PNOZmulti communication interfaces

Configurable Control System PNOZmulti

Operating Manual-1001154-EN-13

This document is a translation of the original document.

SD means Secure Digital

Section 1	**Introduction**		**7**
	1.1	Definition of symbols	7
Section 2	**Overview - Communication options**		**8**
	2.1	Communication via the fieldbus modules	8
	2.2	Communication via the RS232/ETH interfaces	9
	2.3	Communication via Modbus/TCP	10
Section 3	**Safety**		**11**
	3.1	Intended use	11
	3.2	Safety regulations	11
	3.2.1	Use of qualified personnel	11
	3.2.2	Warranty and liability	11
	3.2.3	Disposal	12
Section 4	**Fieldbus modules**		**13**
	4.1	Basics	13
	4.1.1	Input data (to the PNOZmulti)	13
	4.1.2	Output data (from the PNOZmulti)	13
	4.1.3	Note on the PNOZ mc6p (CANopen)	14
	4.1.4	Assignment of Byte 0 ... Byte 3	16
	4.1.5	Assignment of Byte 4 ... Byte 18	17
	4.1.5.1	Example 1	20
	4.1.5.2	Example 2	21
	4.2	PNOZ mc2p, PNOZ mc2.1p, PNOZ mmc11p (SDO and PDO)	21
	4.2.1	Overview	21
	4.2.1.1	PNOZ mc2p	21
	4.2.1.2	PNOZ mc2.1p/PNOZ mmc11p	22
	4.2.2	Object Directory (Manufacturer Specific Profile Area)	23
	4.2.2.1	SDO Index 0x2000	23
	4.2.2.2	SDO Index 0x2001 and Index 0x2002	28
	4.2.2.3	SDO Index 0x2003	29
	4.2.2.4	SDO Index 0x2100	34
	4.2.2.5	SDO Index 0x2004	34
	4.2.2.6	SDO Index 0x2005	37
	4.3	PNOZ mc6p, PNOZ mc6.1p, PNOZ mmc6p, PNOZ mc12p (SDO)	38
	4.3.1	Overview	38
	4.3.2	System requirements	38
	4.3.3	Object Directory	39
	4.3.3.1	Index 2000	39
	4.3.3.2	Index 2001 and 2002	42
	4.3.3.3	Index 2003	44
	4.3.3.4	Index 2004	48
	4.3.3.5	Index 2005	51
	4.3.3.6	Index 2100	51
	4.4	PNOZ mc8p Ethernet IP / Modbus TCP	51
	4.4.1	Introduction	51

	4.4.2	Overview	51
	4.4.3	Module features	52
	4.4.4	Assigning the IP address to your PC	52
	4.4.5	Setting the IP address of the expansion module	52
	4.4.6	Changing IP settings	52
	4.4.7	Data exchange	53
	4.4.7.1	Ethernet IP	53
	4.4.7.2	Modbus TCP	53
	4.4.8	Web interface for commissioning and testing	54
	4.4.9	Restricting access	54
	4.4.10	Input and output data	55
	4.4.10.1	Assigning the inputs/outputs in the PNOZmulti Configurator to the Ethernet IP/Modbus TCP input/output data	55
	4.5	PNOZ mc10p sercos III	56
	4.5.1	Overview	56
	4.5.2	System requirements	56
	4.5.3	Object buffer	57
	4.5.3.1	Output data	57
	4.5.3.2	Diagnostic word	59
	4.5.3.3	Status of the inputs, outputs and LEDs	60
	4.5.3.4	Configuration	64
	4.5.3.5	Element types	67
	4.5.3.6	Input data	67
	4.5.3.7	Diagnostic data	68
	4.5.4	Firmware/FPGA update	69
	4.5.5	Forcing the virtual input data	70
	4.5.6	Communication with the sercos III Master	70
	4.5.6.1	Synchronous data exchange	70
	4.5.6.2	Asynchronous data access	71
	4.5.7	Sercos Master interface	72
	4.5.7.1	Supported profiles	72
	4.5.7.2	Default settings	73
	4.5.7.3	Description of IDNs	73
	4.5.7.4	Communication channels with the PNOZmulti	74
	4.5.7.5	Diagnostics	74
Section 5	**RS232/Ethernet interfaces**		**75**
	5.1	Overview	75
	5.2	System requirements	75
	5.3	Interface description	75
	5.3.1	Ethernet interfaces	75
	5.3.1.1	RJ45 interfaces ("Ethernet")	76
	5.3.1.2	Requirements of the connection cable and connector	76
	5.3.1.3	Interface configuration	77
	5.3.1.4	RJ45 connection cable	77
	5.3.1.5	Process data exchange	77
	5.3.2	Serial interface RS232	78

	5.4	Communication procedure	78
	5.5	Telegram structure	79
	5.5.1	Header	80
	5.5.2	Usable data	80
	5.5.3	Information data	80
	5.6	Usable data	81
	5.6.1	Virtual inputs (Input Byte 0 ... Input Byte 15)	81
	5.6.1.1	Mask (Mask Byte 0 ... Mask Byte 15)	81
	5.6.1.2	Watchdog	81
	5.6.2	Virtual outputs (Output Byte 0 ... Output Byte 15)	81
	5.6.3	LED status	82
	5.6.4	Tables	82
	5.7	Requests	82
	5.7.1	Send virtual inputs to the PNOZmulti	83
	5.7.2	Send virtual inputs to the PNOZmulti, request state of the virtual outputs and LED status from the PNOZmulti	84
	5.7.2.1	Control Byte (Byte 40)	85
	5.7.3	Request state of virtual inputs and outputs from PNOZmulti	86
	5.7.4	Send data from the PNOZmulti in table form	87
	5.7.5	Send input and output data (cf. fieldbus communication)	88
	5.7.5.1	Input data (to the PNOZmulti)	88
	5.7.5.2	Output data (from the PNOZmulti)	89
	5.7.5.3	Control Byte (Byte 5)	90
	5.8	Troubleshooting	91
	5.8.1	Request format does not meet specifications	91
	5.8.2	Error while executing a request	91
Section 6	**Modbus/TCP**		**93**
	6.1	System requirements	93
	6.2	Modbus/TCP - Basics	93
	6.3	Modbus/TCP with PNOZmulti	94
	6.4	Data areas	95
	6.4.1	Overview	95
	6.4.2	Function codes	95
	6.4.3	Data transfer limits	96
	6.4.4	Assignment of data areas	97
	6.4.4.1	Virtual inputs	97
	6.4.4.2	Control Register	98
	6.4.4.3	Virtual outputs	99
	6.4.4.4	LEDs	99
	6.4.4.5	Configuration	100
	6.4.4.6	State of the inputs from the base unit and expansion modules	102
	6.4.4.7	State of the outputs from the base unit and expansion modules	104
	6.4.4.8	LED status	105
	6.4.4.9	Diagnostic word, element types	108
	6.4.4.10	Current state of the virtual inputs	114
	6.4.4.11	Current states of the virtual inputs for the safe Ethernet connection	115

	6.4.4.12	State of process data	115
	6.4.4.13	Safe Ethernet connection	116
	6.4.5	Updating the data areas	116
	6.4.6	Bit addressing in a Register	117
	6.5	Example	118
Section 7	**Safe Ethernet connection**		**119**
	7.1	Overview	119
	7.2	System requirements	119
	7.3	Function description	119
	7.4	Configuration in the PNOZmulti Configurator	119
	7.5	Modbus configuration	120
	7.6	Reaction time	121
	7.7	Application guidelines	124
Section 8	**Diagnostic word**		**128**
	8.1	Introduction	128
	8.2	Elements with diagnostic word	128
	8.3	Structure of the diagnostic word	129
	8.4	Evaluate diagnostic word	129
	8.4.1	Example	131
	8.5	Compilation of the diagnostic words	131
	8.5.1	Function elements	132
	8.5.2	Cascading	134
	8.5.3	Logic elements	134
	8.5.4	Output elements	141
Section 9	**Appendix**		**143**
	9.1	Table assignment	143
	9.2	Table 1	143
	9.3	Table 3	149
	9.4	Table 4	151
	9.5	Table 5	155
	9.6	Table 7	159
	9.7	Table 8	166
	9.8	Table 9	169
	9.9	Table 10	172
	9.10	Table 11	172
	9.11	Element types	173

1 Introduction

1.1 Definition of symbols

Information that is particularly important is identified as follows:

DANGER!

This warning must be heeded! It warns of a hazardous situation that poses an immediate threat of serious injury and death and indicates preventive measures that can be taken.

WARNING!

This warning must be heeded! It warns of a hazardous situation that could lead to serious injury and death and indicates preventive measures that can be taken.

ATTENTION!

This refers to a hazard that can lead to a less serious or minor injury plus material damage, and also provides information on preventive measures that can be taken.

CAUTION!

This describes a situation in which the product or devices could be damaged and also provides information on preventive measures that can be taken. It also highlights areas within the text that are of particular importance.

Information

This gives advice on applications and provides information on special features.

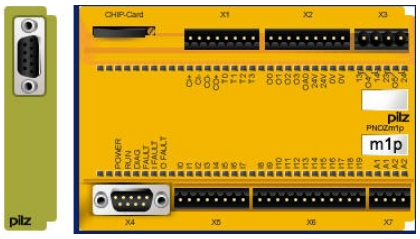
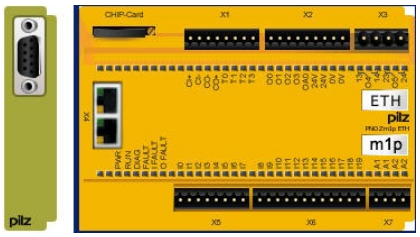
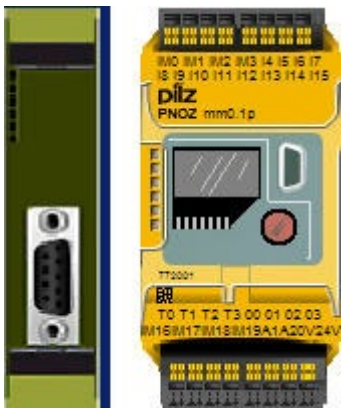
2 Overview - Communication options

2.1 Communication via the fieldbus modules

With communication via the fieldbus modules, the data area provided by the PNOZmulti for communication is divided into subsections, which are stored in tables. Each table consists of one or more segments.
The Master (PC, PLC) can request a segment from a table. This is delivered with the next response telegram. The virtual input and output data is also transmitted in each telegram (exception: communication with CANopen).

Communication via the fieldbus modules is described in detail in the chapter entitled "Fieldbus modules".

The following device combinations are possible:

Fieldbus modules		Base units
PNOZmulti fieldbus modules PNOZ mcXp		PNOZmulti base units with integrated RS232 interface PNOZ mXp
PNOZmulti fieldbus modules PNOZ mcXp		PNOZmulti base units with integrated Ethernet interface PNOZ mXp ETH
PNOZmulti Mini fieldbus modules PNOZ mmcXp		Base units PNOZmulti Mini

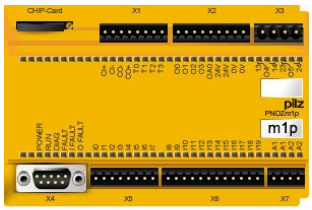
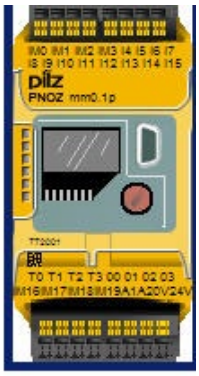
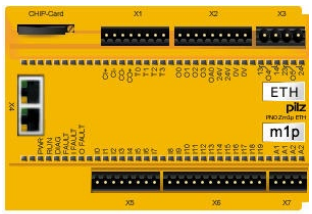
Information

If fieldbus modules are used for communication, the integrated RS232/Ethernet interface is only used to download the project during commissioning

2.2 Communication via the RS232/ETH interfaces

With communication via the integrated RS232 or Ethernet interface, data exchange is defined via a special protocol. This protocol is described in more detail in the chapter entitled RS232/Ethernet interfaces [📖 75].

The following device combinations are possible:

Base units PNOZmulti with integrated interface		Base units PNOZmulti Mini + communication module	
Base units with integrated RS232 interface PNOZ mXp	 	Base units PNOZmulti Mini PNOZ mmXp + Communication module with RS232 interface PNOZ mmc2p	  
Base units with integrated Ethernet interface PNOZ mXp ETH	 	Base units PNOZmulti Mini PNOZ mmXp + Communication module with Ethernet interface PNOZ mmc1p	  

Information

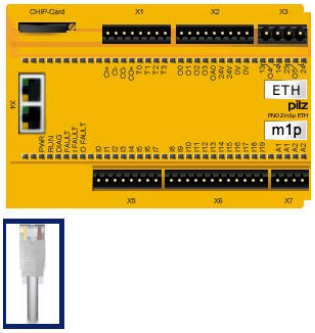
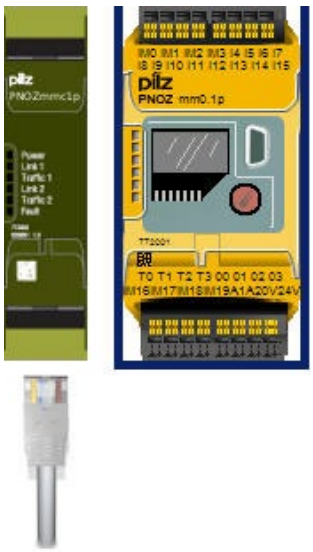
For communication via the integrated RS232 or Ethernet interface, the interface "Inputs/outputs that are downloaded via the integrated interface" must be configured in the hardware configuration in the PNOZmulti Configurator.

2.3 Communication via Modbus/TCP

For data exchange with Modbus/TCP, the PNOZmulti is the connection server. All the diagnostic data is defined in one data record, to which the client has direct access.

Communication with Modbus/TCP is described in detail in the chapter entitled Modbus/TCP [93].

The following device combinations are possible:

Base units PNOZmulti with integrated interface		**Base units PNOZmulti Mini + Communication module**	
PNOZmulti base units with Ethernet interface PNOZ mXp ETH		Base units PNOZmulti Mini + Communication module with Ethernet interface PNOZ mmc1p	

Information

For communication with Modbus/TCP, the interface "Inputs/outputs that are downloaded via the integrated interface" must be configured in the hardware configuration in the PNOZmulti Configurator.

3 Safety

3.1 Intended use

The communication interface on the configurable control system PNOZmulti is used to transfer diagnostic data to an application program. The data may only be used for non-safety purposes, e.g. visualisation.

CAUTION!

For details of the intended use and application of the configurable control system PNOZmulti, please refer to the operating instructions for the respective unit.

The following is deemed improper use in particular:

- Any component, technical or electrical modification to a product
- Use of a product outside the areas described in the product documentation
- Any use that is not in accordance with the documented technical details.

3.2 Safety regulations

3.2.1 Use of qualified personnel

The products may only be assembled, installed, programmed, commissioned, operated, maintained and decommissioned by competent persons.

A competent person is someone who, because of their training, experience and current professional activity, has the specialist knowledge required to test, assess and operate the work equipment, devices, systems, plant and machinery in accordance with the general standards and guidelines for safety technology.

It is the company's responsibility only to employ personnel who:

- Are familiar with the basic regulations concerning health and safety / accident prevention
- Have read and understood the information provided in this description under "Safety"
- And have a good knowledge of the generic and specialist standards applicable to the specific application.

3.2.2 Warranty and liability

All claims to warranty and liability will be rendered invalid if

- The product was used contrary to the purpose for which it is intended
- Damage can be attributed to not having followed the guidelines in the manual
- Operating personnel are not suitably qualified
- Any type of modification has been made (e.g. exchanging components on the PCB boards, soldering work etc.).

