NZM-XDMI612
CANopen Interface

User Manual

05/04 AWB1230-1543GB

MOELLER

Think future. Switch to green.

1st published 2004, edition date 05/04

Author: Lothar Jagusch
Editor: Heidrun Riege
Translator: David Long

Warning!
Dangerous electrical voltage!

Before commencing the installation

- Disconnect the power supply of the device.
- Ensure that devices cannot be accidentally restarted.
- Verify isolation from the supply.
- Earth and short circuit.
- Cover or enclose neighbouring units that are live.
- Follow the engineering instructions (AWA) of the device concerned.
- Only suitably qualified personnel in accordance with EN 50110-1/-2 (VDE 0105 Part 100) may work on this device/system.
- Before installation and before touching the device ensure that you are free of electrostatic charge.
- The functional earth (FE) must be connected to the protective earth (PE) or to the potential equalisation. The system installer is responsible for implementing this connection.
- Connecting cables and signal lines should be installed so that inductive or capacitive interference does not impair the automation functions.
- Install automation devices and related operating elements in such a way that they are well protected against unintentional operation.
- Suitable safety hardware and software measures should be implemented for the I/O interface so that a line or wire breakage on the signal side does not result in undefined states in the automation devices.
- Ensure a reliable electrical isolation of the low voltage for the 24 volt supply. Only use power supply units complying with IEC 60364-4-41 (VDE 0100 Part 410) or HD 384.4.41 S2.
- Deviations of the mains voltage from the rated value must not exceed the tolerance limits given in the specifications, otherwise this may cause malfunction and dangerous operation.
- Emergency stop devices complying with IEC/EN 60204-1 must be effective in all operating modes of the automation devices. Unlatching the emergency-stop devices must not cause restart.
- Devices that are designed for mounting in housings or control cabinets must only be operated and controlled after they have been installed with the housing closed. Desktop or portable units must only be operated and controlled in enclosed housings.
- Measures should be taken to ensure the proper restart of programs interrupted after a voltage dip or failure. This should not cause dangerous operating states even for a short time. If necessary, emergency-stop devices should be implemented.
- Wherever faults in the automation system may cause damage to persons or property, external measures must be implemented to ensure a safe operating state in the event of a fault or malfunction (for example, by means of separate limit switches, mechanical interlocks etc.).

Contents

About this Manual

Target group

This manual has been produced for automation technicians and engineers. A thorough knowledge of the CANopen fieldbus and the programming of a CANopen master is required.

Additional manuals

The following operating manuals should be followed:

- "Circuit-Breaker Communication System" (AWB1230-1441GB),
- "EASY221-CO, CANopen Slave Interface" (AWB2528-1479GB).

All manuals are available on the Internet for download as PDF files. For a fast search enter the documentation number as the search criterion at http://www.moeller.net/support:

Writing conventions

Except for the first page of chapters and empty pages at the end, the top left of the page shows the chapter title and the top right of the page shows the current section for greater clarity.

Abbreviations and symbols used in this manual have the following meanings:

DMI	**D**ata **M**anagement **I**nterface
NZM	Circuit-breaker in general
AE	Distribution circuit protection **E**lectronic
AEF	Distribution circuit protection **E**lectronic, **F**ixed setting
ME	**M**otor protection **E**lectronic
SE	**S**hort circuit protection **E**lectronic
VE	Full protection **E**lectronic
VEF	Full protection **E**lectronic, **F**ixed setting

▶ indicates actions to be taken.

Indicates interesting tips and additional information.

Caution!
Warns of a danger of minor damage.

1 CANopen Interface

This manual describes the exchange of data between the NZM circuit-breaker with DMI (NZM-XDMI612) and a CANopen system. The interface that is used is the EASY221-CO communication module, which provides several functions specifically for operation with the DMI.

The description in this manual is restricted to the functions that are typical for power circuit-breakers. Detailed information on EASY221-CO can be found in → section "Additional manuals", page 3.

System overview

The EASY204-CO communication module was developed for automation tasks that use the CANopen fieldbus.

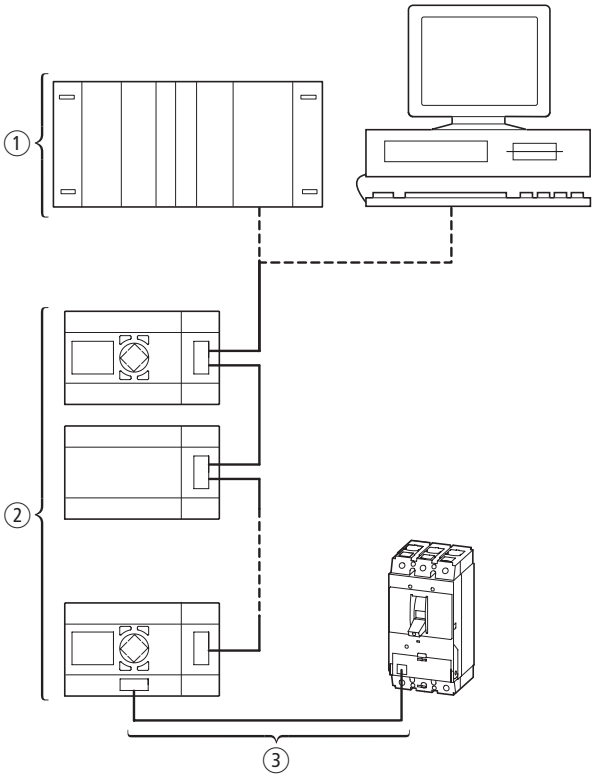


Figure 1: Integration of EASY221-CO in the CANopen network

① Master area, PLC (e.g.: XC600) or PC with CAN card

② Slave area, e.g. DMI or easy control relay with CANopen interface

③ Circuit-breaker area

Scope of functions

EASY221-CO supports the communication objects Service Data Object (SDO) and Process Data Object (PDO) of the CANopen pre-defined connection set.

