a maximum of 8 bytes fit into a telegram.

SDO Telegram Structure

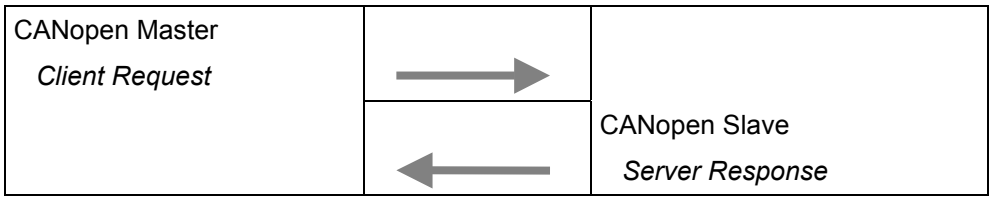
An SDO telegram is always 8 bytes long.
The first byte always contains the codes for reading/writing, etc.
The following 3 bytes contain the index and subindex of the selected function.
A maximum of 4 bytes is available for the function's user data.

Expedited/ Segmented Protocol

A protocol with 4 bytes of user data is known as an expedited protocol.
Because of their functions, devices such as a remote I/O module only have the expedited protocol at their disposal.
A protocol with more than 4 bytes of user data is known as a segmented protocol.
Devices such as displays, to which text sequences are also transferred, require a protocol which handles more than 4 bytes of user data. The protocol controls a sequence of CAN telegrams. In this way it is possible to transmit almost any amount of data.

Both protocols are known as 'confirmed services'.
This means that they always comprise 2 CAN telegrams, a request and a response.
A request usually originates from the CANopen Master. It is the protocol's client.
The response comes from the CANopen Slave, which is the protocol's server.

Expedited Protocol Sequence



Segmented Protocol Sequence

CANopen Master *Client Request*	→	
	←	CANopen Slave *Server Response*
CANopen Master *Client Request*	→	
	←	CANopen Slave *Server Response*
:	:	:
:	:	:
:	:	:
CANopen Master *Client Request*	→	
	←	CANopen Slave Server Response

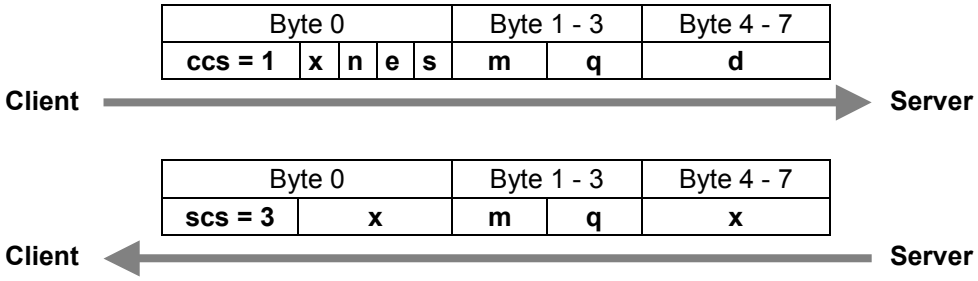
Data Content

The SDO protocols (expedited/segmented) support the reading and writing of data. If data are written from the Master to the Slave, this is known as a *download*. In the download the user data are contained in the request telegram.

If data are read by the Master from the Slave, this is known as an *upload*. In the upload the user data are contained in the response telegram.
A response can also contain error messages.

Note:
Note that in the CAN telegram all data are transmitted in Intel 'Little Endian' format, but are processed and displayed within the module in Motorola format.