3.2.3 Disposal

- In safety-related applications, please comply with the mission time t_M in the safety-related characteristic data.
- When decommissioning, please comply with local regulations regarding the disposal of electronic devices (e.g. Electrical and Electronic Equipment Act).

4 Fieldbus modules

4.1 Basics

The input and output range is each reserved an area of 20 Bytes for communication via fieldbuses; this is updated approx. every 15 ms. The Master (PC, PLC) can send 20 Bytes to the PNOZmulti and receive 20 Bytes from the PNOZmulti. The Master can process the information in bytes, words or in double words.

4.1.1 Input data (to the PNOZmulti)

Double Word	Word	Byte	Content
0	0	0	State of virtual inputs
		1	
	1	2	
		3	Reserved
1	2	4	Table number
		5	Segment number
	3	6	Reserved
		7	Reserved
2	4	8	Reserved
		9	Reserved
	5	10	Reserved
		11	Reserved
3	6	12	Reserved
		13	Reserved
	7	14	Reserved
		15	Reserved
4	8	16	Reserved
		17	Reserved
	9	18	Reserved
		19	Reserved

4.1.2 Output data (from the PNOZmulti)

Double Word	Word	Byte	Content
0	0	0	State of virtual outputs
		1	
	1	2	
		3	LED status

Double Word	Word	Byte	Content
1	2	4	Table number
		5	Segment number
	3	6	Byte 0 of Table x, Segment y
		7	Byte 1 of Table x, Segment y
2	4	8	.
		9	.
	5	10	.
		11	.
3	6	12	.
		13	.
	7	14	.
		15	.
4	8	16	.
		17	.
	9	18	Byte 12 of Table x, Segment y
		19	Reserved

4.1.3 Note on the PNOZ mc6p (CANopen)

The output data on the PNOZmulti is stored as follows:

Byte	Object Index (hex)	Sub Index (hex)	PDO	COB-ID
0	2000	1	TPDO 1	180h + node address
1	2000	2		
2	2000	3		
3	2000	4		
4	2000	5		
5	2000	6		
6	2000	7		
7	2000	8		
8	2000	9	TPDO 2	280h + node address
9	2000	A		
10	2000	B		
11	2000	C		
12	2000	D		
13	2000	E		
14	2000	F		
15	2000	10		

Byte	Object Index (hex)	Sub Index (hex)	PDO	COB-ID
16	2000	11	TPDO 3	PNOZ mc6p: 1C0h + node address PNOZ mc6.1p, PNOZ mmc6p: 380h + node address
17	2000	12		
18	2000	13		
19	2000	14		

The input data on the PNOZmulti is stored as follows:

Byte	Object Index (hex)	Sub Index (hex)	PDO	COB-ID
0	2100	1	RPDO	200h + node address
1	2100	2		
2	2100	3		
3	2100	4		
4	2100	5		
5	2100	6		
6	2100	7		
7	2100	8		
8	2100	9	RPDO 2	300h + node address
9	2100	A		
10	2100	B		
11	2100	C		
12	2100	D		
13	2100	E		
14	2100	F		
15	2100	10		
16	2100	11	RPDO 3	PNOZ mc6p: 240h + node address PNOZ mc6.1p, PNOZ mmc6p: 400h + node address
17	2100	12		
18	2100	13		
19	2100	14		

Key to abbreviations:
TPDO: Transmit Process Data Object
RPDO: Receive Process Data Object
COB-ID: Communication Object Identifier

4.1.4 Assignment of Byte 0 ... Byte 3

The current status of the virtual outputs configured for the fieldbus plus the current status of the LED are always stored in Byte 0 ... Byte 3. All other information is stored in various tables (see Appendix).

Input area

The virtual inputs are defined by the Master and transferred to the PNOZmulti. Each input has a number, e.g. input bit 4 of byte 1 has the number i12.

Byte								
0	i7	i6	i5	i4	i3	i2	i1	i0
1	i15	i14	i13	i12	i11	i10	i9	i8
2	i23	i22	i21	i20	i19	i18	i17	i16
3	Reserved							

Output area

The virtual outputs are defined in the PNOZmulti Configurator. Each output that is used is given a number there, e.g. o0, o5... . The status of output o0 is stored in bit 0 of byte 0; the status of output o5 is stored in bit 5 of byte 0 etc.

Byte								
0	o7	o6	o5	o4	o3	o2	o1	o0
1	o15	o14	o13	o12	o11	o10	o9	o8
2	o23	o22	o21	o20	o19	o18	o17	o16

The status of the LEDs is stored in Byte 3 (output area only):

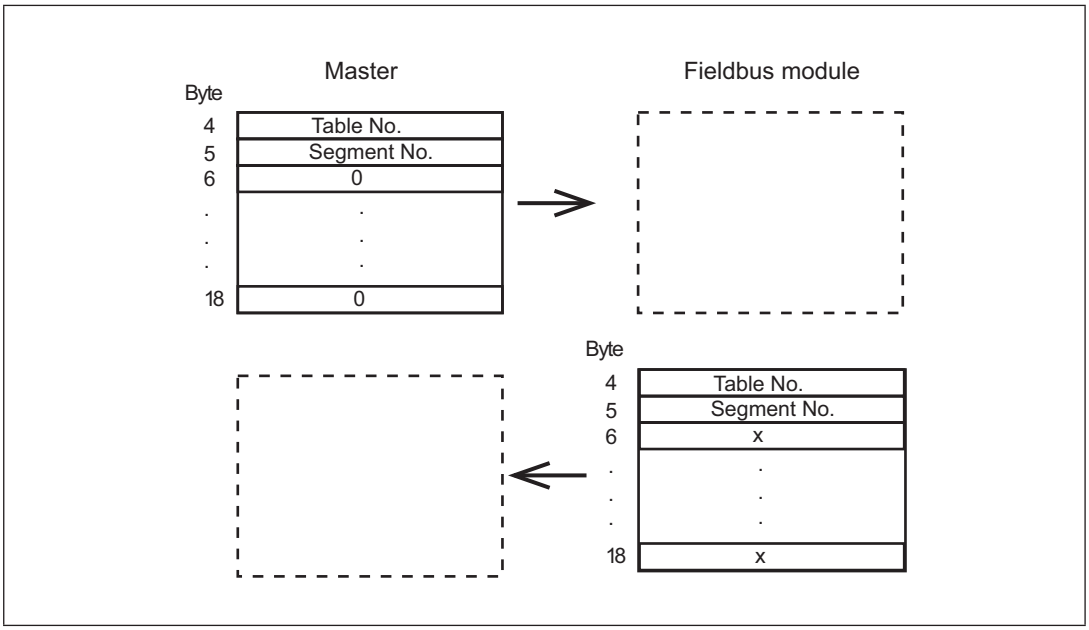
Bit 0 = 1:	LED OFAULT is lit or flashes
Bit 1 = 1:	LED IFAULT is lit or flashes
Bit 2 = 1:	LED FAULT is lit or flashes
Bit 3 = 1:	LED DIAG is lit
Bit 4 = 1:	LED RUN is lit
Bit 5:	Communication between the PNOZmulti and the fieldbus is working
Bit 6:	Reserved
Bit 7:	Reserved

4.1.5 Assignment of Byte 4 ... Byte 18

Byte	Table	
4	Table number	
5	Segment number	
6	Byte 0 of Table Segment 1	Segment 1
7	Byte 1 of Table Segment 1	
8	Byte 2 of Table Segment 1	
9	Byte 3 of Table Segment 1	
10	Byte 4 of Table Segment 1	
11	Byte 5 of Table Segment 1	
12	Byte 6 of Table Segment 1	
13	Byte 7 of Table Segment 1	
14	Byte 8 of Table Segment 1	
15	Byte 9 of Table Segment 1	
16	Byte 10 of Table Segment 1	
17	Byte 11 of Table Segment 1	
18	Byte 12 of Table Segment 1	
6	Byte 0 of Table Segment 2	Segment 2
7	Byte 1 of Table Segment 2	
8	Byte 2 of Table Segment 2	
9	Byte 3 of Table Segment 2	
10	Byte 4 of Table Segment 2	
11	Byte 5 of Table Segment 2	
12	Byte 6 of Table Segment 2	
13	Byte 7 of Table Segment 2	
14	Byte 8 of Table Segment 2	
15	Byte 9 of Table Segment 2	
16	Byte 10 of Table Segment 2	
17	Byte 11 of Table Segment 2	
18	Byte 12 of Table Segment 2	
.	.	.
.	.	.
.	.	.

Byte	Table	
6	Byte 0 of Table Segment n	Segment n
7	Byte 1 of Table Segment n	
8	Byte 2 of Table Segment n	
9	Byte 3 of Table Segment n	
10	Byte 4 of Table Segment n	
11	Byte 5 of Table Segment n	
12	Byte 6 of Table Segment n	
13	Byte 7 of Table Segment n	
14	Byte 8 of Table Segment n	
15	Byte 9 of Table Segment n	
16	Byte 10 of Table Segment n	
17	Byte 11 of Table Segment n	
18	Byte 12 of Table Segment n	

Each table consists of one or more segments. Each segment is made up of 13 Bytes. The tables have a fixed assignment. The Master requests the required data using the table number and segment number. The Slave (e.g. PNOZ mc3p) repeats the two numbers and sends the requested data. If data is requested that is not available, the Slave sends the error message “FF” instead of the segment number. The segments may be requested in any sequence.

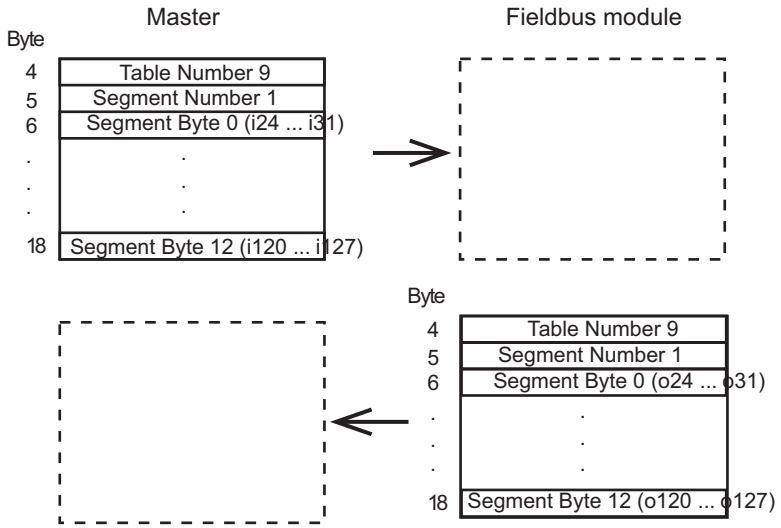


Exception: Table 9 segment 1:

With this table you can set the expanded inputs 24 - 127 and upload the expanded outputs 24 - 127. In contrast to the other tables, in this case the Master not only requests data but also sends input data to the PNOZmulti via the fieldbus module. Each input is assigned a Bit in the segment Bytes 0 ... 12 of the input data, each output is assigned a Bit in the segment Bytes 0 ... 12 of the output data.

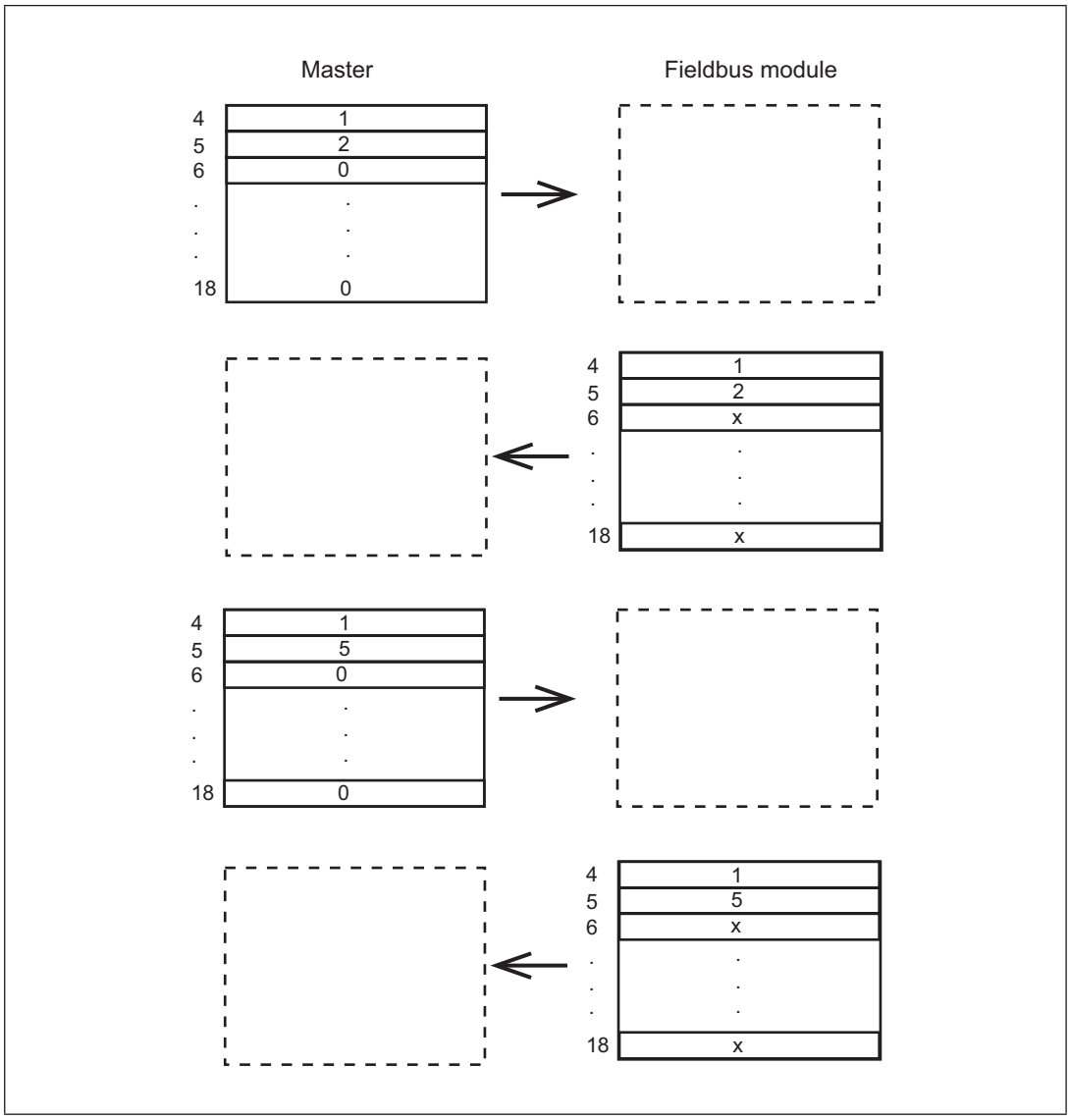
ATTENTION!

The expanded input Bits are only updated when table 9 segment 1 is accessed. In the event of a fieldbus error, input bits i24 ... i127 in the PNOZmulti are frozen!