EASY221-CO facilitates the following functions in conjunction with DMI:

- Reading of the NZM status and DMI inputs through PDO services
  → section "Read NZM status and DMI inputs", page 9,
- Setting of the DMI outputs through PDO services
  → section "Setting the DMI outputs", page 12,
- Access to all the data of the circuit-breaker and the DMI through SDO services
  → section "Data access", page 14.

Commissioning the CANopen interface

Detailed information on the installation and assembly of the EASY221-CO can be found → AWB2528-1479GB.

The following steps must be completed in order to commission the EASY221-CO interface:

- ▶ Use the EASY-Link connection plug to connect the EASY221-CO to the DMI (basic unit).
  To plug the connector into the DMI, push up the cover flap on the right side of the housing.
- ▶ Connect up the 24 V DC power supply for the EASY221-CO interface (**Do not switch on yet!**).
- ▶ Connect up the CANopen connector to the EASY221-CO. The pin assignments can be found → AWB2528-1479GB.
- ▶ Check that all connections are correct.
- ▶ Switch on the power supply for the DMI and the CANopen interface.
- ▶ Set the required bus node address via the display and the keypad on the DMI.
  The procedure can be found → AWB1230-1441GB.
- ▶ Configure the CANopen system, with the EASY221-CO as a new node.
  (Follow the procedure described in AWB2528-1479GB.)
- ▶ Start the CANopen system.

After these steps, the CANopen should be in the "Pre-operational" status (RUN-LED flashes) or "Operational" (RUN-LED lights up continuously).

Further explanation of the displays and operating states can be found → AWB2528-1479GB, Chapter 4.

Communication objects

Process Data Objects PDO

Process data is exchanged in the CANopen by means of PDOs (Process Data Objects).

The PDOs can be used to call up the most important data of the circuit-breaker or to alter the DMI outputs. The individual procedures are described below.

→ Before data can be read or set through a PDO, the CANopen interface must first be put into the "Operational" status. This is done through the NMT Service "Start Remote Node" (Command Specifier = 1). Detailed information can be found → AWB2528-1479GB, Section "Network Management".

Read NZM status and DMI inputs

The Rx-PDO channel 1 is used to read the status information. The data are transmitted from the module every time the status changes. As an alternative, the data can be explicitly requested by a Remote Telegram.

3 data bytes are transmitted, which are arranged as follows:

Byte 0:

Meaning	Bit							
	7	6	5	4	3	2	1	0
not used	–							
Input 5		1/0						
Input 4			1/0					
not used				–				
not used					–			
not used						–		
not used							–	
not used								–

Byte 1:

Meaning	Bit							
	7	6	5	4	3	2	1	0
Overload 1	1/0							
Input 2		1/0						
Input 3[1)]/clockwise[2)]			1/0					
Ready to switch on				1/0				
NZM status					1/0	1/0		
NZM position							1/0	1/0

1) Without motor starter function.
2) With reversing starter.

Byte 2:

Meaning	Bit							
	7	6	5	4	3	2	1	0
Load warning	1/0							
Cause of tripping		1/0	1/0	1/0				
Input 0					1/0			
Reserved						0		
Central warning							1/0	
Input 1[1)]/anti-clockwise[2)] or ON/OFF[3)]								1/0

Inputs 0 to 3 of the DMI can be assigned to 24 V signals as required.
1) Without motor starter function.
2) With reversing starter.
3) With DOL starter.

Meaning of data

Cause of tripping	Indicates the cause of the trip present at the moment.	
	000	No trip
	001	Trip I_r
	010	Trip I_i
	011	Trip I_{sd}
	100	Trip I_{dn}
	101	Trip Com or Trip Temp (detail can be read out through diagnosis)
	110	Overcurrent on neutral pole
Ready to switch on	The circuit breaker is switched off (not tripped!)	
Inputs 0 to 5	Status of the digital inputs on the DMI	
Load warning	At least one phase current is over the load warning limit (I_r = 70 %).	
Reserved	Used for internal purposes. Must **not** be used by the user!	
Central warning	Set if the circuit-breaker reported either a warning or a trip.	
NZM status	Indicates the actual status of the circuit-breaker	
	00	Initialisation in progress
	01	Circuit-breaker is OFF[1)]
	10	Circuit-breaker is ON[1)]
	11	Circuit-breaker has tripped[1)]
NZM position	01	Normal operation of the circuit-breaker
	11	Circuit-breaker faulty or not present
Overload 1	At least one phase current is above the overload 1 limit (I_r = 100 %).	

1) Precondition: auxiliary contacts are connnected to the NZM.

Setting the DMI outputs

The Tx-PDO channel 1 is used to set the outputs and operate the motor starter functions. The corresponding data can be used for direct setting of the DMI outputs.

3 data bytes are transmitted, which are arranged as follows:

Byte 0: always 14_{hex}.