8.1. Initiate Domain Download Protocol



Request Telegram Client (CANopen Master):

	Abbrev.	Comment
Byte 0		
Bit 7-5	**ccs: 001**	Client command specifier
Bit 4	**x:**	Reserved, therefore 0
Bit 3-2	**n:**	Number of bytes containing no valid data, valid if e=1 and s=1
Bit 1	**e:**	0: Segmented transfer 1: Expedited transfer; therefore always 1
Bit 0	**s:**	0: Data size 'n' not valid, therefore 0 data bytes 1: Data size is indicated in 'n' (in normal cases)
Byte 1 - 2	**m:**	Multiplexor (index; 16 bit)
Byte 3	**q:**	Multiplexor (subindex; 8 bit)
Byte 4 - 7	**d:**	Data

Response Telegram Server (CANopen Slave):

	Abbrev.	Comment
Byte 0		
Bit 7-5	**scs: 011**	Server command specifier
Bit 4-0	**x:**	Reserved, therefore 0
Byte 1 - 2	**m:**	Multiplexor (index; 16 bit)
Byte 3	**q:**	Multiplexor (subindex; 8 bit)
Byte 4 - 7	**x:**	Reserved, therefore 0

Example of Telegram Transmission of a command with max. 4 bytes of user data – **expedited transfer**

Request Telegram from Client (Master) to Server (Slave)

Byte	0	1 – 2	3	4 – 7	Comment
Value	2F	30 20	01	08 00 00 00	Transmit one user data byte Index 0x2030 / Subindex 0x01

Response Telegram from Server (Slave) to Client (Master)

Byte	0	1 – 2	3	4 – 7	Comment
Value	60	30 20	01	00 00 00 00	Positive response

8.2. Download Domain Segment Protocol

Request Telegram Client (CANopen Master):

	Abbrev.	Comment
Byte 0		
Bit 7-5	**ccs: 000**	Client command specifier
Bit 4	**t:**	Toggle bit changes its value after each telegram sequence, which consists of a request and a response Start value = 0
Bit 3-1	**n:**	Number of bytes containing no segment data if n=0 then no number is given
Bit 0	**c:**	0: Further segment telegrams follow 1: No further segments follow
Byte 1 - 7	**d:**	Segment data

Response Telegram Server (CANopen Slave):

	Abbrev.	Comment
Byte 0		
Bit 7-5	**scs: 001**	Server command specifier
Bit 4	**t**	Toggle bit changes its value after each telegram sequence, which consists of a request and a response Start value = 0
Bit 3-0	**x:**	Reserved, therefore 0
Byte 1 - 7	**x:**	Reserved, therefore 0

Example of Telegram The communication consists of two phases: initialisation and download of the segmented data.

Phase 1 Initialisation with Initiate Domain Download Protocol
Request for a transfer with 15 bytes of user data – **segmented transfer**

Request Telegram from Client (Master) to Server (Slave)

Byte	0	1 – 2	3	4 – 7	Comment
Value	20	00 28	00	0F 00 00 00	Request for 15 bytes of user data

Response Telegram from Server (Slave) to Client (Master)

Byte	0	1 – 2	3	4 – 7	Comment
Value	60	30 20	01	00 00 00 00	Positive Response

Phase 2 Transfer segmented data with Download Domain Segment Protocol

Request Telegram from Client (Master) to Server (Slave)

Byte	0	1 – 7			Comment
Value	00	42 6c	61	20 42 6c 61	Transfer the first 7 bytes

Response Telegram from Server (Slave) to Client (Master)

Byte	0	1 – 7			Comment
Value	20	00 00	00	00 00 00 00	Response

Request Telegram from Client (Master) to Server (Slave)

Byte	0	1 – 7			Comment
Value	10	20 42	6c	61 20 42 6c	Transfer the second 7 bytes

Response Telegram from Server (Slave) to Client (Master)