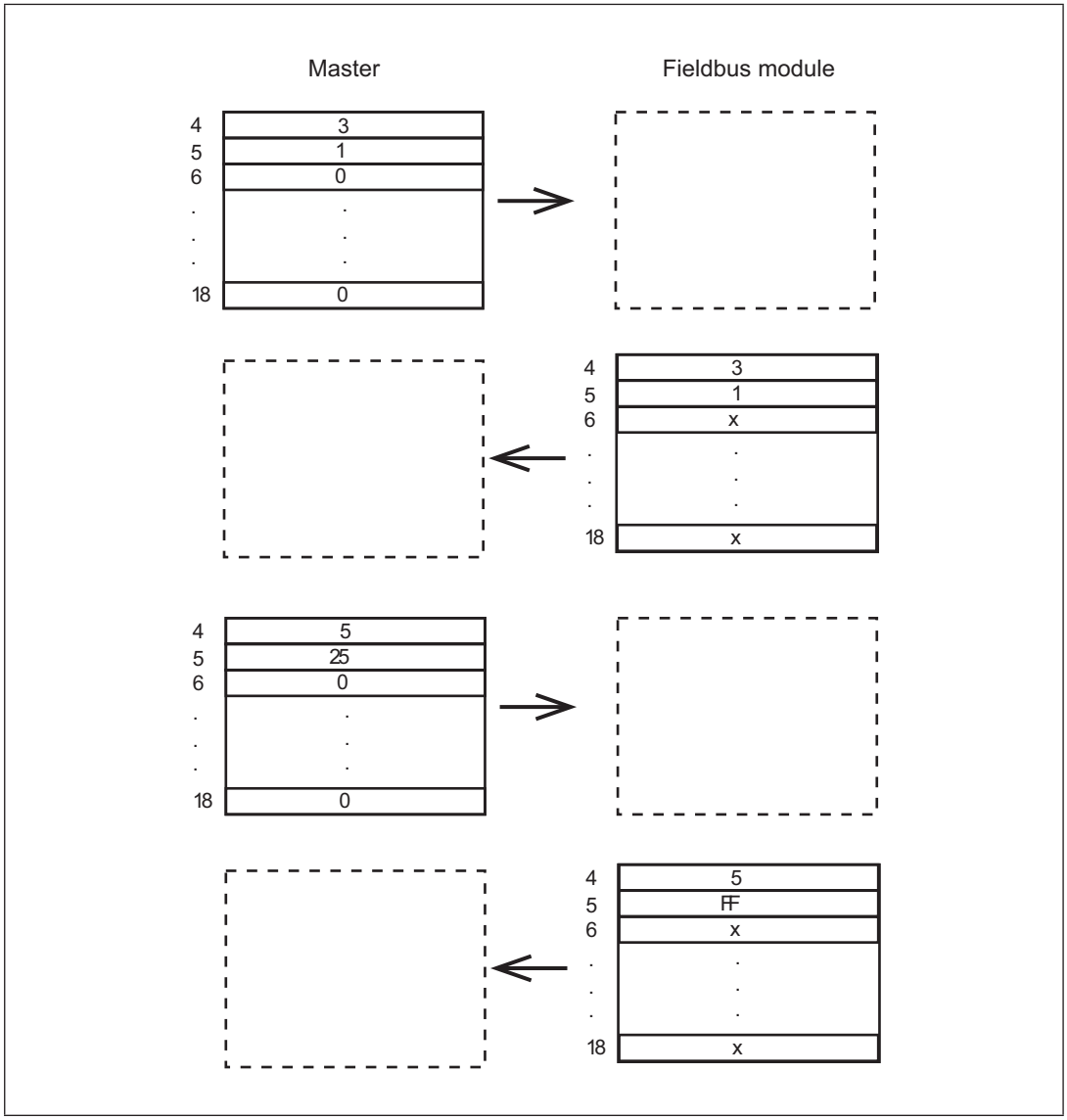
4.1.5.1 Example 1

The Master requests segment 2 from table 1. The fieldbus module repeats both these details and sends segment 2. Then the data from segment 12.70 cm table 1 is transmitted.



4.1.5.2 Example 2

The Master requests segment 1 from table 3. The fieldbus module repeats both of these details and sends segment 1. Then the Master requests segment 25 from table 5. As this table does not contain a segment 25, the Slave registers an error and sends back "FF".



4.2 PNOZ mc2p, PNOZ mc2.1p, PNOZ mmc11p (SDO and PDO)

4.2.1 Overview

4.2.1.1 PNOZ mc2p

All the objects (variables and parameters) that are relevant for these units are entered in the object directory. Service Data Objects (SDOs) are used for read and write access. The object directory is available as an EDS file (Electronic Data Sheet), enabling SDOs to be used in the PNOZ mc2p.

The manufacturer-specific part of the object directory is structured as follows:

PDO	Size	Name	Index	Sub-Index	Content
0x1A00	128	TxPDO	0x2000	0x01–0x80	Output data
0x1A01	128	TxPDO	0x2001	0x01–0x80	Diagnostic word (Low Byte)
0x1A02	128	TxPDO	0x2002	0x01–0x80	Diagnostic word (High Byte)
0x1A03	128	TxPDO	0x2003	0x01–0x80	Status of inputs
					Status of input LED
					State of outputs
					Status of LED
0x1600	20	RxPDO	0x2100	0x01–0x14	Input data

Information

Data with indices 2001 to 2003 is only updated by the PNOZmulti piece by piece in each cycle. This may mean that interdependent data is inconsistent. Updating all of the data can take up to 500 ms.

4.2.1.2 PNOZ mc2.1p/PNOZ mmc11p

All the objects (variables and parameters) that are relevant for these units are entered in the object directory. Service Data Objects (SDOs) are used for read and write access.

The SDOs in the PNOZ mc2.1p, PNOZ mmc11p are integrated in an ESI file (Ethercat Slave Information). The ESI file is incorporated in the EtherCAT Configurator for use of SDOs in the PNOZ mc2.1p, PNOZ mmc11p.

The manufacturer-specific part of the object directory is structured as follows:

PDO	Size	Name	Index	Sub-Index	Content
0x1A00	20	TxPDO	0x2000	0x01–0x14	Output data
0x1A01	128	TxPDO	Configurable	Configurable	Default configuration of important SDOs
0x1600	20	RxPDO	0x2100	0x01–0x14	Input data

Information

Data with indices 2001 to 2003 is only updated by the PNOZmulti piece by piece in each cycle. This may mean that interdependent data is inconsistent. Updating all of the data can take up to 500 ms.

Information

The data length and content of the PDOs can be freely configured by the EtherCAT Master. The maximum length is stated under "Size".

4.2.2 Object Directory (Manufacturer Specific Profile Area)

4.2.2.1 SDO Index 0x2000

This index contains the output data

Index (hex)	Name	Content	Example/explanation
0x2000:01	Input Byte 0	Outputs Bit 0 … 7 Fieldbus module	
0x2000:02	Input Byte 1	Outputs Bit 8 … 15 Fieldbus module	
0x2000:03	Input Byte 2	Outputs Bit 16 … 23 Fieldbus module	
0x2000:04	Input Byte 3	LED status	
0x2000:05	Input Byte 4	Table number	
0x2000:06	Input Byte 5	Segment number	
0x2000:07	Input Byte 6	Byte 0	
0x2000:08	Input Byte 7	Byte 1	
0x2000:09	Input Byte 8	Byte 2	
0x2000:A	Input Byte 9	Byte 3	
0x2000:B	Input Byte 10	Byte 4	
0x2000:C	Input Byte 11	Byte 5	
0x2000:D	Input Byte 12	Byte 6	
0x2000:E	Input Byte 13	Byte 7	
0x2000:F	Input Byte 14	Byte 8	
0x2000:10	Input Byte 15	Byte 9	
0x2000:11	Input Byte 16	Byte 10	
0x2000:12	Input Byte 17	Byte 11	
0x2000:13	Input Byte 18	Byte 12	
0x2000:14 … 0x2000:3F	Input Byte 19 ... Input Byte 62	Reserved	

Index (hex)	Name	Content	Example/explanation
0x2000:40	Input Byte 63	i0 ... i7 Safe Ethernet connection	Inputs, safe Ethernet connection
0x2000:41	Input Byte 64	i8 ... i15 Safe Ethernet connection	
0x2000:42	Input Byte 65	i16 ... i23 Safe Ethernet connection	
0x2000:43	Input Byte 66	i24 ... i31 Safe Ethernet connection	
0x2000:44	Input Byte 67	i32 ... i39 Safe Ethernet connection	
0x2000:45	Input Byte 68	i40 ... i47 Safe Ethernet connection	
0x2000:46 … 0x2000:47	Input Byte 69 … Input Byte 70	Reserved.	
0x2000:48	Input Byte 71	o0 ... o7 Safe Ethernet connection	Outputs, safe Ethernet connection
0x2000:49	Input Byte 72	o8 ... o15 Safe Ethernet connection	
0x2000:4A	Input Byte 73	o16 ... o23 Safe Ethernet connection	
0x2000:4B	Input Byte 74	o24 ... o31 Safe Ethernet connection	
0x2000:4C	Input Byte 75	o32 ... o39 Safe Ethernet connection	
0x2000:4D	Input Byte 76	i40 ... i47 Safe Ethernet connection	
0x2000:4E … 0x2000:4F	Input Byte 77 … Input Byte 78	Reserved.	
0x2000: 50	Input Byte 79	I0 ... I7 1st expansion module, left	Virtual inputs on the 2nd link module PNOZ ml1p:
0x2000:51	Input Byte 80	I8 ... I15 1st expansion module, left	
0x2000:52	Input Byte 81	I16 ... I23 1st expansion module, left	Sub-Index 54:
			I7 | I6 | I5 | I4 | I3 | I2 | I1 | I0
0x2000:53	Input Byte 82	I24 ... I31 1st expansion module, left	Sub-Index 55:
			I15 | I14 | I13 | I12 | I11 | I10 | I9 | I8

Index (hex)	Name	Content	Example/explanation							
0x2000:54	Input Byte 83	I0 ... I7 2nd expansion module, left	Sub-Index 56:							
			I23	I22	I21	I20	I19	I18	I17	I16
0x2000:55	Input Byte 84	I8 ... I15 2nd expansion module, left	Sub-Index 57:							
			I31	I30	I29	I28	I27	I26	I25	I24
0x2000:56	Input Byte 85	I16 ... I23 2nd expansion module, left	If an output has a high signal, the corresponding Bit will contain "1"; if the output is open (low signal), the Bit will contain "0".							
0x2000:57	Input Byte 86	I24 ... I31 2nd expansion module, left								
0x2000:58	Input Byte 87	I0 ... I7 3rd expansion module, left								
0x2000:59	Input Byte 88	I8 ... I15 3rd expansion module, left								
0x2000:5A	Input Byte 89	I16 ... I23 3rd expansion module, left								
0x2000:5B	Input Byte 90	I24 ... I31 3rd expansion module, left								
0x2000:5C	Input Byte 91	I0 ... I7 4th expansion module, left								
0x2000:5D	Input Byte 92	I8 ... I15 4th expansion module, left								
0x2000:5E	Input Byte 93	I16 ... I23 4th expansion module, left								
0x2000:5F	Input Byte 94	I24 ... I31 4th expansion module, left								
0x2000:60	Input Byte 95	I0 ... I7 5th expansion module, left								
0x2000:61	Input Byte 96	I8 ... I15 5th expansion module, left								

<table>
<tr><th>Index (hex)</th><th>Name</th><th>Content</th><th colspan="8">Example/explanation</th></tr>
<tr><td>0x2000:62</td><td>Input Byte 97</td><td>I16 ... I23
5th expansion module, left</td><td colspan="8"></td></tr>
<tr><td>0x2000:63</td><td>Input Byte 98</td><td>I24 ... I31
5th expansion module, left</td><td colspan="8"></td></tr>
<tr><td>0x2000:64</td><td>Input Byte 99</td><td>I0 ... I7
6th expansion module, left</td><td colspan="8"></td></tr>
<tr><td>0x2000:65</td><td>Input Byte 100</td><td>I8 ... I15
6th expansion module, left</td><td colspan="8"></td></tr>
<tr><td>0x2000:66</td><td>Input Byte 101</td><td>I16 ... I23
6th expansion module, left</td><td colspan="8"></td></tr>
<tr><td>0x2000:67</td><td>Input Byte 102</td><td>I24 ... I31
6th expansion module, left</td><td colspan="8"></td></tr>
<tr><td>0x2000:68</td><td>Input Byte 103</td><td>O0 ... O7
1st expansion module, left</td><td colspan="8">Virtual outputs on the 3rd link module PNOZ ml1p:</td></tr>
<tr><td rowspan="2">0x2000:69</td><td rowspan="2">Input Byte 104</td><td rowspan="2">O8 ... O15
1st expansion module, left</td><td colspan="8">Sub-Index 70:</td></tr>
<tr><td>O7</td><td>O6</td><td>O5</td><td>O4</td><td>O3</td><td>O2</td><td>O1</td><td>O0</td></tr>
<tr><td rowspan="2">0x2000:6A</td><td rowspan="2">Input Byte 105</td><td rowspan="2">O16 ... O23
1st expansion module, left</td><td colspan="8">Sub-Index 71:</td></tr>
<tr><td>O15</td><td>O14</td><td>O13</td><td>O12</td><td>O11</td><td>O10</td><td>O9</td><td>O8</td></tr>
<tr><td rowspan="2">0x2000:6B</td><td rowspan="2">Input Byte 106</td><td rowspan="2">O24 ... O31
1st expansion module, left</td><td colspan="8">Sub-Index 72:</td></tr>
<tr><td>O23</td><td>O22</td><td>O21</td><td>O20</td><td>O19</td><td>O18</td><td>O17</td><td>O16</td></tr>
<tr><td rowspan="2">0x2000:6C</td><td rowspan="2">Input Byte 107</td><td rowspan="2">O0 ... O7
2nd expansion module, left</td><td colspan="8">Sub-Index 73:</td></tr>
<tr><td>O31</td><td>O30</td><td>O29</td><td>O28</td><td>O27</td><td>O26</td><td>O25</td><td>O24</td></tr>
<tr><td>0x2000:6D</td><td>Input Byte 108</td><td>O8 ... O15
2nd expansion module, left</td><td colspan="8" rowspan="2">If an output has a high signal, the corresponding Bit will contain "1"; if the output is open (low signal), the Bit will contain "0".</td></tr>
<tr><td>0x2000:6E</td><td>Input Byte 109</td><td>O16 ... O23
2nd expansion module, left</td></tr>
</table>

Index (hex)	Name	Content	Example/explanation
0x2000:6F	Input Byte 110	O24 ... O31 2nd expansion module, left	
0x2000:70	Input Byte 111	O0 ... O7 3rd expansion module, left	
0x2000:71	Input Byte 112	O8 ... O15 3rd expansion module, left	
0x2000:72	Input Byte 113	O16 ... O23 3rd expansion module, left	
0x2000:73	Input Byte 114	O24 ... O31 3rd expansion module, left	
0x2000:74	Input Byte 115	O0 ... O7 4th expansion module, left	
0x2000:75	Input Byte 116	O8 ... O15 4th expansion module, left	
0x2000:76	Input Byte 117	O16 ... O23 4th expansion module, left	
0x2000:77	Input Byte 118	O24 ... O31 4th expansion module, left	
0x2000:78	Input Byte 119	O0 ... O7 5th expansion module, left	
0x2000:79	Input Byte 120	O8 ... O15 5th expansion module, left	
0x2000:7A	Input Byte 121	O16 ... O23 5th expansion module, left	
0x2000:7B	Input Byte 122	O24 ... O31 5th expansion module, left	
0x2000:7C	Input Byte 123	O0 ... O7 6th expansion module, left	

Index (hex)	Name	Content	Example/explanation
0x2000:7D	Input Byte 124	O8 ... O15 6th expansion module, left	
0x2000:7E	Input Byte 125	O16 ... O23 6th expansion module, left	
0x2000:7F	Input Byte 126	O24 ... O31 6th expansion module, left	
0x2000:80	Input Byte 127	Reserved	

4.2.2.2 SDO Index 0x2001 and Index 0x2002

This index contains the diagnostic words and the output bits for the Element IDs.