Byte 1:

Meaning	Bit							
	7	6	5	4	3	2	1	0
Output 4	1/0							
Output 3/reserved[1)]		1/0						
Output 2/reserved[2)]			1/0					
Output 1/anti-clockwise[3)]				1/0				
Output 0[4)]/ON[5)]/clockwise[5)]					1/0			
Reserved						0		
Actuate circuit-breaker							1/0	1/0

1) With reversing star/delta starter.
2) With star/delta DOL starter.
3) With reversing starter.
4) Without motor starter function.
5) With DOL starter.

Byte 2:

Meaning	Bit							
	7	6	5	4	3	2	1	0
Remote operator mode	1/0							
Reserved		1/0						
Output 5			1/0					
Reserved				0				
Reserved					0			
Reserved						0		
Reserved							0	
Reserved								0

Outputs 0 to 3 of the DMI can be written according to the motor starter function that has been selected.

Outputs 4 and 5 can either be set as freely addressable (Remote operator mode = 1), or controlled via "Actuate circuit-breaker" (Remote operator mode = 0).

Meaning of data

Actuate circuit-breaker	The circuit-breaker is switched on or off via a motor operator; Requirement: the "Motor mode" bit is set to 0	
	00	Do not change status
	01	Switch off (Output 4 = 1, Output 5 = 0)
	10	Switch on (Output 4 = 0, Output 5 = 1)
	11	Do not change status
Remote operator mode	Changes the use of the outputs 4 and 5	
	0	Use Outputs 4 and 5 for remote operator (see above)
	1	Outputs 4 and 5 are set directly by the corresponding bits
Outputs 0 to 5	Digital outputs of the DMI	
Inputs 0 to 5	Status of the digital inputs on the DMI	
Reserved	Not evaluated	

Service Data Objects SDO

Service Data Objects are used for read/write access to the entries of the object dictionary.

You can use an SDO to read out or alter all the data of the circuit-breaker or the DMI.

Data access
The manufacturer-specific object "Command" (index 2021_{hex}, subindex 00_{hex}) is used for access to the NZM or DMI data, with an additional command code.

This command code defines whether read or write access is to be performed, to which DMI address it applies (→ chapter "Data", from page 23) and how many bytes it contains.

Table 1: Command codes for SDO access to the DMI data memory

Code	Command	Operand				
		0 (2 bytes)	1	2	3	4
61_{hex}	Read 1 byte	Address in DMI data-area → page 23	0	0	0	0
62_{hex}	Read 2 bytes		0	0	0	0
63_{hex}	Read 3 bytes		0	0	0	0
64_{hex}	Read 4 bytes		0	0	0	0
69_{hex}	Write 1 byte		Data byte 1	0	0	0
$6A_{hex}$	Write 2 bytes			Data byte 2	0	0
$6B_{hex}$	Write 3 bytes				Data byte 3	0
$6C_{hex}$	Write 4 bytes					Data byte 4

For more than 4 data bytes, both read and write access can be carried out by segmented transfer.

Expedited transfer is available for writing individual bytes.

Caution!
easy-specific objects and command codes must **not** be used for operation with the DMI. The only permissible objects → following table 2.

Table 2: Valid obects for DMI

Object	Index hex	Valid for ...
Communication parameters	1000 to 100A, 100C, 100D, 1010, 1011, 1014, 1015, 1017, 1018, 1200, 1400, 1600, 1800, 1A00	all objects of EASY221-CO
Manufacturer-specific objects	2001	
Input data	2011	input/output data with coding → page 9 and 12 (different from EASY221-CO)
Output data	2012	
Command to DMI	2021	DMI data: read/ write[1)]
Response from DMI	2022	

1) Only the above.mentioned read and command codes are permissible. Access to easy data such as counter and analog values will lead to malfunction.

→ The detailed description of the individual objects can be found → AWB2528-1479GB, Chapter 5.

SDO access procedure

Write data

Write using segmented transfer
Segmented transfer can be used to write up to 4 bytes simultaneously to the EASY221-CO. The transfer is made in the following steps:

1. Initiate SDO download
The segmented transfer is initiated. The index and subindex for the "Command" object are transferred to the EASY221-CO, which takes on the server role for this communication link.

Telegram content

Index:	2021_{hex}
Subindex:	00_{hex}
Data bytes 1 to 4:	00

2. Download SDO segment
The segmented transfer is concluded. A maximum of 7 data bytes will be available. These are filled by the command code and the operands from table 1, page 15 :

1st data byte:	Command code (69 to $6C_{hex}$)
2nd data byte:	DMI address, H byte (always 0)
3rd data byte:	DMI address, L byte
Data bytes 4 to 7:	Data (= operand 1 to 4)

This concludes the write procedure.

For the data addresses and interpretation → chapter "Data", from page 23.

Write using expedited transfer
A special case is a write action using expedited transfer. This procedure can be applied if only one byte has to be transferred. The telegram is assembled as follows:

Index:	2021_{hex}
Subindex:	00_{hex}
1st data byte:	Command code 69_{hex} (= write 1 byte)
2nd data byte:	00_{hex} (DMI address, H byte (always 0); → chapter "Data", page 23)
3rd data byte:	DMI address, L byte
4th data byte:	Value to be written

Expedited transfer is particularly suitable for sending commands to the DMI or NZM, or altering individual parameters.