Byte	0	1 – 7			Comment
Value	30	00 00	00	00 00 00 00	Response

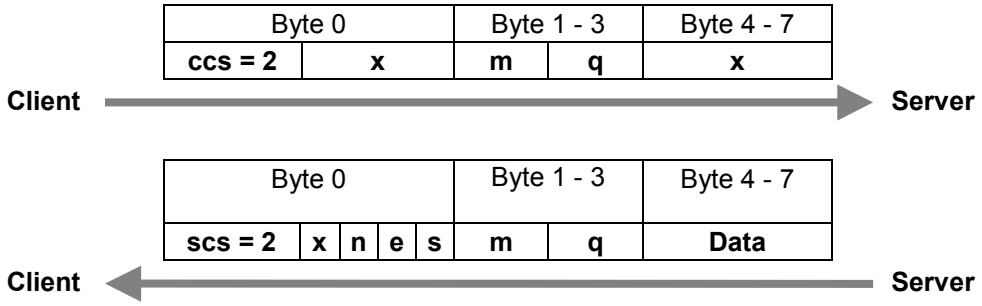
Request Telegram from Client (Master) to Server (Slave)

Byte	0	1 – 7			Comment
Value	0D	61 00	00	00 00 00 00	Transfer the last byte

Response Telegram from Server (Slave) to Client (Master)

Byte	0	1 – 7			Comment
Value	20	00 00	00	00 00 00 00	Response and termination

8.3. Initiate Domain Upload Protocol



Request Telegram Client (CANopen Master):

	Abbrev.	Comment
Byte 0		
Bit 7-5	**ccs: 010**	Client command specifier
Bit 4-0	**x:**	Reserved, therefore 0
Byte 1 - 2	**m:**	Multiplexor (index; 16 bit)
Byte 3	**q:**	Multiplexor (subindex; 8 bit)
Byte 4 - 7	**x:**	Reserved, therefore 0

Response Telegram Server (CANopen Slave):

	Abbrev.	Comment
Byte 0		
Bit 7-5	**scs: 010**	Server command specifier
Bit 4	**x:**	Reserved, therefore 0
Bit 3-2	**n:**	Number of bytes containing no valid data, valid if e=1 and s=1
Bit 1	**e:**	0: Segmented transfer 1: Expedited transfer; therefore always 1
Bit 0	**s:**	0: Data size ‘n’ not valid, therefore 0 data bytes 1: Data size is indicated in ‘n’ (in normal cases)
Byte 1 - 2	**m:**	Multiplexor (index; 16 bit)
Byte 3	**q:**	Multiplexor (subindex; 8 bit)
Byte 4 - 7	**data:**	Data

Example of Telegram Loading of a maximum of 4 bytes of user data – **expedited transfer**

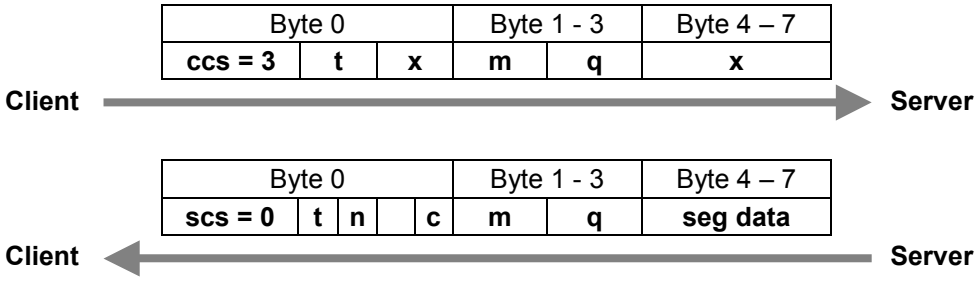
Request Telegram from Client (Master) to Server (Slave)

Byte	0	1 – 2	3	4-7	Comment
Value	40	00 60	01	00 00 00 00	Index 0x6000 Subindex 0x01

Response Telegram from Server (Slave) to Client (Master)

Byte	0	1 – 2	3	4-7	Comment
Value	43	00 60	01	11 17 21 18	Transfer four user data bytes Index 0x6000 / Subindex 0x01

8.4. Upload Domain Segment Protocol



Request Telegram Client (CANopen Master):