<table>
<tr><th>Index (hex)</th><th>Name</th><th colspan="2">Content</th><th colspan="9">Example/explanation</th></tr>
<tr><td>0x2001:01</td><td>Input Byte 128</td><td colspan="2">Low Byte diagnostic word.
Element ID=1</td><td colspan="9" rowspan="2">The diagnostic word is displayed in the PNOZmulti Configurator and on the PVIS expanded diagnostics (see chapter entitled Diagnostic word [ 128] and the online help for the PNOZmulti Configurator)
Element-ID = 1,
e.g. diagnostic word of E-STOP:
Low Byte:</td></tr>
<tr><td>...</td><td>...</td><td colspan="2"></td></tr>
<tr><td>0x2001:64</td><td>Input Byte 227</td><td colspan="2">Low Byte diagnostic word.
Element ID=100</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td></td></tr>
<tr><td></td><td></td><td colspan="2"></td><td colspan="9">Message: Pushbutton operated</td></tr>
<tr><td>0x2001:65
...
0x2001:71</td><td>Input Byte 228
...
Input Byte 240</td><td colspan="2">Output Bits of Element ID = 1 ... 100</td><td colspan="9">Each element is assigned an ID in the PNOZmulti Configurator. If the element's output = 0 (no enable), the corresponding bit is set.</td></tr>
<tr><td></td><td></td><td colspan="2"></td><td colspan="2">Sub Index</td><td colspan="7">Element ID</td></tr>
<tr><td></td><td></td><td colspan="2"></td><td>65</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td></td><td></td><td colspan="2"></td><td>66</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td></tr>
<tr><td></td><td></td><td colspan="2"></td><td>67</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td></tr>
<tr><td></td><td></td><td colspan="2"></td><td colspan="9">...</td></tr>
<tr><td></td><td></td><td colspan="2"></td><td>6F</td><td>88</td><td>87</td><td>86</td><td>85</td><td>84</td><td>83</td><td>82</td><td>81</td></tr>
<tr><td></td><td></td><td colspan="2"></td><td>70</td><td>96</td><td>95</td><td>94</td><td>93</td><td>92</td><td>91</td><td>90</td><td>89</td></tr>
<tr><td></td><td></td><td colspan="2"></td><td>71</td><td>-</td><td>-</td><td>-</td><td>-</td><td>100</td><td>99</td><td>98</td><td>97</td></tr>
<tr><td>0x2001:72
...
0x2001:80</td><td>Input Byte 241
...
Input Byte 255</td><td colspan="2">Reserved</td><td colspan="9"></td></tr>
</table>

Index (hex)	Name	Content	Example/explanation
0x2002:01	Input Byte 256	High Byte diagnostic word. Element ID=1	See Index 2001 for comment Element ID = 1, e.g. diagnostic word of E-STOP: High Byte:
...	...	...	
0x2002:64	Input Byte 355	High Byte diagnostic word. Element ID=100	0 0 0 0 0 0 0 1 Message: Wiring error, clock error
0x2002:65 ... 0x2002:80	Input Byte 356 ... Input Byte 383	Reserved	

4.2.2.3 SDO Index 0x2003

This index contains the state of the inputs/outputs plus the LED status

Index (hex)	Input Byte	Content	Example/Comment
0x2003:01	384	I0 ... I7 base unit IM0 ... I7 base unit Mini	For example: The safety system consists of one base unit PNOZ m1p and one expansion module PNOZ mi1p
0x2003:02	385	I8 ... I15 base unit, I8 ... I15 base unit Mini	
0x2003:03	386	I16 ... I19 base unit IM16 ... IM19 base unit Mini	
0x2003:04	387	0	Sub-Index 1: PNOZ m1p I7 I6 I5 I4 I3 I2 I1 I0
0x2003:05	388	0	Sub-Index 2: PNOZ m1p I15 I14 I13 I12 I11 I10 I9 I8
0x2003:06	389	I0 ... I7 1st expansion module, right	Sub-Index 3: PNOZ m1p 0 0 0 0 I19 I18 I17 I16
0x2003:07	390	I0 ... I7 2nd expansion module, right	Sub-Index 4: 0 0 0 0 0 0 0 0
0x2003:08	391	I0 ... I7 3rd expansion module, right	Sub-Index 5: 0 0 0 0 0 0 0 0
0x2003:09	392	I0 ... I7 4th expansion module, right	Sub-Index 6: PNOZ mi1p I7 I6 I5 I4 I3 I2 I1 I0

<table>
<tr><th>Index (hex)</th><th>Input Byte</th><th>Content</th><th>Example/Comment</th></tr>
<tr><td>0x2003:A</td><td>393</td><td>I0 ... I7 5th expansion module, right</td><td rowspan="5">If an input has a high signal, the corresponding bit will be "1"; if an input has a low signal, the bit will be "0".

INFORMATION:
On the base units PNOZmulti Mini, the status of the configurable inputs/outputs is only displayed if they are configured as inputs in the PNOZmulti Configurator.
Assignment of Bytes on the base units PNOZmulti Mini:</td></tr>
<tr><td>0x2003:B</td><td>394</td><td>I0 ... I7 6th expansion module, right</td></tr>
<tr><td>0x2003:C</td><td>395</td><td>I0 ... I7 7th expansion module, right</td></tr>
<tr><td>0x2003:D</td><td>396</td><td>I0 ... I7 8th expansion module right</td></tr>
<tr><td rowspan="4">0x2003:E
...
0x2003:10</td><td rowspan="4">397
...
399</td><td rowspan="4">Reserved</td><td>Sub-Index 1: PNOZ mmxp</td></tr>
<tr><td>I7 | I6 | I5 | I4 | IM3 | IM2 | IM1 | IM0</td></tr>
<tr><td>Sub-Index 2: PNOZ mmxp</td></tr>
<tr><td>I15 | I14 | I13 | I12 | I11 | I10 | I9 | I8
Sub-Index 3: PNOZ mmxp
0 | 0 | 0 | 0 | IM19 | IM18 | IM17 | IM16</td></tr>
<tr><td>0x2003:11</td><td>400</td><td>LED I0 ... I7 base unit</td><td rowspan="3">For example: The safety system consists of one base unit PNOZ m1p and one expansion module PNOZ mi1p</td></tr>
<tr><td>0x2003:12</td><td>401</td><td>LED I8 ... I15 base unit</td></tr>
<tr><td>0x2003:13</td><td>402</td><td>LED I16 ... I19 base unit</td></tr>
<tr><td rowspan="2">0x2003:14</td><td rowspan="2">403</td><td rowspan="2">0</td><td>Sub-Index 11: PNOZ m1p</td></tr>
<tr><td>I7 | I6 | I5 | I4 | I3 | I2 | I1 | I0</td></tr>
<tr><td rowspan="2">0x2003:15</td><td rowspan="2">404</td><td rowspan="2">0</td><td>Sub-Index 12: PNOZ m1p</td></tr>
<tr><td>I15 | I14 | I13 | I12 | I11 | I10 | I9 | I8</td></tr>
<tr><td rowspan="2">0x2003:16</td><td rowspan="2">405</td><td rowspan="2">LED I0 ... I7 1st expansion module, right</td><td>Sub-Index 13: PNOZ m1p</td></tr>
<tr><td>0 | 0 | 0 | 0 | I19 | I18 | I17 | I16</td></tr>
<tr><td rowspan="2">0x2003:17</td><td rowspan="2">406</td><td rowspan="2">LED I0 ... I7 2nd expansion module, right</td><td>Sub-Index 14:</td></tr>
<tr><td>0 | 0 | 0 | 0 | 0 | 0 | 0 | 0</td></tr>
<tr><td rowspan="2">0x2003:18</td><td rowspan="2">407</td><td rowspan="2">LED I0 ... I7 3rd expansion module, right</td><td>Sub-Index 15:</td></tr>
<tr><td>0 | 0 | 0 | 0 | 0 | 0 | 0 | 0</td></tr>
<tr><td rowspan="2">0x2003:19</td><td rowspan="2">408</td><td rowspan="2">LED I0 ... I7 4th expansion module, right</td><td>Sub-Index 16: PNOZ mi1p</td></tr>
<tr><td>I7 | I6 | I5 | I4 | I3 | I2 | I1 | I0</td></tr>
</table>

Index (hex)	Input Byte	Content	Example/Comment
0x2003:1A	409	LED I0 ... I7 5th expansion module, right	If the LED on an input is flashing, the corresponding bit contains "1"; if the LED is not flashing, the bit contains "0".
0x2003:1B	410	LED I0 ... I7 6th expansion module, right	
0x2003:1C	411	LED I0 ... I7 7th expansion module, right	
0x2003:1D	412	LED I0 ... I7 8th expansion module right	
0x2003:1E ... 0x2003:20	413 ... 415	Reserved	
0x2003:21	416	IM0 ... IM3 base unit Mini	Assignment of Bytes on the **base units PNOZmulti:**
0x2003:22	417	0	
0x2003:23	418	IM16 ... T3M23 base unit Mini	
0x2003:24	419	O0 ... O3 base unit	Sub-Index 24:
			0 | 0 | 1 | 1 | O3 | O2 | O1 | O0
	420	O4 and O5 base unit	Sub-Index 25:
			0 | 0 | 0 | 0 | 0 | 0 | O5 | O4
0x2003:26	421	O0 ... O7 1st expansion module, right	**PNOZ mo1p**
			Sub-Index 26 ... 2D:
0x2003:27	422	O0 ... O7 2nd expansion module, right	0 | 0 | 0 | 0 | O3 | O2 | O1 | O0
			Sub-Index 36 ... 3D:
0x2003:28	423	O0 ... O7 3rd expansion module, right	0 | 0 | 0 | 0 | 0 | 0 | 0 | 0
			PNOZ mo2p, PNOZ mo3p
0x2003:29	424	O0 ... O7 4th expansion module, right	Sub-Index 26 ... 2D:
			0 | 0 | 0 | 0 | 0 | 0 | O1 | O0
0x2003:2A	425	O0 ... O7 5th expansion module right	Sub-Index 36 ... 3D:
			0 | 0 | 0 | 0 | 0 | 0 | 0 | 0
0x2003:2B	426	O0 ... O7 6th expansion module right	**PNOZ mo4p, PNOZ mo5p**
			Sub-Index 26 ... 2D:
0x2003:2C	427	O0 ... O7 7. Expansion module, right	0 | 0 | 0 | 0 | O3 | O2 | O1 | O0
			Sub-Index 36 ... 3D:
0x2003:2D	428	O0 ... O7 8th Expansion module, right	**PNOZ mc1p**
0x2003:2E ... 0x2003:30	429 ... 431	Reserved	Sub-Index 26 ... 2D:
			A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0
			Sub-Index 36 ... 3D:

<table>
<tr><th>Index (hex)</th><th>Input Byte</th><th>Content</th><th colspan="8">Example/Comment</th></tr>
<tr><td rowspan="2">0x2003:31
...
0x2003:35</td><td rowspan="2">432
...
436</td><td rowspan="2">0</td><td>A15</td><td>A14</td><td>A13</td><td>A12</td><td>A11</td><td>A10</td><td>A9</td><td>A8</td></tr>
<tr><td colspan="8">If an output has a high signal, the corresponding Bit will contain "1"; if the output is open (low signal), the Bit will contain "0".

INFORMATION
On the base units PNOZmulti Mini, the status of the configurable inputs/outputs is only displayed if they are configured as outputs in the PNOZmulti Configurator.
Assignment of Bytes on the base units PNOZmulti Mini:</td></tr>
<tr><td rowspan="2">0x2003:36</td><td rowspan="2">437</td><td rowspan="2">O8 ... O15 1st expansion module, right</td><td colspan="8">Sub-Index 21:</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>IM3</td><td>IM2</td><td>IM1</td><td>IM0</td></tr>
<tr><td rowspan="2">0x2003:37</td><td rowspan="2">438</td><td rowspan="2">O8 ... O15 2nd expansion module, right</td><td colspan="8">Sub-Index 23:</td></tr>
<tr><td>T3
M23</td><td>T2
M22</td><td>T1
M21</td><td>T0;20</td><td>IM19</td><td>IM18</td><td>IM17</td><td>IM16</td></tr>
<tr><td>0x2003:38</td><td>439</td><td>O8 ... O15 3rd expansion module, right</td><td colspan="8" rowspan="8"></td></tr>
<tr><td>0x2003:39</td><td>440</td><td>O8 ... O15 4th expansion module, right</td></tr>
<tr><td>0x2003:3A</td><td>441</td><td>O8 ... O15 5th expansion module, right</td></tr>
<tr><td>0x2003:3B</td><td>442</td><td>O8 ... O15 6th expansion module right</td></tr>
<tr><td>0x2003:3C</td><td>443</td><td>O8 ... O15 7th expansion module, right</td></tr>
<tr><td>0x2003:3D</td><td>Input Byte 444</td><td>O8 ... O15 8th expansion module, right</td></tr>
<tr><td>0x2003:3E
...
0x2003:40</td><td>445
...
447</td><td>Reserved</td></tr>
</table>

Index (hex)	Input Byte	Content	Example/Comment
0x2003:41	448	RUN	Depending on the LED status, the following Hex code will be in Sub-Index 41 ... 4D 00 hex: LED off FF hex: LED on 30 hex: LED flashes
0x2003:42	449	DIAG	
0x2003:43	450	FAULT	
0x2003:44	451	IFAULT	
0x2003:45	452	OFAULT	
0x2003:46	453	FAULT 1: Expansion module, right	
0x2003:47	454	FAULT 2: Expansion module, right	
0x2003:48	455	FAULT 3: Expansion module, right	
0x2003:49	456	FAULT 4: Expansion module, right	
0x2003:4A	457	FAULT 5: Expansion module, right	
0x2003:4B	458	FAULT 6: Expansion module, right	
0x2003:4C	459	FAULT 7: Expansion module, right	
0x2003:4D	460	FAULT 8: Expansion module, right	
0x2003:4E	461	FAULT 1: Expansion module, left	Depending on the LED status, the following Hex code will be in Sub-Index 4E ... 53: 00 hex: LED off FF hex: LED on 30 hex: LED flashes
0x2003:4F	462	FAULT 2: Expansion module, left	
0x2003:50	463	FAULT 3: Expansion module, left	
0x2003:51	464	FAULT 4: Expansion module, left	
0x2003:52	465	FAULT 5: Expansion module, left	
0x2003:53	466	FAULT 6: Expansion module, left	
0x2003:54 ... 0x2003:80	467 ... 511	Reserved	

4.2.2.4 SDO Index 0x2100

This index contains the input data

Index (hex)	Name	Content	Example/Comment
0x2100:01	Output Byte 0	Inputs Bit 0 … 7	
0x2100:02	Output Byte 1	Inputs Bit 8 … 15	
0x2100:03	Output Byte 2	Inputs Bit 16 … 23	
0x2100:04	Output Byte 3	Reserved	
0x2100:05	Output Byte 4	Table number	
0x2100:06	Output Byte 5	Segment number	
0x2100:07 ... 0x2100:14	Output Byte 6 ... Output Byte 19	Reserved	