Read data

Here the procedure is more complicated than for writing. The following steps must be observed:

1. Request data (segmented transfer),
2. Check that the data are ready for transfer (expedited transfer),
3. Read out the data (segmented transfer).

Request data
Initiate SDO download
To request data you need the following information:

- the address from where the data are to be read out,
- the number of bytes to be read,
- the statement that this is a read access.

First of all, the segmented transfer is initated. The index and subindex for the "Command" object are transferred to the EASY221-CO.

Telegram content

Index:	2021_{hex}
Subindex:	00_{hex}
Data bytes 1 to 4:	00

Download SDO segment

The transfer is now concluded. A maximum of 7 data bytes will be available. These are filled by the command code and the operands from table 1, page 15 .

1st data byte:	Command code (61 to 64_{hex})
2nd data byte:	DMI address, H byte (always 0)
3rd data byte:	DMI address, L byte
Data bytes 4 to 7:	0

Check that the data are available

The check is performed by reading the status (index 2020_{hex}, subindex 00_{hex}). This check reading can be made using expedited transfer, since the volume of data is only 1 byte.

The requested data become available when the status goes to the value "00".

Read out data

Initiate SDO upload

First of all, request the response to the previously performed read command. To do this, segmented transfer is initiated, and the index and subindex for the "Response" object are sent to the EASY221-CO.

Telegram content

Index:	2022_{hex}
Subindex:	00_{hex}
Data bytes 1 to 4:	00

Upload SDO segment

The previously requested data are now read out from the master. The master concludes the transfer, without sending any data itself. All the data bytes remain at "0".

Byte 1 to byte 7: 0

The module reacts to this by returning the requested address with the data that are present in the response telegram from byte 4 :

Byte 2: DMI address, H byte (always 0)

Byte 3: DMI address, L byte

Byte 4: Data byte 0

Byte 5: Data byte 1, if requested, otherwise "00"

Byte 6: Data byte 2, if requested, otherwise "00"

Byte 7: Data byte 3, if requested, otherwise "00"

After this response, the read cycle is finished.

→ For the data addresses and interpretation → chapter "Data", from page 23.

Updating of process data

For SDO access, data of the CANopen master are passed on immediately from EASY221-CO to the DMI or the circuit-breaker.

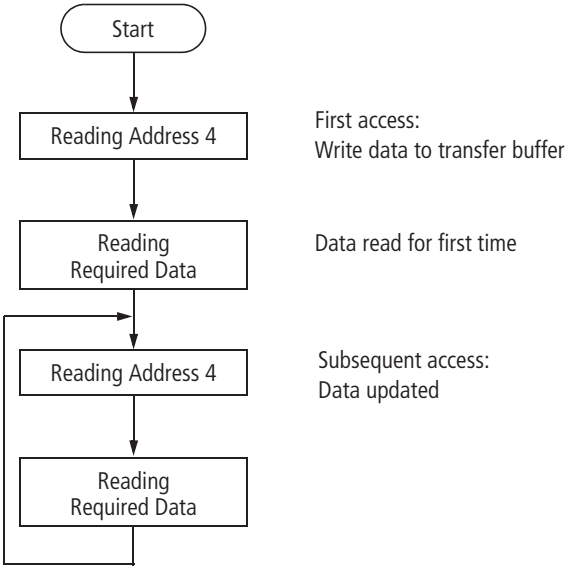
In the other direction, from DMI to the CANopen master, the circuit-breaker amd DMI data are buffered, so that there is always a self-consistent data set available. This data set is updated as soon as an access is made to address 4 (→ table 3, page 23).

When starting the SDO transfer, it is advisable to access address 4 first, so that the latest data are fed into the buffer. Data can then be read out at will.

After all the required data have been read out, access to address 4 initiates a new cycle.

→ To update the data, read-access to address 4 is always required.

The flowchart below illustrates the basic read sequence.



Read diagnosis data from the circuit-breaker

Diagnosis data represent a special case. In this case, several sets of data share one address area.

First of all, you must use a command to select a data set for transmission (→ section "Circuit-breaker-specific commands"page 22). Then you can read out this set of diagnosis data.

To carry on reading out the latest process data, you must use the command code "1A" to leave the diagnosis mode. Later, you can restart the procedure by selecting the new diagnosis data set or read out the process data.

→ Process data will not be updated as long as the diagnosis mode is active.

Circuit-breaker-specific commands

Address "00" in the memory area of the DMI can be used to transfer various NZM-specific commands. To do this, the required command code is written to this address, using expedited transfer, for example. The command codes can be found → table 5, page 31. They cover:

- commands for remote operator functions,
- commands for operating motor starter functions,
- acknowledgements for certain messages,
- the trip command for the NZM, and
- the control of diagnosis data.