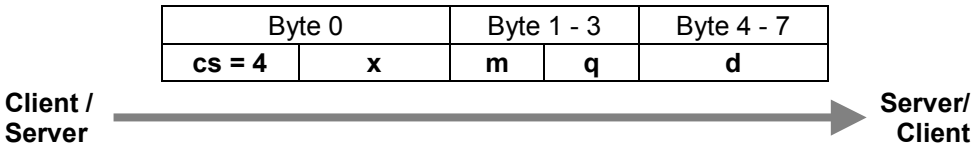
	Abbrev.	Comment
Byte 0		
Bit 7-5	**ccs: 011**	Client command specifier
Bit 4-0	**t:**	Toggle bit changes its value after each telegram sequence, which consists of a request and a response. Start value = 0
Bit 3-0	**x:**	Reserved, therefore 0
Byte 1 - 7	**x:**	Reserved, therefore 0

Response Telegram Server (CANopen Slave):

	Abbrev.	Comment
Byte 0		
Bit 7-5	**scs: 000**	Server command specifier
Bit 4	**t:**	Toggle bit changes its value after each telegram sequence, which consists of a request and a response. Start value = 0.
Bit 3-1	**n:**	Number of bytes containing no segment data if n=0, then no number is given
Bit 0	**c:**	0: Further segment telegrams follow 1: No further segments follow
Byte 1 - 7	**seg data:**	Segment data

8.5. Abort Domain Transfer Protocol

Note:
With this telegram, both the client and the server can abort a domain transfer at any time.



Abort Telegram

Client / Server:

	Abbrev.	Comment
Byte 0		
Bit 4-0	**cs: 100**	Command specifier
Bit 7-5	**x:**	Reserved, therefore 0
Byte 1 - 2	**m:**	Multiplexor (index; 16 bit)
Byte 3	**q:**	Multiplexor (subindex; 8 bit)
Byte 4 - 7	**d:**	Error code (application-specific)

Example of Telegram

Attempt to load a maximum of 4 bytes of user data – **expedited transfer**

Request

Telegram from Client (Master) to Server (Slave)

Byte	0	1 – 2	3	4-7	Comment
Value	40	00 70	01	00 00 00 00	Index 0x7000 Subindex 0x01

Response

Telegram from Server (Slave) to Client (Master) - **abort**

Byte	0	1 – 2	3	4-7	Comment
Value	43	00 70	01	00 00 02 06	Error code signifies: Index does not exist on device

8.5.1. Error Codes

The following error codes come from the CANopen Communication Profile V4.00. The abort codes refer to 2 fundamental error causes or error locations.

On the one hand, an SDO can contain data which are not acceptable to the addressed Slave in type, format or content. On the other hand, the Slave can be in a device state in which it is momentarily not able or permitted to carry out certain instructions. The error code always originates from the CANopen Slave.

Error code	Description
0503 0000	Toggle bit not alternated
0504 0000	SDO protocol timed out
0504 0001	Client/server command specifier not valid or unknown
0504 0005	Out of memory
0601 0000	Unsupported access to an object
0601 0001	Attempt to read a write only object
0601 0002	Attempt to write a read only object
0602 0000	Object does not exist in the object dictionary
0604 0041	Object cannot be mapped to the PDO
0604 0042	The number and length of the objects would exceed PDO length
0604 0043	General parameter incompatibility reason
0604 0047	General internal incompatibility in the device
0606 0000	Access failed due to a hardware error
0607 0010	Data type does not match, length of service parameter does not match
0607 0012	Data type does not match, length of service parameter too high
0607 0013	Data type does not match, length of service parameter too low
0609 0011	Subindex does not exist
0609 0030	Value range of parameter exceeded (only for write access)
0609 0031	Value of parameter written too high
0609 0032	Value of parameter written too low

Blank page

9. Annex

9.1. Environmental Protection

9.1.1. Emission

When used correctly, our modules do not produce any harmful emissions.

