NZM-XDMI612
DeviceNet Interface

User Manual

04/05 AWB1230-1572GB

Moeller

Think future. Switch to green.

1st published 2005, edition date 04/05
© Moeller GmbH, 53105 Bonn

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. Fundamental knowledge concerning the DeviceNet fieldbus and programming of a DeviceNet master control is assumed. Furthermore, you should be familiar with the handling of the easy control relay and the communication system NZM-XDMI612.

Additional manuals

The following operating manuals should be followed:

- “Circuit-Breaker Communication System” (AWB1230-1441GB),
- “EASY222-DN, DeviceNet Slave Interface" (AWB2528-1427GB).

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.

▶ indicates actions to be taken.

→ Indicates interesting tips and additional information

▽ **Caution!**
Warns of a danger of minor damage.

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

AE	Distribution circuit protection **E**lectronic
AEF	Distribution circuit protection **E**lectronic, **F**ixed setting
DMI	**D**ata **M**anagement **I**nterface
HEX	Hexadecimal (Number system based on 16)
ID	**Id**entifier
ME	**M**otor protection **E**lectronic
MS-LED	**M**odule**S**tatus LED
NS-LED	**N**etwork-**S**tatus LED
NZM	Circuit-breaker in general
SE	**S**hort circuit protection **E**lectronic
VE	Full protection **E**lectronic
VEF	Full protection **E**lectronic, **F**ixed setting

1 DeviceNet connection

In this manual the exchange of data between the circuit-breaker NZM with DMI and a DeviceNet system is described. The interface that is used is the EASY222-DN 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 can be found in → section "Additional manuals", page 3.

System overview

The communication module EASY222-DN has been developed for automation tasks with the DeviceNet fieldbus.

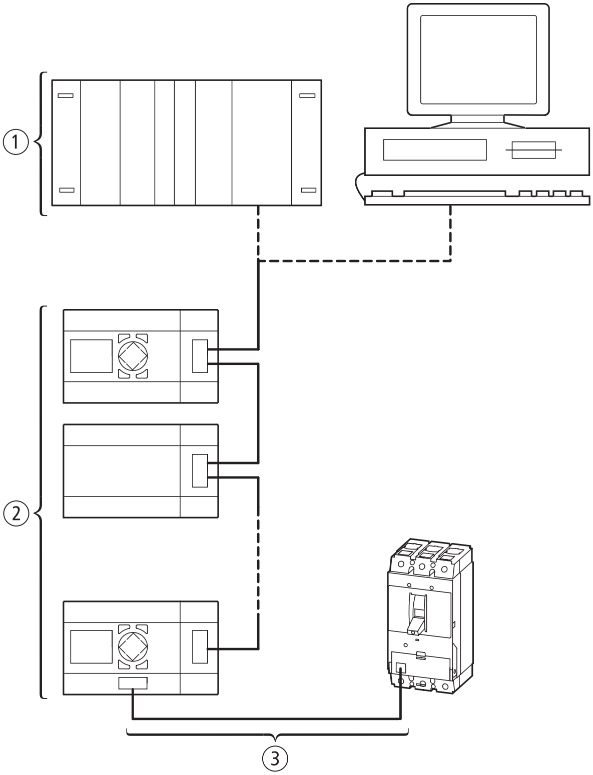


Figure 1: Integration of EASY222-DN in the DeviceNet

1. Master area, PLC (e.g.: SLC 500) or PC with CAN card
2. Slave area e.g.: DMI or control relay easy with DeviceNet connection
3. Circuit-breaker area

Scope of functions

EASY222-DN supports the I/O messages and explicit messages of the predefined master/slave connection setting.

EASY222-DN enables the following functions in conjunction with the DMI:

- Reading of the NZM status and DMI inputs via I/O messages
  → section "Read NZM status and DMI inputs", page 9,
- Setting the DMI outputs via I/O messages
  → section "Setting the DMI outputs", page 12,
- Access to all data of the circuit-breaker and the DMI via explicit messages
  → section "Data access via explicit messages", page 13.