4.2.2.5 SDO Index 0x2004

This index contains the PNOZmulti's configuration data

Index (hex)	Content	Example/explanation
0x2004:01	Data transfer	Subindex 1: Bit 0 = 1: All configuration data has been downloaded to the fieldbus module
0x2004:02	Reserved	
0x2004:03	Number of elements	Number of configured elements with Element ID
0x2004:04 ... 0x2004:10	Reserved	
0x2004:11 ... 0x2004:14	Product number (hex)	Product number 733 100: 000BCBEC hex Sub-Index 11: 00, Sub-Index 12: 0B, Sub-Index 13: CB,Sub-Index 14: EC
0x2004:15 ... 0x2004:18	Unit version (hex)	Unit version 20: 14 hex Sub-Index 15: 00, Sub-Index 16: 00, Sub-Index 17: 00, Sub-Index 18: 14
0x2004:19 ... 0x2004:1C	Serial number (hex)	Serial number 123 456: 0001E240 hex. Sub-Index 19: 00, Sub-Index 1A: 01, Sub-Index 1B: E2,Sub-Index 1C: 40
0x2004:1D ... 0x2004:1E	Check sum safe (hex)	Check sum A1B2 hex: Sub-Index 1D: A1, Sub-Index 1E: B2
0x2004:1F ... 0x2004:20	Overall project check sum (hex)	Check sum 3C5A hex: Sub-Index 1F: 3C, Byte 32: 5A

Index (hex)	Content	Example/explanation
0x2004:21 ... 0x2004:24	Reserved	
0x2004:25 ... 0x2004:28	Project creation date (hex)	Creation date: 28.11.2003 Sub-Index 25: 1C, Sub-Index 26: 0B, Sub-Index 27: 07, Sub-Index 28: D3
0x2004:29 ... 0x2004:2B	Reserved	
0x2004:2C	Configuration, fieldbus module/integrated interface	Sub-Index 2C contains the Hex code for a fieldbus module (installed on the left) or for inputs and outputs via the integr. interface: (see Table 1 Segment 2 Byte 0) Sub-Index 2D ... 34 contains the Hex code for the expansion modules on the right: PNOZ mi1p: 08 PNOZ mi2p: 38 PNOZ mo1p: 18 PNOZ mo2p: 10 PNOZ mo3p: 30 PNOZ mo4p: 28 PNOZ mo5p: 48 PNOZ mc1p: 20 PNOZ ms3p: 68 PNOZ ms4p: 78 PNOZ ms1p/PNOZ ms2p: 88 PNOZ ms2p HTL: 58 PNOZ ms3p HTL: 64 PNOZsigma with one output: 11 PNOZsigma with two outputs: 22 No expansion module: 00
0x2004:2D	Configuration, 1st expansion module, right	
0x2004:2E	Configuration, 2nd expansion module, right	
0x2004:2F	Configuration, 3rd expansion module, right	
0x2004:30	Configuration, 4th expansion module, right	
0x2004:31	Configuration, 5th expansion module, right	
0x2004:32	Configuration, 6th expansion module, right	
0x2004:33	Configuration, 7th expansion module, right	
0x2004:34	Configuration, 8th expansion module right	
0x2004:35 ... 0x2004:38	Reserved	

Index (hex)	Content	Example/explanation
0x2004:39	1st character (Low Byte)	Sub-Index 39 ... 58 contains the project name defined in the PNOZmultil Configurator under "Enter project data"; this is stored in UNICODE format, 2 Bytes contain the Hex code of the individual UNICODE characters.
0x2004:3A	1st character (High Byte)	
0x2004:3B	2nd character (Low Byte)	
0x2004:3C	2nd character (High Byte)	
0x2004:3D	3rd character (Low Byte)	
0x2004:3E	3rd character (High Byte)	
0x2004:3F	4th character (Low Byte)	
0x2004:40	4th character (High Byte)	
0x2004:41	5th character (Low Byte)	
0x2004:42	5th character (High Byte)	
0x2004:43	6th character (Low Byte)	
0x2004:44	6th character (High Byte)	
0x2004:45	7th character (Low Byte)	
0x2004:46	7th character (High Byte)	
0x2004:47	8th character (Low Byte)	
0x2004:48	8th character (High Byte)	
0x2004:49	9th character (Low Byte)	
0x2004:4A	9th character (High Byte)	
0x2004:4B	10th character (Low Byte)	
0x2004:4C	10th character (High Byte)	
0x2004:4D	11th character (Low Byte)	
0x2004:4E	11th character (High Byte)	
0x2004:4F	12th character (Low Byte)	
0x2004:50	12th character (High Byte)	
0x2004:51	13th character (Low Byte)	
0x2004:52	13th character (High Byte)	
0x2004:53	14th character (Low Byte)	
0x2004:54	14th character (High Byte)	
0x2004:55	15th character (Low Byte)	
0x2004:56	15th character (High Byte)	
0x2004:57	16th character (Low Byte)	
0x2004:58	16th character (High Byte)	

Index (hex)	Content	Example/explanation
0x2004:59	Day	Date on which the program on the chip card was last modified Date modified : 28.11.2003 Sub-Index 59: 1C, Sub-Index 5A: 0B, Sub-Index 5B: 07, Sub-Index 5C: D3 Time: 14 hours 25 minutes Sub-Index 5D: 0E, Sub-Index 5E: 19 Time zone 1: Sub-Index 5F: 01
0x2004:5A	Month	
0x2004:5B	Year (High Byte)	
0x2004:5C	Year (Low Byte)	
0x2004:5D	Hour	
0x2004:5E	Minute	
0x2004:5F	Time zone	
0x2004:60	Configuration, 1st expansion module, left	Sub-Index 60 ... 65 contains the Hex code for the expansion modules to the left of the base unit. Any fieldbus module in these Sub-Indices will not be considered (see Index 2004, Sub-Index 2C). PNOZ ml1p: A8 PNOZ ml2p: C8 PNOZ ma1p: B8
0x2004:61	Configuration, 2nd expansion module, left	
0x2004:62	Configuration, 3rd expansion module, left	
0x2004:63	Configuration, 4th expansion module, left	
0x2004:64	Configuration, 5th expansion module, left	
0x2004:65	Configuration, 6th expansion module left	
0x2004:66 ... 0x2004:80	Reserved	

4.2.2.6 SDO Index 0x2005

This index contains the element types

Index (hex)	Content	Example/explanation
0x2005:01	Element type. Element ID = 1	Element with ID = 1: Single-pole semiconductor output with feedback loop Sub-Index 1: 51 hex See List containing the element types [📖 173] in the Appendix
...	...	
0x2005:64	Element type. Element ID = 100	
0x2005:65 ... 0x2005:80	Reserved	

4.3 PNOZ mc6p, PNOZ mc6.1p, PNOZ mmc6p, PNOZ mc12p (SDO)

4.3.1 Overview

All the CANopen objects (variables and parameters) that are relevant for these units are entered in the CANopen object directory. Service Data Objects (SDOs) are used for read and write access. The object directory is available as an EDS file (Electronic Data Sheet), enabling the PNOZ mc6p fieldbus module to be incorporated easily into a CANopen network.

The manufacturer-specific part of the object directory is structured as follows:

Index	Content
2000	Output data
2001	Diagnostic word (Low Byte)
2002	Diagnostic word (High Byte)
2003	Status of inputs
	Status of input LED
	State of outputs
	Status of LED
2004	Configuration
2005	Element types
2100	Input data

Information

Data with indices 2001 to 2003 is only updated by the PNOZmulti piece by piece in each cycle. This may mean that interdependent data is inconsistent. Updating all of the data can take up to 500 ms.

4.3.2 System requirements

Communication via SDOs is only possible with units from the stated version number onwards:

- PNOZ mc6p from Version 1.1
- PNOZ mc6.1p, PNOZ mmc6p from Version 1.0
- PNOZ m1p from Version 4.0
- All other PNOZmulti base units from Version 1.0

4.3.3 Object Directory

4.3.3.1 Index 2000

This index contains the output data

Sub-Index (dec)	Content	Example/explanation
1	Outputs Bit 0 … 7 fieldbus module	For information on the sub-indices please see chapter entitled "Communication with fieldbus systems"
2	Outputs Bit 8 … 15 fieldbus module	
3	Outputs Bit 16 … 23 fieldbus module	
4	LED status	
5	Table number	
6	Segment number	
7	Byte 0	
8	Byte 1	
9	Byte 2	
10	Byte 3	
11	Byte 4	
12	Byte 5	
13	Byte 6	
14	Byte 7	
15	Byte 8	
16	Byte 9	
17	Byte 10	
18	Byte 11	
19	Byte 12	
20 ... 63	Reserved	
64	i0 ... i7 Safe Ethernet connection	Inputs on the safe Ethernet connection
65	i8 ... i15 Safe Ethernet connection	
66	i16 ... i23 Safe Ethernet connection	
67	i24 ... i31 Safe Ethernet connection	
68	i32 ... i39 Safe Ethernet connection	
69	i40 ... i47 Safe Ethernet connection	
70 ... 71	Reserved	

Sub-Index (dec)	Content	Example/explanation							
72	o0 ... o7 Safe Ethernet connection	Outputs on the safe Ethernet connection							
73	o8 ... o15 Safe Ethernet connection								
74	o16 ... o23 Safe Ethernet connection								
75	o24 ... o31 Safe Ethernet connection								
76	o32 ... o39 Safe Ethernet connection								
77	o40 ... o47 Safe Ethernet connection								
78 …79	Reserved								
80	I0 ... I7 1st expansion module, left	Virtual inputs on the 2nd link module PNOZ mI1p:							
81	I8 ... I15 1st expansion module, left								
82	I16 ... I23 1st expansion module, left	Sub-Index 84:							
83	I24 ... I31 1st expansion module, left	I7	I6	I5	I4	I3	I2	I1	I0
84	I0 ... I7 2nd expansion module, left	Sub-Index 85:							
85	I8 ... I15 2nd expansion module, left	I15	I14	I13	I12	I11	I10	I9	I8
86	I16 ... I23 2nd expansion module, left	Sub-Index 86:							
87	I24 ... I31 2nd expansion module, left	I23	I22	I21	I20	I19	I18	I17	I16
88	I0 ... I7 3rd expansion module, left	Sub-Index 87:							
89	I8 ... I15 3rd expansion module, left	I31	I30	I29	I28	I27	I26	I25	I24
90	I16 ... I23 3rd expansion module, left	If an output has a high signal, the corresponding Bit will contain "1"; if the output is open (low signal), the Bit will contain "0".							
91	I24 ... I31 3rd expansion module, left								
92	I0 ... I7 4th expansion module, left								
93	I8 ... I15 4th expansion module, left								
94	I16 ... I23 4th expansion module, left								
95	I24 ... I31 4th expansion module, left								
96	I0 ... I7 5th expansion module, left								

Sub-Index (dec)	Content	Example/explanation							
97	I8 ... I15 5th expansion module, left								
98	I16 ... I23 5th expansion module, left								
99	I24 ... I31 5th expansion module, left								
100	I0 ... I7 6th expansion module, left								
101	I8 ... I15 6th expansion module, left								
102	I16 ... I23 6th expansion module, left								
103	I24 ... I31 6th expansion module, left								
104	O0 ... O7 1st expansion module, left	Virtual outputs on the 3rd link module PNOZ ml1p:							
105	O8 ... O15 1st expansion module, left								
106	O16 ... O23 1st expansion module, left								
107	O24 ... O31 1st expansion module, left								
108	O0 ... O7 2nd expansion module, left								
109	O8 ... O15 2nd expansion module, left								
110	O16 ... O23 2nd expansion module, left								
111	O24 ... O31 2nd expansion module, left								
112	O0 ... O7 3rd expansion module, left	Sub-Index 112:							
113	O8 ... O15 3rd expansion module, left	O7	O6	O5	O4	O3	O2	O1	O0
114	O16 ... O23 3rd expansion module, left	Sub-Index 113:							
115	O24 ... O31 3rd expansion module, left	O15	O14	O13	O12	O11	O10	O9	O8
116	O0 ... O7 4th expansion module left	Sub-Index 114:							
117	O8 ... O15 4th expansion module, left	O23	O22	O21	O20	O19	O18	O17	O16
118	O16 ... O23 4th expansion module, left	Sub-Index 115:							
119	O24 ... O31 4th expansion module, left	O31	O30	O29	O28	O27	O26	O25	O24

Sub-Index (dec)	Content	Example/explanation
120	O0 ... O7 5th expansion module, left	If an output has a high signal, the corresponding Bit will contain "1"; if the output is open (low signal), the Bit will contain "0".
121	O8 ... O15 5th expansion module, left	
122	O16 ... O23 5th expansion module, left	
123	O24 ... O31 5th expansion module, left	
124	O0 ... O7 6th expansion module, left	
125	O8 ... O15 6th expansion module, left	
126	O16 ... O23 6th expansion module, left	
127	O24 ... O31 6th expansion module, left	
128	Reserved	

4.3.3.2 Index 2001 and 2002

This index contains the diagnostic words and the output bits for the Element IDs.

Index (hex) 2001:

<table>
<tr><th>Sub-Index (dec)</th><th>Contents</th><th>Example/explanation</th></tr>
<tr><td>1</td><td>Low Byte diagnostic word.
Element ID=1</td><td rowspan="4">The diagnostic word is displayed in the PNOZmulti Configurator and on the PVIS expanded diagnostics (see chapter entitled Diagnostic word [128] and the online help for the PNOZmulti Configurator)
Element ID = 1,
e.g. diagnostic word of E-STOP:
Low Byte:
0 | 0 | 0 | 0 | 0 | 0 | 1 | 0
Message: Pushbutton operated</td></tr>
<tr><td>...</td><td></td></tr>
<tr><td rowspan="2">100</td><td rowspan="2">Low Byte diagnostic word.
Element ID=100</td></tr>
</table>

<table>
<tr><th>Sub-Index (dec)</th><th>Contents</th><th>Example/explanation</th></tr>
<tr><td>101
...113</td><td>Output Bits of Element ID = 1 ... 100</td><td>Each element is assigned an ID in the PNOZmulti Configurator. If the element's output = 0 (no enable), the corresponding bit is set.
<table>
<tr><td>Sub Index</td><td colspan="8">Element ID</td></tr>
<tr><td>101</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>102</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td></tr>
<tr><td>103</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td></tr>
<tr><td colspan="9"></td></tr>
<tr><td>111</td><td>88</td><td>87</td><td>86</td><td>85</td><td>84</td><td>83</td><td>82</td><td>81</td></tr>
<tr><td>112</td><td>96</td><td>95</td><td>94</td><td>93</td><td>92</td><td>91</td><td>90</td><td>89</td></tr>
<tr><td>113</td><td>-</td><td>-</td><td>-</td><td>-</td><td>100</td><td>99</td><td>98</td><td>97</td></tr>
</table></td></tr>
<tr><td>114
...128</td><td>Reserved</td><td></td></tr>
</table>

Index 2002:

<table>
<tr><th>Sub-Index (dec)</th><th>Contents</th><th>Example/explanation</th></tr>
<tr><td>1</td><td>High Byte diagnostic word.
Element ID=1</td><td>See Index 2001 for comment
Element ID = 1,
e.g. diagnostic word of E-STOP:
High Byte:
<table>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
</table>
Message: Wiring error, clock error</td></tr>
<tr><td>...</td><td>...</td><td></td></tr>
<tr><td>100</td><td>High Byte diagnostic word.
Element ID=100</td><td></td></tr>
<tr><td>101...128</td><td>Reserved</td><td></td></tr>
</table>