2 Data

Data with a width exceeding 1 byte are stored in the following form:

- Low byte at the least significant address
- High byte at the most significant address

Reading circuit-breaker data

Table 3: Address location of the input data

Address	Data type	Data item	Value	Unit/ reference
0	Communi-cation	Command	→ page 31	–
1		Reserved	–	
2		Circuit-breaker communication status, byte 0	→ page 33	
3		Circuit-breaker communication status, byte 1		
4		DMI ID	5 (fixed)	
5	Circuit-breaker status	Circuit-breaker status 0	→ page 34	–
6		Circuit-breaker status 1		
7		Circuit-breaker status 2		
8		Circuit-breaker status 3		
9		Circuit-breaker status 4		
10		Reserved	–	
11		Reserved	–	
12	Module status	DMI inputs	00 to $0x3F_{hex}$	–
13		Reserved	–	
14		Outputs	00 to $0x3F_{hex}$	
15		Reserved	–	

Address	Data type	Data item	Value	Unit/ reference
16	Currents	I_{eff1}: effective phase current L1	0 to 65 535	A
17				
18		I_{eff2}: effective phase current L2, in relation to I_r		
19				
20		I_{eff3}: effective phase current L3, in relation to I_r		
21				
22		I_{effN}: effective neutral pole current, in relation to I_r		
23				
24		Maximum current for all phases		
25				
26		$I_{dn\ eff}$: effective fault current		
27				
28		Reserved	–	–
29	Current parameters,	I_r	→ page 36	–
30		I_i	→ page 37 and → page 38	
31		T_r	→ page 40	
32		I_{sd}	→ page 39	
33		T_{sd}	→ page 40	
34		I_{dn}	→ page 36	
35		T_{vdn}	→ page 40	
36		Function switch	0 = I^2t OFF 1 = I^2t ON	
37	Time, date	Second	0 to 59	Sec.
38		Minute	0 to 59	Min.
39		Hour	0 to 23	Hr
40		Day	1 to 31	Day
41		Month	1 to 12	Month
42		Year (from 2000)	0 to 255	Year

Address	Data type	Data item	Value	Unit/ reference
43	Device	Serial number LSB	0 to 2^{24}-1	–
44				
45		Serial number MSB		
46		Circuit-breaker type	→ page 41	
47		Circuit-breaker function	→ page 42 (IEC, (UL/CSA)	
48		I_{nom} (only for UL/CSA devices)	0 to 1600	
49				
50		NZM software version	→ page 43	
51		Hardware module (not supported)	–	
52		Software module[1)]	1.0.0 to 3.7.7	
66	Device settings	I_r	→ page 36	
67		I_i	→ page 37 and Page 38	
68		T_r	→ page 40	
69		I_{sd}	→ page 39	
70		T_{sd}	→ page 40	
71		I_{dn}	→ page 36	
72		T_{vdn}	→ page 40	
73		Function switch	0 = I^2t OFF 1 = I^2t ON	
74	DMI mode	Output Q0 Mode	→ page 44	–
75		Output Q1 Mode		
76		Output Q2 Mode		
77		Output Q3 Mode		
78		Output Q4 Mode		
79		Output Q5 Mode		

Address	Data type	Data item	Value	Unit/ reference
80	DMI mode	Switch element	1 = pushbutton, 0 = switch	–
81		Changeover t	0 to 99900 (in steps of 100)	ms
82				
83	Display mode	Display line 1	→ page 45	–
84		Display line 2		
85		Display line 3		
86		Display line 4		
87		Display line 5		
88		Display line 6		
89	DMI parameters	Password	0 to 9999	–
90				
91		Language	→ page 46	
92		Reserved	–	–
93		Stop serial number LSB	NZM serial number for starting-stop test	0 to 2^{24}-1
94				
95		Stop serial number MSB		
96		DMI options	→ page 46	–
97		Software DMI, main index	1 to 9	
98		Software DMI, aux. index	1 to 9	
99		Software DMI, subindex	0 to 9	

Address	Data type	Data item	Value	Unit/ reference
100	Operating hours counters	Hours of operation NZM LSB	0 to 2^{32}-1	Hours
101				
102				
103		Hours of operation NZM MSB		
104		Hours of operation DMI LSB	0 to 2^{32}-1	
105				
106				
107		Hours of operation NZM MSB		
108	Diagnostics	Status byte 0	→ page 34	–
109		Status byte 1		
110		Status byte 2		
111		Status byte 3		
112		Status byte 4		
113		Reserved	–	
114		I_r	→ page 36	
115		I_i	→ page 37 and Page 38	
116	Diagnostics	T_r	→ page 40	–
117		I_{sd}	→ page 39	
118		T_{sd}	→ page 40	
119		I_{dn}	→ page 36	
120		T_{vdn}	→ page 40	
121		Function switch, Data set No.	→ page 46	
122		Second	0 to 59	Sec.
123		Minute	0 to 59	Min.
124		Hour	0 to 23	Hr
125		Day	1 to 31	Day

Address	Data type	Data item	Value	Unit/ reference
126	Diagnostics	Month	1 to 12	Month
127		Year (from 2000)	4 to 255	Year
128		Operations	0 to 65535	–
129				

1) If the code for a motor starter function is entered at this location, then it simultaneously defines the operating mode of the DMI. For instance, code 18 stands for "reversing starter"

Writing circuit-breaker data

Table 4: Address location of the output data

Address	Data type	Data item	Value	Unit/ reference
0	Communication	Command	→ page 31	–
1		Reserved	–	
2	Parameters	I_r	→ page 36	–
3		I_i	→ page 37 and Page 38	
4		T_r	→ page 40	
5		I_{sd}	→ page 39	
6		T_{sd}	→ page 40	
7		I_{dn}	→ page 36	
8		T_{vdn}	→ page 40	
9		Function switch	0 = I^2t OFF 1 = I^2t ON	
10	Time, date	Second	0 to 59	Sec.
11		Minute	0 to 59	Min.
12		Hour	0 to 23	Hr
13		Day	1 to 31	Day
14		Month	1 to 12	Month
15		Year (from 2000)	0 to 255	Year
16	Outputs	Digital outputs of the DMI	00 to 0x3F$_{hex}$	–