9.1.2. Disposal

At the end of their service life, modules may be returned to the manufacturer against payment of an all-inclusive charge to cover costs. The manufacturer will then arrange for the modules to be recycled.

9.2. Maintenance/Upkeep

Warning ! Do not insert, apply, detach or touch connections while in operation – risk of destruction or malfunction.
Disconnect all incoming power supplies before working on our modules; this also applies to connected peripheral equipment such as externally powered sensors, programming devices, etc.
All ventilation openings must always be kept free of any obstruction.

The modules are maintenance-free when used correctly.
Clean only with a dry, non-fluffing cloth.
Do not use detergents.

9.3. Repairs/Service

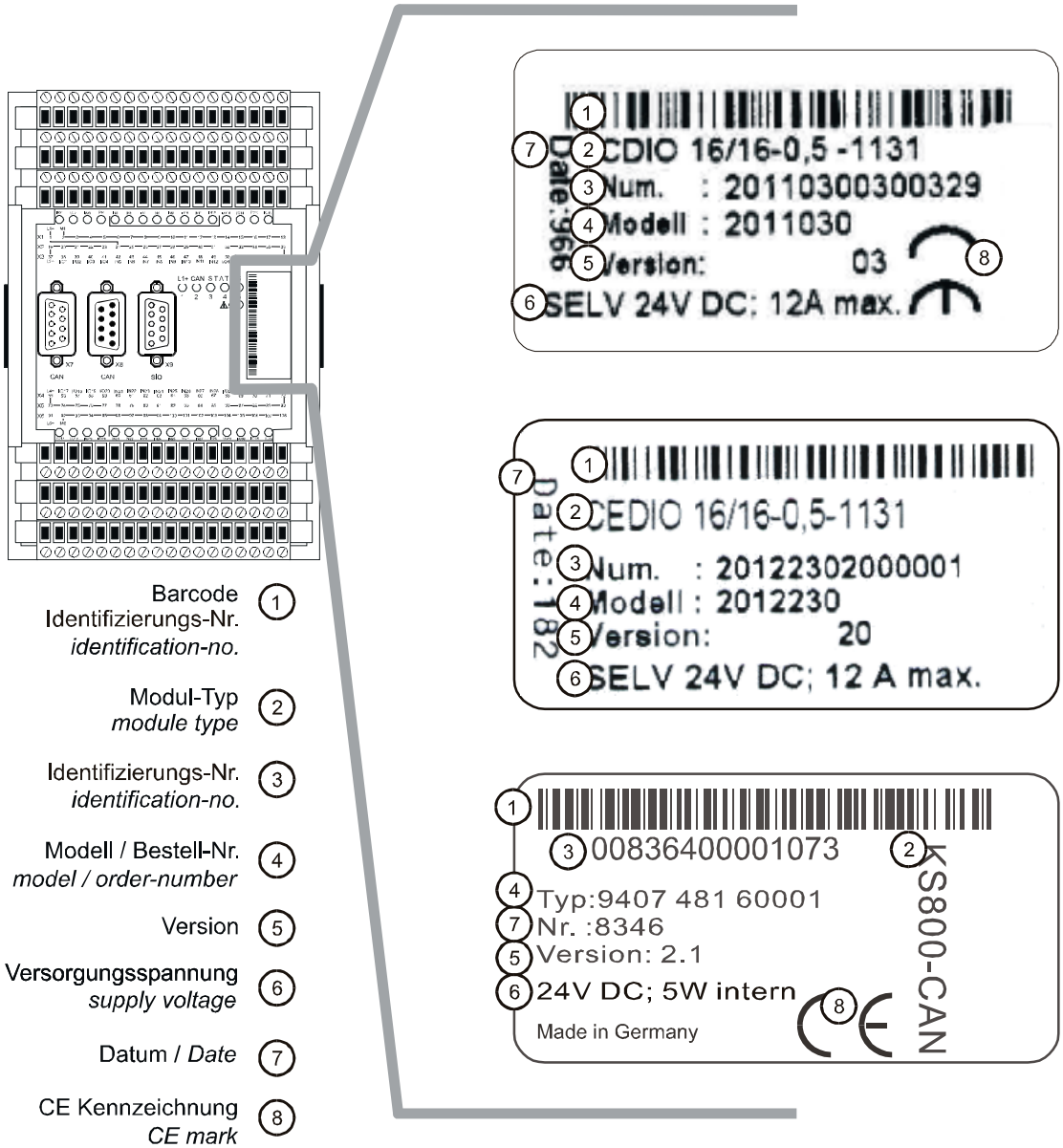
Warning ! Repair work may only be carried out by the manufacturer or its authorised service engineers.