Commissioning the DeviceNet interface

Detailed information concerning the installation and design of the EASY222-DN can be found in → AWB2528-1427GB, sections 1 and 2.

The following steps are necessary for commissioning of the EASY222-DN interface:

- ▶ Connect the EASY222-DN via the "EASY-Link" connector 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 the 24 V DC power supply for the EASY222-DN (**Do not switch it on just yet!**).
- ▶ Connect the 5-pole DeviceNet connector to the EASY222-DN.
  The pin assignments can be found in
  → AWB2528-1427GB.
- ▶ Check that all connections are correct.
- ▶ Switch on the power supply for the DMI and the interface.
- ▶ Set the required bus node address via the display and the keypad on the DMI.
  The procedure can be found in → AWB1230-1441GB.
- ▶ Configure the DeviceNet system with the EASY222-DN as a new station.
- ▶ Start the DeviceNet system.

After these steps the EASY222-DN interface should be in the "not online" (MS-LED lights continuously green, NS-LED is off) or "online without active communication connection" (MS-LED lights continuously green, NS-LED flashes green) state.

→ A complete overview of all LED status displays to assist quick analysis of the problem can be found in → AWB2528-1427GB, section 3.

Supported messages

Data exchange via I/O messages

The exchange of the process data is implemented in DeviceNet with the assistance of I/O messages.

The most important I/O messages can be used to quickly access the most important status data of the circuit-breaker or to change the outputs of the DMI.

→ Before data can be read or set using I/O messages an active communication connection must be established to the DeviceNet interface. Detailed information can be found in → AWB2528-1427GB, Section “Management objects”.

Read NZM status and DMI inputs

The data is output via polled I/O messages.

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)			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 diagnostics)
	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 connected to the NZM.

Setting the DMI outputs

The outputs of the DMI can be controlled directly via I/ O messages.

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)		1/0						
Output 2/reserved2)			1/0					
Output 1/anti-clockwise 3)				1/0				
Output 04)/ON5)/clockwise5)					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		

Meaning	Bit							
	7	6	5	4	3	2	1	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).

Data access via explicit messages

Explicit messages serve reading or writing access to all supported data objects such as

- Management objects,
- Connection objects,
- Application-specific objects.

The management objects (identity object and message router object) and the connection objects (DeviceNet object and connection object) incorporate general services, which are necessary for operation of the device on the DeviceNet.

The application-specific objects (assembly object and easy object) enable application-specific preparation of the data. For communication with the circuit-breaker or with the DMI, a portion of the easy object is used.

→ The detailed description of the individual objects can be found in → AWB2528-1427GB, section 4.

Table 1: Overview of the supported objects

Object	Class ID	Instance ID	Attribute ID	Note
	[hex]	[hex]	[hex]	
Management objects				
Identity Object	01	01	01 to 07, 9.0 A	→ AWB2528-1427
Message Router Object	02	01	–	
Connection object				
DeviceNet Object	03	01	01 to 04	
Connection Object	05	01 to 14	–	
Application-specific objects				
Assembly Object	04	100 to 102	–	
easy Object	64	01	02	
			03	→ NZM status and DMI inputs, page 9
			04	→ DMI outputs, page 12
			61, 62, 69	extended access via command code; service code: 32_{hex}

Caution!
easy-specific objects may **not** be used for operation with the DMI. Only objects from Table 1 are permissible.

In the following only the use of the easy object for scanning and entering the NZM and DMI data is described.

All data of the circuit-breaker or the DMI can be accessed or modified by use of the easy object. The easy object with Class ID: 64_{hex}, Instance ID 01 and Service Code: 32_{hex} is used.

The attribute defines if it is a read access on one or two bytes or a write access on one byte. In the data field the details of the address to which the attribute refers (→ chapter 2, page 21), and if applicable, the byte to be written, are contained.