Address	Data type	Data item	Value	Unit/ reference
17	DMI mode	Output Q0 Mode[1)]	→ page 44	–
18		Output Q1 Mode		
19		Output Q2 Mode		
20		Output Q3 Mode		
21		Output Q4 Mode		
22		Output Q5 Mode		
23		Switch element	1 = pushbutton, 0 = switch	
24		Changeover t	0 to 999	100 ms
25				
26	Display mode	Display line 1	→ page 45	–
27		Display line 2		
28		Display line 3		
29		Display line 4		
30		Display line 5		
31		Display line 6		
32	DMI parameters	Password	0 to 99900	–
33				
34		Language	→ page 46	–
35		Reserved	–	–
36		Stop serial number LSB	NZM serial number for starting-stop test	0 to 2^{24}-1
37				
38		Stop serial number MSB		
39		DMIOpt	→ page 46	–

1) If a value that stands for a motor starter function is written at this location, then this will also define the operating mode of the DMI. For instance, the value 18 will switch the DMI into the reversing starter mode.

Circuit-breaker-specific commands

Circuit-breaker-specific commands can also be sent via address 0 of the DMI data array (→ table 3, Page 23 and Table 4, Page 29). The following commands are available:

Table 5: Circuit-breaker commands via PROFIBUS-DP

Code	Command	Function
04	Switch on NZM	Switch on NZM remotely (Q4/Q5 on DMI)
05	Switch off NZM	Switch off NZM remotely (Q4/Q5 on DMI)
06	Switch off motor	Switch off motor Precondition: motor starter function is selected on the DMI
07	Switch on motor, anti-clockwise	Switch on motor (direct starter) or start anti-clockwise (reversing starter) Precondition: motor starter function is selected on the DMI
08	Motor rotates clockwise	Start clockwise Precondition: motor starter function is selected on the DMI
09	Acknowledge trip	Acknowledges an NZM trip that has occurred
10_{hex}	Read diagnostics data set 0 [2)] (= most recent diagnostics data)	Reads the data of the most recently stored (= latest) diagnostics event
11_{hex}	Read diagnostics data set 1[2)]	Reads the data of the older diagnostics event that is stored
12_{hex}	Read diagnostics data set 2[2)]	
13_{hex}	Read diagnostics data set 3[2)]	
14_{hex}	Read diagnostics data set 4[2)]	
15_{hex}	Read diagnostics data set 5[2)]	

Code	Command	Function
16_{hex}	Read diagnostics data set 6[2)]	Reads the data of the older diagnostics event that is stored
17_{hex}	Read diagnostics data set 7[2)]	
18_{hex}	Read diagnostics data set 8[2)]	
19_{hex}	Read diagnostics data set 9[2)]	Reads the data of the oldest diagnostics event that is stored
22_{hex}[1)]	Acknowledge startup stop	When startup stop is active, this acknowledges the serial number of the NZM and thereby accepts it as a newly valid serial number for checking the startup stop
35_{hex}[1)]	Trip	Forces the NZM to trip
$1A_{hex}$	Exit Diagnostics mode	Returns to the transfer of process data

1) **Caution!** These commands intervene directly in the functioning of the NZM!
2) In Diagnostics mode, only the selected diagnostics data and the latest parameter settings are transferred. The most recent process data is only visible when Diagnostics mode has been exited.

Caution!
Only the codes listed here may be used.

The command code is written to cell 0 by the master. During the read operation, the master also receives the same value back in cell 0 until the command has been recognised, and the inverted value until the command has been executed. The contents of these cells are then reset to 0.

Circuit-breaker communication status

Table 6: Circuit-breaker communication status

Byte 0	
Bit 0	1 if circuit-breaker detected with new serial number and the Stop mode is activated. Otherwise 0
Bit 1	1 if parameters of DMI and LS are not identical Otherwise 0
Bit 2	1 if parameters set for motor protection function, but no NZM-ME type is connected otherwise 0
Bit 3 and bit 4	Reserved
Bit 5	1 after initialisation of the interface, until an LS ID (01_{hex}) or PC ID is detected while establishing the connection. Otherwise 0
Bit 6	1 after initialisation of the interface, until the ID block ($1E_{hex}$) is correctly received when making the connection. Otherwise 0
Bit 7	1 after initialisation of the interface, until the first transfer command ($3F_{hex}$) is received while making the connection.
Byte 1:	
Bit 0 to bit 3	Number of the last data record entered in the diagnostics data (0 to 9)
Bit 4	"0" DMI has received latest operating data from NZM "1" DMI has received latest diagnostics data from NZM
Bit 5 to Bit 7	Reserved

The following applies to all status bits:

- 1 = Condition fulfilled
- 0 = Condition not fulfilled

Table 7: Circuit-breaker status

Byte 0	**Trip conditions**
Bit 0	Trip I_i: Short-circuit
Bit 1	Trip I^2t (if I^2t function selected): Overload, variable, short-time delayed Trip I_{sd} (if I^2t function not selected): Overload constant, short-time delayed
Bit 2	Trip I_r: Overload, variable, long-time delayed
Bit 3	Trip I_{dn}: Residual current, constant, short-time delayed
Bit 4	Reserved
Bit 5	Trip Comm: trip via communication occurred
Bit 6	Trip Temp: trip due to device overtemperature (NZM)
Bit 7	Reserved
Byte 1:	**Alarm conditions**
Bit 0	Overload early warning
Bit 1	Overload range 1
Bit 2	Overload range 2
Bit 3	Motor protection active
Bit 4	Unbalance
Bit 5	Reserved
Bit 6	Reserved
Bit 7	Reserved
Byte 2:	**Phase status**
Bit 1 and Bit 0	Status of phase L1
Bit 3 and Bit 2	Status of phase L2
Bit 5 and Bit 4	Status of phase L3
Bit 7 and Bit 6	Status of neutral pole
	Possible phase states
00	Normal range
01	Load early warning
10	Overload range 1
11	Overload range 2

Byte 3:	Tripping phase
Bit 0	Trip due to L1
Bit 1	Trip due to L2
Bit 2	Trip due to L3
Bit 3	Trip due to N
Bit 4	Reserved
Bit 5	Reserved
Bit 6	Reserved
Bit 7	Reserved
Byte 4:	**Phase reporting alarm**
Bit 0	Reserved
Bit 1	Reserved
Bit 2	Reserved
Bit 3	Reserved
Bit 4	from to L1
Bit 5	from L2
Bit 6	from L3
Bit 7	from N

Circuit-breaker parameters

Settings I_r and I_{dn}

Value	I_r [A]	I_{dn} [A]
0	$0.5 \times I_n$	$0.2 \times I_n$
1	$0.55 \times I_n$	$0.3 \times I_n$
2	$0.6 \times I_n$	$0.4 \times I_n$
3	$0.65 \times I_n$	$0.5 \times I_n$
4	$0.7 \times I_n$	$0.6 \times I_n$
5	$0.75 \times I_n$	$0.7 \times I_n$
6	$0.8 \times I_n$	$0.8 \times I_n$
7	$0.85 \times I_n$	$0.9 \times I_n$
8	$0.9 \times I_n$	$1.0 \times I_n$
9	$0.925 \times I_n$	$0.2 \times I_n$
10	$0.95 \times I_n$	–
11	$0.975 \times I_n$	
12	$1.0 \times I_n$	
13	$0.5 \times I_n$	
14	$0.5 \times I_n$	
15	$0.5 \times I_n$	

Settings I_i for ZM2-... and ZM4-...

Value	I_i [A]			
	ZM2-AE ZM2-AEF-NA ZM2-VE ZM2-VE-NA ZM2-VEF-NA	**ZM2-ME ZM2-SE-CNA**	**ZM4-AE ZM4-AE-NA ZM4-AEF-NA ZM4-VE ZM4-VE-NA ZM4-VEF-NA**	**ZM4-ME ZM4-SE-CNA**
0	$2 \times I_n$	$2 \times I_n$	$2 \times I_n$	$2 \times I_n$
1	$3 \times I_n$	$3 \times I_n$	$3 \times I_n$	$3 \times I_n$
2	$4 \times I_n$	$4 \times I_n$	$4 \times I_n$	$4 \times I_n$
3	$5 \times I_n$	$5 \times I_n$	$5 \times I_n$	$5 \times I_n$
4	$6 \times I_n$	$6 \times I_n$	$6 \times I_n$	$6 \times I_n$
5	$7 \times I_n$	$8 \times I_n$	$7 \times I_n$	$8 \times I_n$
6	$8 \times I_n$	$10 \times I_n$	$8 \times I_n$	$10 \times I_n$
7	$10 \times I_n$	$12 \times I_n$	$10 \times I_n$	$12 \times I_n$
8	$12 \times I_n$	$14 \times I_n$	$12 \times I_n$	$14 \times I_n$
9	$2 \times I_n$	$2 \times I_n$	$2 \times I_n$	$2 \times I_n$

Settings I_i für ZM3-...

Value	I_i [A]		
	ZM3-AE-250,-400 ZM3-AE-250,-400-NA ZM3-AEF-250...400-NA ZM3-VE-250,-400 ZM3-VE-250,-400-NA ZM3-VEF-250...400-NA	**ZM3-AE-630 ZM3-AE-600-NA ZM3-AEF-450,-550-NA ZM3-AEF-600-NA ZM3-VE-600,-630-NA ZM3-VEF-450,-550-NA ZM3-VEF-600-NA**	**ZM3-ME-220,-350,-500 ZM3-SE-220,-350-CNA**
0	$2 \times I_n$	$2 \times I_n$	$2 \times I_n$
1	$3 \times I_n$	$2.5 \times I_n$	$3 \times I_n$
2	$4 \times I_n$	$3 \times I_n$	$4 \times I_n$
3	$5 \times I_n$	$3.5 \times I_n$	$5 \times I_n$
4	$6 \times I_n$	$4 \times I_n$	$6 \times I_n$
5	$7 \times I_n$	$5 \times I_n$	$8 \times I_n$
6	$8 \times I_n$	$6 \times I_n$	$10 \times I_n$
7	$9 \times I_n$	$7 \times I_n$	$12 \times I_n$
8	$11 \times I_n$	$8 \times I_n$	$14 \times I_n$
9	$2 \times I_n$	$2 \times I_n$	$2 \times I_n$