9.3.1. Warranty

Sold under statutory warranty conditions. Warranty lapses in the event of unauthorised attempts to repair the equipment and/or product, or in the event of any other form of intervention.

9.4. Nameplate

Erklärungen zu den Typenschildern (Beispiel)
nameplate descriptions (example)



1 Barcode Identifizierungs-Nr. *identification-no.*
2 Modul-Typ *module type*
3 Identifizierungs-Nr. *identification-no.*
4 Modell / Bestell-Nr. *model / order-number*
5 Version
6 Versorgungsspannung *supply voltage*
7 Datum / *Date*
8 CE Kennzeichnung *CE mark*

2VF100080DG01.cdr

① **Barcode**
same as identification number.

② **Module type**
plain-text name of module.

③ **Identification no.**
module's identification number.

④ **Model/order no.**
You only need to give this number when ordering a module. The module will be supplied in its current hardware and software version.

⑤ **Version**
defines the design-level of the module as supplied ex-works.

⑥ **Supply voltage**

⑦ **Date**
internal code.

⑧ **CE mark**

Note:
The 'Version' (supply version) panel specifies the design-level of the module as supplied ex-works. When replacing a module, users, with the CNW (Cantrol Node Wizard) tool, can read off the current software version of the newly supplied module, and then reload their 'own' software version for a particular project if necessary.
With the latter in mind, before the download you should always keep a record of the existing software levels in your project documentation (software version, node IDs, baud rate, etc.)

9.5. Addresses and Bibliography

9.5.1. Addresses

CiA

'CAN in Automation', international manufacturers and users organisation for CAN users in the field of automation:

CiA - CAN in Automation e.V.
Am Weichselgarten 26
D-91058 Erlangen /Germany
e-mail: headquarters@can-cia.de
http://www.can-cia.de

DIN-EN Standards

Beuth Verlag GmbH
10772 Berlin

or

VDE-Verlag GmbH
10625 Berlin

IEC Standards

VDE Verlag GmbH
10625 Berlin

or

Internet search
http://www.iec.ch/

9.5.2. Standards/Bibliography

IEC61131-1/EN61131-1	Programmable controllers Part 1: General information
IEC61131-2/EN61131-2	Programmable controllers Part 2: Equipment requirements and tests
IEC61131-3/EN61131-3	Programmable controllers Part 3: Programming languages
IEC61131-4/EN61131Bl1	Programmable logic controllers Supplementary Sheet 1: User guidelines
EN 50081 Parts 1+2	German EMC Act: Emitted interference
EN 50082 Parts 1+2	German EMC Act: Noise immunity
ISO/DIS 11898	Draft International Standard: Road vehicles - Interchange of digital information - **C**ontroller **A**rea **N**etwork (CAN) for high-speed communication
EN 954-1	Safety of machinery: Safety-related parts of control systems (Part 1)
Bibliography	A variety of specialist publications on the CANbus is available from specialist bookshops, or can be obtained through the CiA users' organisation.

Note:
Our Technical Support team will be glad to provide other literature references on request.