The following table makes the association clear.

Table 2: Attributes for explicit message accesses to the data memory of the DMI

Attribute	Command	Operand		
		Address H byte	Address L byte	Date
61_{hex}	Read 1 byte	0	Address in the DMI data range→ page 21.	–
62_{hex}	Read 2 bytes			–
69_{hex}	Write 1 byte			Value

Write data

Using the easy object and the command code described in → table 1, the DeviceNet telegram can write a byte to the EASY222-DN. The telegram has the following content:

Class ID:	64_{hex}
Instance ID:	01
Attribute:	69_{hex}
Service Code:	32_{hex} (extended access)
1st or 2nd data byte:	Address
3rd data byte:	Value to be written

The EASY222-DN responds with the value "00" in the telegram data fields.

Read data

Using the easy object and the command code described in → table 1, the DeviceNet telegram can request one or two bytes from the EASY222-DN. The telegram has the following content:

Class ID:	64_{hex}
Instance ID:	01
Attribute:	61_{hex} (1 Byte), 62_{hex} (2 bytes)
Service Code:	32_{hex} (extended access)
1st or 2nd data byte:	Address

The EASY222-DN responds with a telegram, which feeds back the used address in the first two data fields, followed by the data actually requested (1 or 2 bytes, depending on the attribute).

Updating of process data

Data which the DeviceNet master sends is immediately transferred to the DMI or onto the circuit-breaker.

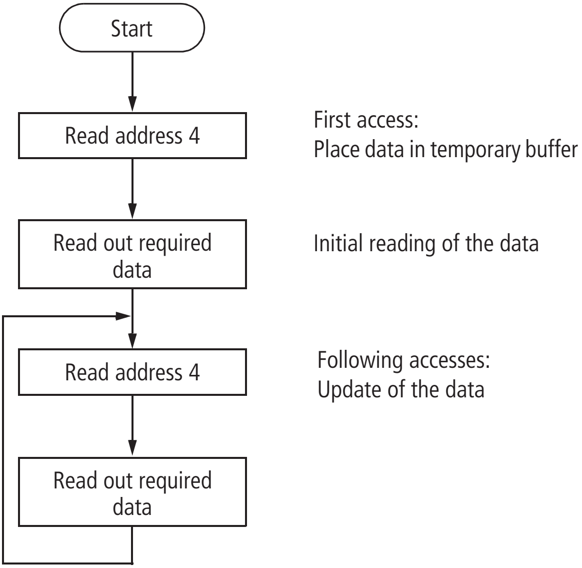
In the reverse direction from the DMI to the DeviceNet master, the data of the circuit-breaker and of the DMI is buffered so that a consistent data set is always available. This data set is updated as soon as an access is made to address 4 (→ table 3, page 21).

During the start of the explicit message connection, it is therefore useful to access address 4 in order to accept the current data 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.

→ A read access to address 4 is always required in order to update the data.

The following diagram indicates the operating procedure with the read access:



Read diagnostics data from the circuit-breaker

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

First of all you must select a diagnostics data set for transfer with a command (→ section "Circuit-breaker-specific commands", page 19). Then you can read out this set of diagnostics data.