4.3.3.3 Index 2003

This index contains the state of the inputs/outputs plus the LED status

<table>
<tr><th>Sub-Index (dec)</th><th>Content</th><th colspan="8">Example/Comment</th></tr>
<tr><td>1</td><td>I0 ... I7 base unit,
IM0 ... I7 base unit Mini</td><td colspan="8" rowspan="4">For example: The safety system consists of one base unit PNOZ m1p and one expansion module PNOZ mi1p
Assignment of Bytes on the base units PNOZmulti</td></tr>
<tr><td>2</td><td>I8 ... I15 base unit,
I8 ... I15 base unit Mini</td></tr>
<tr><td>3</td><td>I16 ... I19 base unit
IM16 ... IM19 base unit Mini</td></tr>
<tr><td>4</td><td>0</td></tr>
<tr><td>5</td><td>0</td><td colspan="8">For example: The safety system consists of one base unit PNOZ m1p and one expansion module PNOZ mi1p
Assignment of Bytes on the base units PNOZmulti</td></tr>
<tr><td>6</td><td>I0 ... I7 1st expansion module, right</td><td colspan="8">Sub-Index 1: PNOZ m1p</td></tr>
<tr><td>7</td><td>I0 ... I7 2nd expansion module, right</td><td>I7</td><td>I6</td><td>I5</td><td>I4</td><td>I3</td><td>I2</td><td>I1</td><td>I0</td></tr>
<tr><td>8</td><td>I0 ... I7 3rd expansion module, right</td><td colspan="8">Sub-Index 2: PNOZ m1p</td></tr>
<tr><td>9</td><td>I0 ... I7 4th expansion module, right</td><td>I15</td><td>I14</td><td>I13</td><td>I12</td><td>I11</td><td>I10</td><td>I9</td><td>I8</td></tr>
<tr><td>10</td><td>I0 ... I7 5th expansion module, right</td><td colspan="8">Sub-Index 3: PNOZ m1p</td></tr>
<tr><td>11</td><td>I0 ... I7 6th expansion module, right</td><td>0</td><td>0</td><td>0</td><td>0</td><td>I19</td><td>I18</td><td>I17</td><td>I16</td></tr>
<tr><td>12</td><td>I0 ... I7 7th expansion module, right</td><td colspan="8">Sub-Index 4:</td></tr>
<tr><td rowspan="8">13</td><td rowspan="8">I0 ... I7 8th expansion module right</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td colspan="8">Sub-Index 5:</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td colspan="8">Sub-Index 6: PNOZ mi1p</td></tr>
<tr><td>I7</td><td>I6</td><td>I5</td><td>I4</td><td>I3</td><td>I2</td><td>I1</td><td>I0</td></tr>
<tr><td colspan="8">If an input has a high signal, the corresponding bit will be "1"; if an input has a low signal, the bit will be "0".</td></tr>
</table>

Sub-Index (dec)	Content	Example/Comment							
		INFORMATION: On the base units PNOZmulti Mini, the status of the configurable inputs/outputs is only displayed if they are configured as inputs in the PNOZmulti Configurator. Assignment of Bytes on the base units PNOZmulti Mini:							
		Sub-Index 1: PNOZ mmxp							
		I7	I6	I5	I4	IM3	IM2	IM1	IM0
		Sub-Index 2: PNOZ mmxp							
		I15	I14	I13	I12	I11	I10	I9	I8
		Sub-Index 3: PNOZ mmxp							
		0	0	0	0	IM19	IM18	IM17	IM16
14 ... 16	Reserved								
17	LED I0 ... I7 base unit	For example: The safety system consists of one base unit PNOZ m1p and one expansion module PNOZ mi1p							
18	LED I8 ... I15 base unit								
19	LED I16 ... I19 base unit								
20	0	Sub-Index 17: PNOZ m1p							
21	0	I7	I6	I5	I4	I3	I2	I1	I0
22	LED I0 ... I7 1st expansion module, right	Sub-Index 18: PNOZ m1p							
		I15	I14	I13	I12	I11	I10	I9	I8
23	LED I0 ... I7 2nd expansion module, right	Sub-Index 19: PNOZ m1p							
		0	0	0	0	I19	I18	I17	I16
24	LED I0 ... I7 3rd expansion module, right	Sub-Index 20:							
		0	0	0	0	0	0	0	0
25	LED I0 ... I7 4th expansion module, right	Sub-Index 21							
		0	0	0	0	0	0	0	0
26	LED I0 ... I7 5th expansion module, right	Sub-Index 22: PNOZ mi1p							
		I7	I6	I5	I4	I3	I2	I1	I0
27	LED I0 ... I7 6th expansion module, right								
28	LED I0 ... I7 7th expansion module, right								
29	LED I0 ... I7 8th expansion module, right								
30 ... 32	Reserved								

33	IM0 ... IM3 base unit Mini	Assignment of Bytes depends on the unit: **E.g. Base unit PNOZ m1p**						
34	0							
35	IM16 ... T3M23 base unit Mini							
36	O0 ... O3 base unit							
37	O4 and O5 base unit							
38	O0 ... O7 1st expansion module, right	Sub-Index 36:						
39	O0 ... O7 2nd expansion module, right	0	0	1	1	O3	O2	O1	O0
40	O0 ... O7 3rd expansion module, right	Sub-Index 37:						
41	O0 ... O7 4th expansion module, right	0	0	0	0	0	0	O5	O4
42	O0 ... O7 5th expansion module, right	**PNOZ mo1p**						
43	O0 ... O7 6th expansion module, right	Sub-Index 38 ... 45:						
44	O0 ... O7 7th expansion module, right	0	0	0	0	O3	O2	O1	O0
45	O0 ... O7 8th expansion module, right	Sub-Index 54 ... 61:						
46 ... 48	Reserved	0	0	0	0	0	0	0	0
49 ... 53	0	**PNOZ mo2p, PNOZ mo3p**						
54	O8 ... O15 1st expansion module, right	Sub-Index 38 ... 45:						
55	O8 ... O15 2nd expansion module, right	0	0	0	0	0	0	O1	O0
56	O8 ... O15 3rd expansion module, right	Sub-Index 54 ... 61:						
57	O8 ... O15 4th expansion module, right	0	0	0	0	0	0	0	0
58	O8 ... O15 5th expansion module, right	**PNOZ mo4p, PNOZ mo5p**						
59	O8 ... O15 6th expansion module, right	Sub-Index 38 ... 45:						
60	O8 ... O15 7th expansion module, right	0	0	0	0	O3	O2	O1	O0

<table>
<tr><td rowspan="7">61</td><td rowspan="7">O8 ... O15 8th expansion module, right</td><td colspan="8">Sub-Index 54 ... 61:</td></tr>
<tr><td colspan="8">PNOZ mc1p</td></tr>
<tr><td colspan="8">Sub-Index 38 ... 45:</td></tr>
<tr><td>A7</td><td>A6</td><td>A5</td><td>A4</td><td>A3</td><td>A2</td><td>A1</td><td>A0</td></tr>
<tr><td colspan="8">Sub-Index 54 ... 61:</td></tr>
<tr><td>A15</td><td>A14</td><td>A13</td><td>A12</td><td>A11</td><td>A10</td><td>A9</td><td>A8</td></tr>
<tr><td colspan="8">If an output has a high signal, the corresponding Bit will contain "1"; if the output is open (low signal), the Bit will contain "0".</td></tr>
<tr><td rowspan="5"></td><td rowspan="5"></td><td colspan="8">INFORMATION:
On the base units PNOZmulti Mini, the status of the configurable inputs/outputs is only displayed if they are configured as outputs in the PNOZmulti Configurator.
Assignment of Bytes on the base units PNOZmulti Mini:</td></tr>
<tr><td colspan="8">Sub-Index 33:</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>IM3</td><td>IM2</td><td>IM1</td><td>IM0</td></tr>
<tr><td colspan="8">Sub-Index 35:</td></tr>
<tr><td>T3M23</td><td>T2M22</td><td>T1M21</td><td>T0;20</td><td>IM19</td><td>IM18</td><td>IM17</td><td>IM16</td></tr>
<tr><td>62 ... 64</td><td>Reserved</td><td colspan="8"></td></tr>
</table>

<table>
<tr><td>65</td><td>RUN</td><td rowspan="13">Depending on the LED status, the following Hex code will be in Sub-Index 65 ... 77:
00 hex: LED off
FF hex: LED on
30 hex: LED flashes</td></tr>
<tr><td>66</td><td>DIAG</td></tr>
<tr><td>67</td><td>FAULT</td></tr>
<tr><td>68</td><td>IFAULT</td></tr>
<tr><td>69</td><td>OFAULT</td></tr>
<tr><td>70</td><td>FAULT 1: Expansion module, right</td></tr>
<tr><td>71</td><td>FAULT 2: Expansion module, right</td></tr>
<tr><td>72</td><td>FAULT 3: Expansion module, right</td></tr>
<tr><td>73</td><td>FAULT 4: Expansion module, right</td></tr>
<tr><td>74</td><td>FAULT 5: Expansion module, right</td></tr>
<tr><td>75</td><td>FAULT 6: Expansion module, right</td></tr>
<tr><td>76</td><td>FAULT 7: Expansion module, right</td></tr>
<tr><td>77</td><td>FAULT 8: Expansion module, right</td></tr>
</table>

78	FAULT 1: Expansion module, left	Depending on the LED status, the following Hex code will be in Sub-Index 78 ... 83: 00 hex: LED off FF hex: LED on 30 hex: LED flashes
79	FAULT 2: Expansion module, left	
80	FAULT 3: Expansion module, left	
81	FAULT 4: Expansion module, left	
82	FAULT 5: Expansion module, left	
83	FAULT 6: Expansion module, left	
84 ... 128	Reserved	

4.3.3.4 Index 2004

This index contains the PNOZmulti's configuration data

Sub-Index (dec)	Contents	Example/explanation
1	Data transfer	Sub-Index 1: Bit 0 = 1: All configuration data has been downloaded to the fieldbus module
2	Reserved	
3	Number of elements	Number of configured elements with Element ID
4 ... 16	Reserved	
17 ... 20	Product number (hex)	Product number 733 100: 000BCBEC hex Sub-Index 17: 00, Sub-Index 18: 0B, Sub-Index 19: CB,Sub-Index 20: EC
21 ... 24	Unit version (hex)	Unit version 20: 14 hex Sub-Index 21: 00, Sub-Index 22: 00, Sub-Index 23: 00, Sub-Index 24: 14
25 … 28	Serial number (hex)	Serial number 123 456: 0001E240 hex. Sub-Index 25: 00, Sub-Index 26: 01, Sub-Index 27: E2,Sub-Index 28: 40
29 ... 30	Check sum safe (hex)	Check sum A1B2 hex: Sub-Index 29: A1, Sub-Index 30: B2
31 ... 32	Overall project check sum (hex)	Check sum 3C5A hex: Sub-Index 31: 3C, Byte 32: 5A
33 ... 36	Reserved	
37 ... 40	Project creation date (hex)	Creation date: 28.11.2003 Sub-Index 37: 1C, Sub-Index 38: 0B, Sub-Index 39: 07,Sub-Index 40: D3
41 ... 43	Reserved	
44	Configuration, fieldbus module/integrated interface	Sub-Index 44 contains the Hex code for a fieldbus module (installed on the left) or for inputs and outputs via the integr. interface (see Table 1 Seg 2 Byte 0)

Sub-Index (dec)	Contents	Example/explanation
45	Configuration, 1st expansion module, right	Sub-Index 45 ... 52 contains the Hex code for the expansion modules on the right: PNOZ mi1p: 08 PNOZ mi2p: 38 PNOZ mo1p: 18 PNOZ mo2p: 10 PNOZ mo3p: 30 PNOZ mo4p: 28 PNOZ mo5p: 48 PNOZ mc1p: 20 PNOZ ms3p: 68 PNOZ ms4p: 78 PNOZ ms1p/PNOZ ms2p: 88 PNOZ ms2p HTL: 58 PNOZ ms3p HTL: 64 PNOZsigma with one output: 11 PNOZsigma with two outputs: 22 No expansion module: 00
46	Configuration, 2nd expansion module, right	
47	Configuration, 3rd expansion module, right	
48	Configuration, 4th expansion module, right	
49	Configuration, 5th expansion module, right	
50	Configuration, 6th expansion module, right	
51	Configuration, 7th expansion module, right	
52	Configuration, 8th expansion module right	
53 ... 56	Reserved	
57	1st character (Low Byte)	Sub-Index 57 ... 88 contains the project name defined in the PNOZmultil Configurator under "Enter project data"; this is stored in UNICODE format, 2 Bytes contain the Hex code of the individual UNICODE characters.
58	1st character (High Byte)	
59	2nd character (Low Byte)	
60	2nd character (High Byte)	
61	3rd character (Low Byte)	
62	3rd character (High Byte)	
63	4th character (Low Byte)	
64	4th character (High Byte)	
65	5th character (Low Byte)	
66	5th character (High Byte)	
67	6th character (Low Byte)	
68	6th character (High Byte)	
69	7th character (Low Byte)	
70	7th character (High Byte)	
71	8th character (Low Byte)	
72	8th character (High Byte)	

Sub-Index (dec)	Contents	Example/explanation
73	9th character (Low Byte)	
74	9th character (High Byte)	
75	10th character (Low Byte)	
76	10th character (High Byte)	
77	11th character (Low Byte)	
78	11th character (High Byte)	
79	12th character (Low Byte)	
80	12th character (High Byte)	
81	13th character (Low Byte)	
82	13th character (High Byte)	
83	14th character (Low Byte)	
84	14th character (High Byte)	
85	15th character (Low Byte)	
86	15th character (High Byte)	
87	16th character (Low Byte)	
88	16th character (High Byte)	
89	Day	Date on which the program on the chip card was last modified Date modified : 28.11.2003 Sub-Index 89: 1C, Sub-Index 90: 0B, Sub-Index 91: 07, Sub-Index 92: D3 Time: 14 hours 25 minutes Sub-Index 93: 0E, Sub-Index 94: 19 Time zone 1: Sub-Index 95: 01
90	Month	
91	Year (High Byte)	
92	Year (Low Byte)	
93	Hour	
94	Minute	
95	Time zone	
96	Configuration, 1st expansion module, left	Sub-Index 96 ... 101 contains the Hex code for the expansion modules to the left of the base unit. Any fieldbus module in these Sub-Indices will not be considered (see Index 2004, Sub-Index 44). PNOZ ml1p: A8 PNOZ ml2p: C8 PNOZ ma1p: B8
97	Configuration, 2nd expansion module, left	
98	Configuration, 3rd expansion module, left	
99	Configuration, 4th expansion module, left	
100	Configuration, 5th expansion module, left	
101	Configuration, 6th expansion module left	
102 ... 128	Reserved	

4.3.3.5 Index 2005

This index contains the element types

Sub-Index (dec)	Content	Example/Comment
1	Element type. Element ID = 1	Element with ID = 1: Single-pole semiconductor output with feedback loop Sub-Index 1: 51 hex See list containing the element types in the Appendix
...	...	
100	Element type. Element ID = 100	
101 ... 128	Reserved	

4.3.3.6 Index 2100

This index contains the input data

Sub-Index (dec)	Content	Example/explanation
1	Inputs Bit 0 … 7	For information on the sub-indices see Input data (to the PNOZmulti) [13]
2	Inputs Bit 8 … 15	
3	Inputs Bit 16 … 23	
4	Reserved	
5	Table number	
6	Segment number	
7 ... 128	Reserved	

4.4 PNOZ mc8p Ethernet IP / Modbus TCP

4.4.1 Introduction

This chapter describes the special features of communication with the PNOZ mc8p expansion module on Ethernet IP and Modbus TCP. Acess to PNOZmulti data via tables and segments is described in Chapters Basics [13] and PNOZ mc2p, PNOZ mc2.1p, PNOZ mmc11p (SDO and PDO) [21].