Settings I_{sd}

Value	I_{sd} [A]	
	ZM3-VE-630 ZM3-VE-250...400 A-NA ZM3-VEF-250...400 A-NA ZM3-VE-450...600-NA ZM3-VEF-450...600-NA	**All others**
0	$1.5 \times I_r$	$2 \times I_r$
1	$2 \times I_r$	$3 \times I_r$
2	$2.5 \times I_r$	$4 \times I_r$
3	$3 \times I_r$	$5 \times I_r$
4	$3.5 \times I_r$	$6 \times I_r$
5	$4 \times I_r$	$7 \times I_r$
6	$5 \times I_r$	$8 \times I_r$
7	$6 \times I_r$	$9 \times I_r$
8	$7 \times I_r$	$10 \times I_r$
9	$1.5 \times I_r$	$2 \times I_r$

Settings T_r, T_{sd}, T_{vdn}

Value	T_r [s]	T_{sd} [ms]	T_{vdn} [ms]
0	2	0	0
1	4	20	20
2	6	60	60
3	8	100	100
4	10	200	200
5	14	300	300
6	17	500	500
7	20	750	750
8	without bimetal	1000	1000
9	2	0	0

Circuit-breaker identification

Value	Refers to	Coding	Currents [A]	Size
Bit 0 to Bit 3	Trans-formers	0000	63	2
		0001	100	
		0010	160	
		0011	250	
		0100	250	3
		0101	400	
		0110	630	
		0111	630	4
		1000	800	
		1001	1000	
		1010	1250	
		1011	1600	
Bit 4 and Bit 5	Frame size	00	Size 2	–
		01	Size 3	
		11	Size 4	
Bit 6	Reserved	–	–	
Bit 7	Reserved	–	–	

Circuit-breaker functionality

Table 8: IEC devices

	Function	Value	Circuit-breaker type
Bit 0	N conductor	0	60 %
		1	100 %
Bit 1	N overload	0	Without N overload
		1	With N overload
Bit 2	No. of poles	0	3-pole
		1	4-pole
Bit 3 and bit 4	Type	00	NZM...-AE
		01	NZM...-ME
		10	NZM...-VE
		11	Reserved
Bit 5	Device	0	IEC
Bit 6	Reserved	–	–
Bit 7	Reserved	–	–

Table 9: UL/CSA devices

	Value	Circuit-breaker type
Bit 0 to bit 3	0	-VE
	1	-VEF
	2	-AE
	3	-AEF
	4	-SE
Bit 4	Reserved	–
Bit 5	1	UL/CSA
Bit 6	Reserved	–
Bit 7	Reserved	–

Caution!
For UL/CSA devices of types -AEF, -VEF and -SE, the identification byte only provides the type of converter that is used. The rated (nominal) current is entered in plain text, in I_{nom} (bytes 8+9 in data set 0).

The type display is composed of the type code + I_{nom}, e.g. NZM3-AE-600. The value 600 comes from I_{nom}.

This means that the current display must be recalculated according to the relationship between the converter current I_{nom}. The transmitted current values refer to the converter current.

Example:
A converter indicates 630 A as 100 %. The actual current flowing is 630 A. Referred to a module with a 600 A rated current, this is equivalent to 105 %.

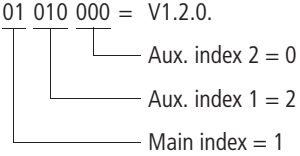
For UL/CSA devices of types -AE and -VE, the nominal current display is derived from the converter identification. The display uses the values → section "Circuit-breaker identification", Page 41.

Version level

Table 10: Coding of the version level

Bit 0 to Bit 2	Aux. index 2
Bit 3 to Bit 5	Aux. index 1
Bit 6 and Bit 7	Main index

Example:



DMI function assignment

Table 11: Function assignment for the DMI outputs

Value	Function
0	Trip I_i
1	Trip I_r
2	Trip I_{sd}
3	Trip I^2t
4	Trip I_{dn}
5	Overtemperature 1
6	Overload 1
7	Overload 2
8	Load warning
9	Asymmetry
10	Parameters
11	Trip
12	Alarm
13	Motor protection active
14	Bus
15	OFF
16	ON
17	Direct starter (only for Q0)
18	Reversing starter (only for Q0, Q1)
19	Star/delta starter (only for Q0, Q2, Q3)
20	Star/delta reversing starter (only for Q0, Q1, Q2, Q3)
21	Remote actuation (= motor drive)

DMI display assignment

Table 12: Display assignment for the DMI display

Value	Display
0	not used
1	I_r
2	I_i
3	I_{sd}
4	I_{dn}
5	T_r
6	T_{sd}
7	T_{vdn}
8	I^2t
9	I_{1eff}
10	I_{2eff}
11	I_{3eff}
12	I_{neff}
13	I_{dneff}
14	motor function (only for the motor starter function with ME types)
15	motor status (only for the motor starter function with ME types)
16	I
17	Q
18	Time
19	Data item

Coding for the language setting

Value	Language
0	German
1	English
2	French
3	Italian
4	Spanish

Coding for the DMI options

Value	Meaning
0	Absolute value display (0 = OFF, 1 = ON)
1	Input 0 for acknowledgement (0 = NO, 1 = YES)
2	Startup stop is activated (0 = NO, 1 = YES)
3	Reserved
4	
5	
6	
7	

Function switch and data set No.

Bit 0	Function switch (0 = I^2t OFF, 1 = I^2t ON)
Bit 1 to Bit 4	Reserved
Bit 5 and Bit 7	Data set No. (0 to 4)