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

→ Process data will not be updated as long as the diagnostics 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. The required command code is written to this address. The command codes can be found in → table 5, page 29. 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 diagnostics 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	Date	Value	Unit/ reference
0	Communication	Command	→ page 29	–
1		Reserved	–	
2		Circuit-breaker communication status, byte 0	→ page 31	
3		Circuit-breaker communication status, byte 1		
4		DMI ID	5 (fixed)	
5	Circuit-breaker status	Circuit-breaker status 0	→ page 32	–
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	Date	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 N conductor current, in relation to I_r		
23				
24		Maximum current for all phases		
25				
26		$I_{\text{dn eff}}$: effective fault current		
27				
28		Reserved	–	–
29	Current parameters,	I_r	→ page 34	–
30		I_i	→ page 35 and → page 36	
31		T_r	→ page 38	
32		I_{sd}	→ page 37	
33		T_{sd}	→ page 38	
34		I_{dn}	→ page 34	
35		T_{vdn}	→ page 38	
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	Date	Value	Unit/ reference
43	Device	Serial number LSB	0 to 2^{24}-1	–
44				
45		Serial number MSB		
46		Circuit-breaker type	→ page 39	
47		Circuit-breaker function	→ page 40 (IEC, (UL/CSA)	
48		I_{nom} (only for UL/CSA devices)	0 to 1600	
49				
50		NZM software version	→ page 41	
51		Hardware module (not supported)	–	
52		Software module[1)]	1.0.0 to 3.7.7	
66	Device settings	I_r	→ page 34	
67		I_i	→ page 35 and page 36	
68		T_r	→ page 38	
69		I_{sd}	→ page 37	
70		T_{sd}	→ page 38	
71		I_{dn}	→ page 34	
72		T_{vdn}	→ page 38	
73		Function switch	0 = I^2t OFF 1 = I^2t ON	
74	DMI mode	Output Q0 Mode	→ page 42	–
75		Output Q1 Mode		
76		Output Q2 Mode		
77		Output Q3 Mode		
78		Output Q4 Mode		
79		Output Q5 Mode		

Address	Data type	Date	Value	Unit/ reference
80	DMI mode	Switch element	1 = pushbutton, 0 = switch	–
81		Changeover time	0 to 99900 (in steps of 100)	ms
82				
83	Display mode	Display line 1	→ page 43	–
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 44	
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 44	–
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	Date	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 32	–
109		Status byte 1		
110		Status byte 2		
111		Status byte 3		
112		Status byte 4		
113		Reserved	–	
114		I_r	→ page 34	
115		I_i	→ page 35 and page 36	
116	Diagnostics	T_r	→ page 38	–
117		I_{sd}	→ page 37	
118		T_{sd}	→ page 38	
119		I_{dn}	→ page 34	
120		T_{vdn}	→ page 38	
121		Function switch, Data set No.	→ page 44	
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	Date	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	Date	Value	Unit/ reference
0	Communication	Command	→ page 29	–
1		Reserved	–	
2	Parameters	I_r	→ page 34	–
3		I_i	→ page 35 and page 36	
4		T_r	→ page 38	
5		I_{sd}	→ page 37	
6		T_{sd}	→ page 38	
7		I_{dn}	→ page 34	
8		T_{vdn}	→ page 38	
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	Date	Value	Unit/ reference
17	DMI mode	Output Q0 Mode[1]	→ page 42	–
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 time	0 to 999	100 ms
25				
26	Display mode	Display line 1	→ page 43	–
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 44	–
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 44	–

[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 21 and Table 4, page 27). 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]	
13_{hex}	Read diagnostics data set[2]	
14_{hex}	Read diagnostics data set[2]	
15_{hex}	Read diagnostics data set[2]	

Code	Command	Function
16_{hex}	Read diagnostics data set[2]	Reads the data of the older diagnostics event that is stored
17_{hex}	Read diagnostics data set[2]	
18_{hex}	Read diagnostics data set[2]	
19_{hex}	Read diagnostics data set[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	Alarm by L1
Bit 5	Alarm by L2
Bit 6	Alarm by L3
Bit 7	Alarm by 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 for 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	Transformers	0000	63	2
		0001	100	
		0010	160	
		0011	250	
		0100	250	3
		0101	400	
		0110	630	
		0111	630	4
		1000	800	
		1001	1 000	
		1010	1 250	
		1011	1 600	
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 39.

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:

01 010 000 = V1.2.0.

Aux. index 2 = 0

Aux. index 1 = 2

Main index = 1

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)