4.4.2 Overview

The expansion module PNOZ mc8p connects the configurable control system PNOZmulti via Ethernet to control systems that support the Ethernet IP and Modbus TCP protocols. Ethernet IP and Modbus TCP are designed for fast data exchange at field level. The expansion module PNOZ mc8p is a passive Ethernet IP (adapter) or Modbus TCP (slave) subscriber. The basic functions of communication with Ethernet IP or Modbus TCP conform to IEEE 802.3. The central controller (master) reads input information from the slaves and writes output information to the slaves as part of each cycle. As well as the cyclical transfer of usable data, the PNOZ mc8p can also be used for diagnostics and commissioning functions.

4.4.3 Module features

- Can be configured in the PNOZmulti Configurator
- Network protocols: Ethernet IP, Modbus TCP
- Status indicators for communication and for errors
- Transmission rate 10 MBit/s (10BaseT) and 100 MBit/s (100BaseTX), full and half duplex
- Setting of the IP address using DIP switches on the front panel

4.4.4 Assigning the IP address to your PC

- Please refer to the operating instructions for your operating system for the procedure.
- Set the IP address, e.g. 192.168.0.1, with subnet mask 255.255.255.0.

4.4.5 Setting the IP address of the expansion module

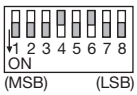
The IP address of the PNOZ mc8p is set using DIP switches on the front panel.
Please note: Only set the IP address when the supply voltage is switched off.

The first three bytes of the IP address are:

- IP address: 192.168.0
- Subnet mask: 255.255.255.0

The last byte is configured using the DIP switches.
Value range: 1 ... 255
Note: The IP address of the PNOZ mc8p should not be the same as the IP address for the PC.

Example: DIP switch: 00010100 (20 decimal)



IP address: 192.168.0.20

Once the IP address has been set via the DIP switches you can connect the supply voltage to the base unit.

4.4.6 Changing IP settings

You can change the IP settings of the PNOZ mc8p after configuring the IP addresses of the PC and PNOZ mc8p.

- Connect the PNOZ mc8p to the PC.
- Call up the following html page: http://192.168.0.20/config.htm.
- Configure the settings for the PNOZ mc8p.
  Example: IP address: 172.16.216.139
  Subnet mask: 255.255.0.0
  Gateway address:--
  DNS1 address:--
  DNS1 address:--

Host name:---
Domain name:--
SMTP server:--
DHCP enabled:No

- Click the *Store Configuration* button. The settings are transferred to the expansion module.
- Switch off the supply voltage.
- Set all DIP switches to zero.
- Switch on the supply voltage. The new IP address for the unit is now set.

4.4.7 Data exchange

Twenty bytes always must be sent and received for communication with the PNOZmulti.

4.4.7.1 Ethernet IP

The input/output data from the PNOZmulti can be polled using the assembly object (Class 04h).
Data from the PNOZmulti are requested with instance 64h.
Instance 96h writes the data from the Ethernet IP scanner to the PNOZmulti.

4.4.7.2 Modbus TCP

No connection needs to be configured for the PNOZ mc8p. Port 502 is used as required by the Modbus TCP specification.
Modbus TCP supports the following function codes:

Function code	Function name
1	Read coils
2	Read input discretes
3	Read multiple registers
4	Read input registers
5	Write coil
6	Write single register
7	Read exception status
15	Force multiple coils
16	Force multiple registers
22	Mask write register
23	Read/Write registers

The address input range begins with Register 0. The address output range begins with Register 1024. The byte sequence of a word is high byte/low byte

Word	
Left byte	**Right byte**
Low Byte (Bit 07 ... 00)	High Byte (Bit 15 ... 08)

Error codes on Modbus TCP

Code	Name	Description
01	Invalid function	The PNOZ mc8p does not support the function code in the enquiry.
02	Invalid data address	The data address received in the enquiry is outside the memory range.
03	Invalid data	Invalid data requested.

4.4.8 Web interface for commissioning and testing

A Pilz web interface can be used when commissioning or as a testing aid. It can be used to poll data from the PNOZmulti.

- Commission a base unit and PNOZ mc8p as described in the operating instructions.
- Connect the PNOZ mc8p to the PC.
- Enter the IP address (URL) in your browser's address bar,
  e.g.: http://172.16.216.139
- The input mask provides access to the inputs and outputs on the PNOZmulti system and to the table segments.

4.4.9 Restricting access

In principle, each Ethernet subscriber can set up a connection to the PNOZ mc8p. This access can be restricted.

- To establish a connection to the FTP site, enter the IP address (URL) of the PNOZ mc8p in your browser's address bar.
  A login window appears.
- Log in to gain access to the user range of the PNOZ mc8p.
  The default access data is: User name: User
  Password: Password
- Save the file ip_access.cfg on your PC and open it using an editor. Once opened the file contains the following information:

  [MODBUS/TCP]

  ****.*.*.****

  [Ethernet/IP]

  ****.*.*.****

  If ****.*.*.**** is entered, all subscribers will have unrestricted access.
- Instead of the characters ****.*.*.****, enter the IP addresses of the subscribers to which you wish to grant restricted access, e.g.:

  [MODBUS/TCP]

  172.16.205.24

  172.16.205.40

  [Ethernet/IP]

  172.16.205.96

- Save the file ip_access.cfg on your PC.
- Download the file to the PNOZ mc8p.
- Restart the PNOZmulti.

4.4.10 Input and output data

The data is structured as follows:

Input area

The inputs are defined in the master and transferred to the PNOZmulti. Each input has a number, e.g. input bit 4 of byte 1 has the number i12.

Output area

The outputs are defined in the PNOZmulti Configurator. Each output that is used is given a number there, e.g. o0, o5... .
The status of output o0 is stored in bit 0 of byte 0; the status of output o5 is stored in bit 5 of byte 0 etc.

Output area only: Byte 3

- Bits 0 … 4: Status of LEDs on the PNOZmulti
  - Bit 0: OFAULT
  - Bit 1: IFAULT
  - Bit 2: FAULT
  - Bit 3: DIAG
  - Bit 4: RUN
- Bit 5: Data is being exchanged.

Information

Please refer to the chapter entitled "Basics", under Input data (to the PNOZmulti) [📖 13] / Output data (from the PNOZmulti) [📖 13]

4.4.10.1 Assigning the inputs/outputs in the PNOZmulti Configurator to the Ethernet IP/ Modbus TCP input/output data

Multi Configurator inputs	I0 ... I7	I8 ... I15	I16 ... I23
Ethernet IP or Modbus TCP input data	Byte 0: Bit 0 ... 7	Byte 1 :Bit 0 ... 7	Byte 2 :Bit 0 ... 7
PNOZmulti Configurator outputs	O0 ... O7	O8 ... O15	O16 ... O23
Ethernet IP or Modbus TCP output data	Byte 0 : Bit 0 ... 7	Byte 1 :Bit 0 ... 7	Byte 2 :Bit 0 ... 7

4.5 PNOZ mc10p sercos III

4.5.1 Overview

The data from the PNOZmulti is stored in a buffer. There is read/write access to the input data (Byte 2048 to 2067); other data is read-only.

The object buffer is structured as follows:

Byte	Content
0 - 19	Output data
79 - 127	I/O expansion modules, left
128 - 255	Diagnostic word (Low Byte)
256 - 383	Diagnostic word (High Byte)
384 - 511	Status of inputs
	Status of input LED
	Status of outputs
	Status of LED
512 - 639	Configuration
640 - 767	Element types
2048 - 2067	Input data
2112 - 2117	Diagnostic data

Information

The first 20 input/output bytes are transferred cyclically by the PNOZmulti. The rest of the data is updated piece by piece in each cycle. This may mean that interdependent data is inconsistent. Updating all of the data can take up to 500 ms

4.5.2 System requirements

Communication via sercos III is only possible with units from the stated version number onwards:

- PNOZ mc10p from Version 1
- Base units PNOZ mXp from Version 6.5

4.5.3 Object buffer

4.5.3.1 Output data

These bytes contain the output data

Byte	Content	Example/explanation
0	Outputs Bit 0 … 7 fieldbus module	
1	Outputs Bit 8 … 15 fieldbus module	
2	Outputs Bit 16 … 23 fieldbus module	
3	LED status	
4	Table number	
5	Segment number	
6	Byte 0 of Table x, Segment y	
7	Byte 1 of Table x, Segment y	
8	Byte 2 of Table x, Segment y	
9	Byte 3 of Table x, Segment y	
10	Byte 4 of Table x, Segment y	
11	Byte 5 of Table x, Segment y	
12	Byte 6 of Table x, Segment y y	
13	Byte 7 of Table x, Segment y	
14	Byte 8 of Table x, Segment y	
15	Byte 9 of Table x, Segment y	
16	Byte 10 of Table x, Segment y	
17	Byte 11 of Table x, Segment y	
18	Byte 12 of Table x, Segment y	
19 ... 62	Reserved	
63	i0 ... i7 Safe Ethernet connection	Inputs on the safe Ethernet connection
64	i8 ... i15 Safe Ethernet connection	
65	i16 ... i23 Safe Ethernet connection	
66	i24 ... i31 Safe Ethernet connection	
67	i32 ... i39 Safe Ethernet connection	
68	i40 ... i47 Safe Ethernet connection	
69 ... 70	Reserved	
71	o0 ... o7 Safe Ethernet connection	Inputs on the safe Ethernet connection
72	o8 ... o15 Safe Ethernet connection	
73	o16 ... o23 Safe Ethernet connection	
74	o24 ... o31 Safe Ethernet connection	
75	o32 ... o39 Safe Ethernet connection	
76	o40 ... o47 Safe Ethernet connection	
77 … 78	Reserved	

Byte	Content	Example/explanation							
79	I0 ... I7 1st expansion module, left	Virtual inputs on the 2nd link module PNOZ ml1p:							
80	I8 ... I15 1st expansion module, left								
81	I16 ... I23 1st expansion module, left	Byte 83:							
82	I24 ... I31 1st expansion module, left	I7	I6	I5	I4	I3	I2	I1	I0
83	I0 ... I7 2nd expansion module, left	Byte 84:							
84	I8 ... I15 2nd expansion module, left	I15	I14	I13	I12	I11	I10	I9	I8
85	I16 ... I23 2nd expansion module, left	Byte 85:							
86	I24 ... I31 2nd expansion module, left	I23	I22	I21	I20	I19	I18	I17	I16
87	I0 ... I7 3rd expansion module, left	Byte 86:							
88	I8 ... I15 3rd expansion module, left	I31	I30	I29	I28	I27	I26	I25	I24
89	I16 ... I23 3rd expansion module, left	If an output has a high signal, the corresponding Bit will contain "1"; if the output is open (low signal), the Bit will contain "0".							
90	I24 ... I31 3rd expansion module, left								
91	I0 ... I7 4th expansion module, left								
92	I8 ... I15 4th expansion module, left								
93	I16 ... I23 4th expansion module, left								
94	I24 ... I31 4th expansion module, left								
95	I0 ... I7 5th expansion module, left								
96	I8 ... I15 5th expansion module, left								
97	I16 ... I23 5th expansion module, left								
98	I24 ... I31 5th expansion module, left								
99	I0 ... I7 6th expansion module, left								
100	I8 ... I15 6th expansion module, left								
101	I16 ... I23 6th expansion module, left								
102	I24 ... I31 6th expansion module, left								
103	O0 ... O7 1st expansion module, left	Virtual outputs on the 3rd link module PNOZ ml1p:							
104	O8 ... O15 1st expansion module, left								
105	O16 ... O23 1st expansion module, left								
106	O24 ... O31 1st expansion module, left								
107	O0 ... O7 2nd expansion module, left								
108	O8 ... O15 2nd expansion module, left								
109	O16 ... O23 2nd expansion module, left								
110	O24 ... O31 2nd expansion module, left								
111	O0 ... O7 3rd expansion module, left	Byte 111:							
112	O8 ... O15 3rd expansion module, left	O7	O6	O5	O4	O3	O2	O1	O0
113	O16 ... O23 3rd expansion module, left	Byte 112:							
114	O24 ... O31 3rd expansion module, left	O15	O14	O13	O12	O11	O10	O9	O8
115	O0 ... O7 4th expansion module left	Byte 113:							
116	O8 ... O15 4th expansion module, left	O23	O22	O21	O20	O19	O18	O17	O16

Byte	Content	Example/explanation							
117	O16 ... O23 4th expansion module, left	Byte 114:							
118	O24 ... O31 4th expansion module, left	O31	O30	O29	O28	O27	O26	O25	O24
119	O0 ... O7 5th expansion module, left	If an output has a high signal, the corresponding Bit will contain "1"; if the output is open (low signal), the Bit will contain "0".							
120	O8 ... O15 5th expansion module, left								
121	O16 ... O23 5th expansion module, left								
122	O24 ... O31 5th expansion module, left								
123	O0 ... O7 6th expansion module, left								
124	O8 ... O15 6th expansion module, left								
125	O16 ... O23 6th expansion module, left								
126	O24 ... O31 6th expansion module, left								
127	Reserved								

4.5.3.2 Diagnostic word

The following bytes contain the diagnostic words and the output bits for the Element IDs.

<table>
<tr><th>Byte</th><th>Content</th><th>Example/explanation</th></tr>
<tr><td>128</td><td>Low Byte diagnostic word.
Element ID=1</td><td rowspan="3">The diagnostic word is displayed in the PNOZmulti Configurator and on the PVIS expanded diagnostics (see chapter entitled Diagnostic word [📖 128] and the online help for the PNOZmulti Configurator)
Element ID = 1,
e.g. diagnostic word of E-STOP:
Low Byte:
<table>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">Message: Pushbutton operated</td></tr>
</table></td></tr>
<tr><td>...</td><td></td></tr>
<tr><td>227</td><td>Low Byte diagnostic word.
Element ID=100</td></tr>
<tr><td>228 ... 240</td><td>Output Bits of Element ID = 1 ... 100</td><td>Each element is assigned an ID in the PNOZmulti Configurator. If the element's output = 0 (no enable), the corresponding bit is set.
<table>
<tr><td>Sub Index</td><td colspan="8">Element ID</td></tr>
<tr><td>101</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>102</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td></tr>
<tr><td>103</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td></tr>
<tr><td colspan="9"></td></tr>
<tr><td>111</td><td>88</td><td>87</td><td>86</td><td>85</td><td>84</td><td>83</td><td>82</td><td>81</td></tr>
<tr><td>112</td><td>96</td><td>95</td><td>94</td><td>93</td><td>92</td><td>91</td><td>90</td><td>89</td></tr>
<tr><td>113</td><td>-</td><td>-</td><td>-</td><td>-</td><td>100</td><td>99</td><td>98</td><td>97</td></tr>
</table></td></tr>
<tr><td>241 ... 255</td><td>Reserved</td><td></td></tr>
</table>

Byte	Content	Example/explanation
256	High Byte diagnostic word. Element ID=1	See diagnostic word for comment Element ID = 1, e.g. diagnostic word of E-STOP: High Byte:
...	...	
355	High Byte diagnostic word. Element ID=100	0 0 0 0 0 0 0 1 Message: Wiring error, clock error
356 ... 383	Reserved	

4.5.3.3 Status of the inputs, outputs and LEDs

These bytes contain the state of the inputs/outputs plus the LED status

Byte	Content	Example/Comment
384	I0 ... I7 base unit, IM0 ... I7 base unit Mini	For example: The safety system consists of one base unit PNOZ m1p and one expansion module PNOZ mi1p Assignment of Bytes on the base units PNOZmulti
385	I8 ... I15 base unit, I8 ... I15 base unit Mini	
386	I16 ... I19 base unit IM16 ... IM19 base unit Mini	
387	0	
388	0	
389	I0 ... I7 1st expansion module, right	Byte 384: PNOZ m1p
390	I0 ... I7 2nd expansion module, right	I7 I6 I5 I4 I3 I2 I1 I0
391	I0 ... I7 3rd expansion module, right	Byte 385: PNOZ m1p
392	I0 ... I7 4th expansion module, right	I15 I14 I13 I12 I11 I10 I9 I8
393	I0 ... I7 5th expansion module, right	Byte 386: PNOZ m1p
394	I0 ... I7 6th expansion module, right	0 0 0 0 I19 I18 I17 I16
395	I0 ... I7 7th expansion module, right	Byte 387:

Byte	Content	Example/Comment							
396	I0 ... I7 8th expansion module right	0	0	0	0	0	0	0	0
		Byte 388:							
		0	0	0	0	0	0	0	0
		Byte 389: PNOZ mi1p							
		I7	I6	I5	I4	I3	I2	I1	I0
		If an input has a high signal, the corresponding bit will be "1"; if an input has a low signal, the bit will be "0".							
		INFORMATION: On the base units PNOZmulti Mini, the status of the configurable inputs/outputs is only displayed if they are configured as inputs in the PNOZmulti Configurator. Assignment of Bytes on the base units PNOZmulti Mini:							
		Byte 384: PNOZ mmxp							
		I7	I6	I5	I4	IM3	IM2	IM1	IM0
		Byte 385: PNOZ mmxp							
		I15	I14	I13	I12	I11	I10	I9	I8
		Byte 386: PNOZ mmxp							
		0	0	0	0	IM 19	IM 18	IM 17	IM 16
397 ... 399	Reserved								
400	LED I0 ... I7 base unit	For example: The safety system consists of one base unit PNOZ m1p and one expansion module PNOZ mi1p							
401	LED I8 ... I15 base unit								
402	LED I16 ... I19 base unit								
403	0	Byte 400: PNOZ m1p							
404	0	I7	I6	I5	I4	I3	I2	I1	I0
405	LED I0 ... I7 1st expansion module, right	Byte 401: PNOZ m1p							
		I15	I14	I13	I12	I11	I10	I9	I8
406	LED I0 ... I7 2nd expansion module, right	Byte 402: PNOZ m1p							
		0	0	0	0	I19	I18	I17	I16
407	LED I0 ... I7 3rd expansion module, right	Byte 403:							
		0	0	0	0	0	0	0	0
408	LED I0 ... I7 4th expansion module, right	Byte 404:							
		0	0	0	0	0	0	0	0
409	LED I0 ... I7 5th expansion module, right	Byte 405: PNOZ mi1p							
		I7	I6	I5	I4	I3	I2	I1	I0

Byte	Content	Example/Comment
410	LED I0 ... I7 6th expansion module, right	
411	LED I0 ... I7 7th expansion module, right	
412	LED I0 ... I7 8th expansion module, right	
413 ... 415	Reserved	

Byte	Content	Example/Comment							
416	IM0 ... IM3 base unit Mini	Assignment of Bytes depends on the unit: **E.g. Base unit PNOZ m1p**							
417	0								
418	IM16 ... T3M23 base unit Mini								
419	O0 ... O3 base unit								
420	O4 and O5 base unit								
421	O0 ... O7 1st expansion module, right	Byte 419:							
422	O0 ... O7 2nd expansion module, right	0	0	1	1	O3	O2	O1	O0
423	O0 ... O7 3rd expansion module, right	Byte 420:							
424	O0 ... O7 4th expansion module, right	0	0	0	0	0	0	O5	O4
425	O0 ... O7 5th expansion module, right	**PNOZ mo1p**							
426	O0 ... O7 6th expansion module, right	Byte 421 ... 428:							
427	O0 ... O7 7th expansion module, right	0	0	0	0	O3	O2	O1	O0
428	O0 ... O7 8th expansion module, right	Byte 437 ... 444:							
429 ... 431	Reserved	0	0	0	0	0	0	0	0
432 ... 436	0	**PNOZ mo2p, PNOZ mo3p**							
437	O8 ... O15 1st expansion module, right	Byte 421 ... Byte 428:							
438	O8 ... O15 2nd expansion module, right	0	0	0	0	0	0	O1	O0
439	O8 ... O15 3rd expansion module, right	Byte 437 ... Byte 444:							
440	O8 ... O15 4th expansion module, right	0	0	0	0	0	0	0	0
441	O8 ... O15 5th expansion module, right	**PNOZ mo4p, PNOZ mo5p**							
442	O8 ... O15 6th expansion module, right	Byte 421 ... 428:							

Byte	Content	Example/Comment							
443	O8 ... O15 7th expansion module, right	0	0	0	0	O3	O2	O1	O0
444	O8 ... O15 8th expansion module, right	Byte 437 ... Byte 444:							
		PNOZ mc1p							
		Byte 421 ... 428:							
		A7	A6	A5	A4	A3	A2	A1	A0
		Byte 437 ... Byte 444:							
		A15	A14	A13	A12	A11	A10	A9	A8
		If an output has a high signal, the corresponding Bit will contain "1"; if the output is open (low signal), the Bit will contain "0".							
		INFORMATION: On the base units PNOZmulti Mini, the status of the configurable inputs/outputs is only displayed if they are configured as outputs in the PNOZmulti Configurator. Assignment of Bytes on the **base units PNOZmulti Mini:**							
		Byte 416:							
		0	0	0	0	IM3	IM2	IM1	IM0
		Byte 418:							
		T3 M23	T2 M22	T1 M21	T0 M20	IM19	IM18	IM17	IM16
445 ... 447	Reserved								

448	RUN	Depending on the LED status, the following Hex code will be in Byte 448 ... 460: 00 hex: LED off FF hex: LED on 30 hex: LED flashes
449	DIAG	
450	FAULT	
451	IFAULT	
452	OFAULT	
453	FAULT 1st expansion module, right	
454	FAULT 2nd expansion module, right	
455	FAULT 3rd expansion module, right	
456	FAULT 4th expansion module, right	
457	FAULT 5th expansion module, right	
458	FAULT 6th expansion module, right	
459	FAULT 7th expansion module, right	
460	FAULT 8th expansion module, right	

461	FAULT 1st expansion module, left	Depending on the LED status, the following Hex code will be in Byte 461 ... 466: 00 hex: LED off FF hex: LED on 30 hex: LED flashes
462	FAULT 2nd expansion module, left	
463	FAULT 3rd expansion module, left	
464	FAULT 4th expansion module, left	
465	FAULT 5th expansion module, left	
466	FAULT 6th expansion module, left	
467 ... 511	Reserved	

4.5.3.4 Configuration

These bytes contain the PNOZmulti's configuration data

Byte	Contents	Example/Comment
512	Data transfer	Byte 512 = 1: Bit 1 = 1: All configuration data has been downloaded to the fieldbus module
513	Reserved	
514	Number of elements	Number of configured elements with Element ID
515 ... 527	Reserved	
528 ... 531	Product number (hex)	Product number 733 100: 000BCBEC hex Byte 528: 00, Byte 529: 0B, Byte 530: CB,Byte 531: EC
532 ... 535	Unit version (hex)	Unit version 20: 14 hex Byte 532: 00, Byte 533: 00, Byte 534: 00,Byte 535: 14
536 … 539	Serial number (hex)	Serial number 123 456: 0001E240 hex. Byte 536: 00, Byte 537: 01, Byte 538: E2,Byte 539: 40
540 ... 541	Check sum safe (hex)	Check sum A1B2 hex: Byte 540: A1, Byte 541: B2
542 ... 543	Overall project check sum (hex)	Check sum 3C5A hex: Byte 542: 3C, Byte 543: 5A
544 ... 547	Reserved	
548 ... 551	Project creation date (hex)	Creation date: 28.11.2003 Byte 548: 1C, Byte 549: 0B, Byte 550: 07,Byte 551: D3
552 ... 554	Reserved	
555	Configuration, fieldbus module/integrated interface	Byte 555 contains the Hex code for a fieldbus module (installed on the left) or for inputs and outputs via the integr. interface (see Table 1 Seg 2 Byte 0)

Byte	Contents	Example/Comment
556	Configuration, 1st expansion module, right	Byte 556 ... 563 contains the Hex code for the expansion modules on the right: PNOZ mi1p: 08 PNOZ mi2p: 38 PNOZ mo1p: 18 PNOZ mo2p: 10 PNOZ mo3p: 30 PNOZ mo4p: 28 PNOZ mo5p: 48 PNOZ mc1p: 20 PNOZ ms3p: 68 PNOZ ms4p: 78 PNOZ ms1p/PNOZ ms2p: 88 PNOZ ms2p HTL: 58 PNOZ ms3p HTL: 64 PNOZsigma with one output: 11 PNOZsigma with two outputs: 22 No expansion module: 00
557	Configuration, 2nd expansion module, right	
558	Configuration, 3rd expansion module, right	
559	Configuration, 4th expansion module, right	
560	Configuration, 5th expansion module, right	
561	Configuration, 6th expansion module, right	
562	Configuration, 7th expansion module, right	
563	Configuration, 8th expansion module right	
564 ... 567	Reserved	
568	1st character (Low Byte)	Byte 568 ... 599 contains the project name defined in the PNOZmultil Configurator under "Enter project data"; this is stored in UNICODE format, 2 Bytes contain the Hex code of the individual UNICODE characters.
569	1st character (High Byte)	
570	2nd character (Low Byte)	
571	2nd character (High Byte)	
572	3rd character (Low Byte)	
573	3rd character (High Byte)	
574	4th character (Low Byte)	
575	4th character (High Byte)	
576	5th character (Low Byte)	
577	5th character (High Byte)	
578	6th character (Low Byte)	
579	6th character (High Byte)	
580	7th character (Low Byte)	
581	7th character (High Byte)	
582	8th character (Low Byte)	
583	8th character (High Byte)	
584	9th character (Low Byte)	
585	9th character (High Byte)	
586	10th character (Low Byte)	
587	10th character (High Byte)	

Byte	Contents	Example/Comment
588	11th character (Low Byte)	
589	11th character (High Byte)	
590	12th character (Low Byte)	
591	12th character (High Byte)	
592	13th character (Low Byte)	
593	13th character (High Byte)	
594	14th character (Low Byte)	
595	14th character (High Byte)	
596	15th character (Low Byte)	
597	15th character (High Byte)	
598	16th character (Low Byte)	
599	16th character (High Byte)	
600	Day	Date on which the program on the chip card was last modified Date modified : 28.11.2003 Byte 600: 1C, Byte 601: 0B, Byte 602: 07, Byte 603: D3 Time: 14 hours 25 minutes Byte 604: 0E, Byte 605: 19 Time zone 1: Byte 606: 01
601	Month	
602	Year (High Byte)	
603	Year (Low Byte)	
604	Hour	
605	Minute	
606	Time zone	
607	Configuration, 1st expansion module, left	Byte 607 ... 612 contains the Hex code for the expansion modules to the left of the base unit. Any fieldbus module in these Sub-Indices will not be considered (see Byte 555). PNOZ ml1p: A8 PNOZ ml2p: C8 PNOZ ma1p: B8
608	Configuration, 2nd expansion module, left	
609	Configuration, 3rd expansion module, left	
610	Configuration, 4th expansion module, left	
611	Configuration, 5th expansion module, left	
612	Configuration, 6th expansion module left	
613 ... 639	Reserved	

4.5.3.5 Element types

These bytes contain the element types

Byte	Content	Example/explanation
640	Element type. Element ID = 1	Element with ID = 1: Single-pole semiconductor output with feedback loop Byte 640: 51 hex See list containing the Element types [173] in the Appendix
...	...	
739	Element type. Element ID = 100	
740 ... 2047	Reserved	

4.5.3.6 Input data

These bytes contain the input data

Byte	Content	Example/explanation
2048	Inputs Bit 0 … 7	See chapter entitled "Basics", under Input data (to the PNOZmulti) [13]
2049	Inputs Bit 8 … 15	
2050	Inputs Bit 16 … 23	
2051	Reserved	
2052	Table number	
2053	Segment number	
2054	Byte 0	
2055	Byte 1	
2056	Byte 2	
2057	Byte 3	
2058	Byte 4	
2059	Byte 5	
2060	Byte 6	
2061	Byte 7	
2062	Byte 8	
2063	Byte 9	
2064	Byte 10	
2065	Byte 11	
2066	Byte 12	
2067 ... 2111	Reserved	

4.5.3.7 Diagnostic data

These bytes contain the diagnostic data

Byte	Diag_Bit	Content
2112	000	RUN, base unit is in RUN condition
	001	STOP, base unit is in STOP condition
	002	Base unit was stopped by the Configurator
	003	Start failed. External cause
	004	External fault
	005	Internal fault
	006	External fault at the inputs
	007	Internal fault at the inputs
2113	008	External fault at the outputs
	009	Internal fault at the outputs
	010	Fault on 1st expansion module, left
	011	Fault on 2nd expansion module, left
	012	Fault on 3rd expansion module, left
	013	Fault on 4th expansion module, left
	014	Fault on 5th expansion module, left
	015	Fault on 6th expansion module, left
2114	016	Fault on the base unit
	017	Fault on the 1st expansion module, right
	018	Fault on the 2nd expansion module, right
	019	Fault on the 3rd expansion module, right
	020	Fault on the 4th expansion module, right
	021	Fault on the 5th expansion module, right
	022	Fault on the 6th expansion module, right
	023	Fault on the 7th expansion module, right
2115	024	Fault on the 8th expansion module, right
	025	Fault on the link module
	026	Fault on the analogue input module
	027	Reserved
	028	Reserved
	029	Reserved
	030	Reserved
	031	Internal fault on the left-hand expansion module

Byte	Diag_Bit	Content
2116	032	Configuration error
	033	Fault in the application program
	034	Periphery error
	035	Fault on the speed monitor
	036	Fault on the bus module
	037	Internal self test error
	038	Internal data error
	039	Internal parameter error
2117	040	Internal serial/I2C error
	041	Internal time error
	042	Internal processor error
	043	Internal compare error
	044	Internal sequence error
	045	Internal periphery error
	046	Internal bus module error
	047	Internal speed monitor error

Information

All error messages or status messages that come from the PNOZmulti can overwrite each other. In particular, a PNOZmulti error message can be overwritten by a PNOZmulti status or error message without having to delete the error explicitly (via S-0-0099).

4.5.4 Firmware/FPGA update

Update the firmware as follows:

1. Make sure that the PNOZ mc10p is in NRT status (no data traffic with the Master).
2. Copy the update file (*.kfu) into the web server's root directory via FTP or TFTP. The file contains the firmware and the FPGA image.
3. Carry out a restart (power-on reset).
4. The firmware is updated on start-up. This takes approx. 1 minute. Do not interrupt this process. Afterwards the PNOZ mc10p will restart. Please note that at this point, communication between the PNOZmulti base unit and thePNOZ mc10p is interrupted (DIAG-LED flashes).
5. Carry out another restart (power-on reset) to restore the connection between the PNOZmulti base unit and PNOZ mc10p.

4.5.5 Forcing the virtual input data

Requests may be placed for the 24 virtual input data and table data via a web server integrated within the PNOZ mc10p (enter IP address of the PNOZ mc10p in Internet Explorer) (Bits 24 to 128 can be read and written using Table 9 Segment 1). The 20 Bytes (I/O and table data from the PNOZmulti) are uploaded accordingly. The web server can only be used in NRT status and not in the sercos III communication phases.

4.5.6 Communication with the sercos III Master

The input/output data is transmitted synchronously. The slow data stored in the object buffer is read asynchronously.

4.5.6.1 Synchronous data exchange

Default configuration for synchronous data exchange:

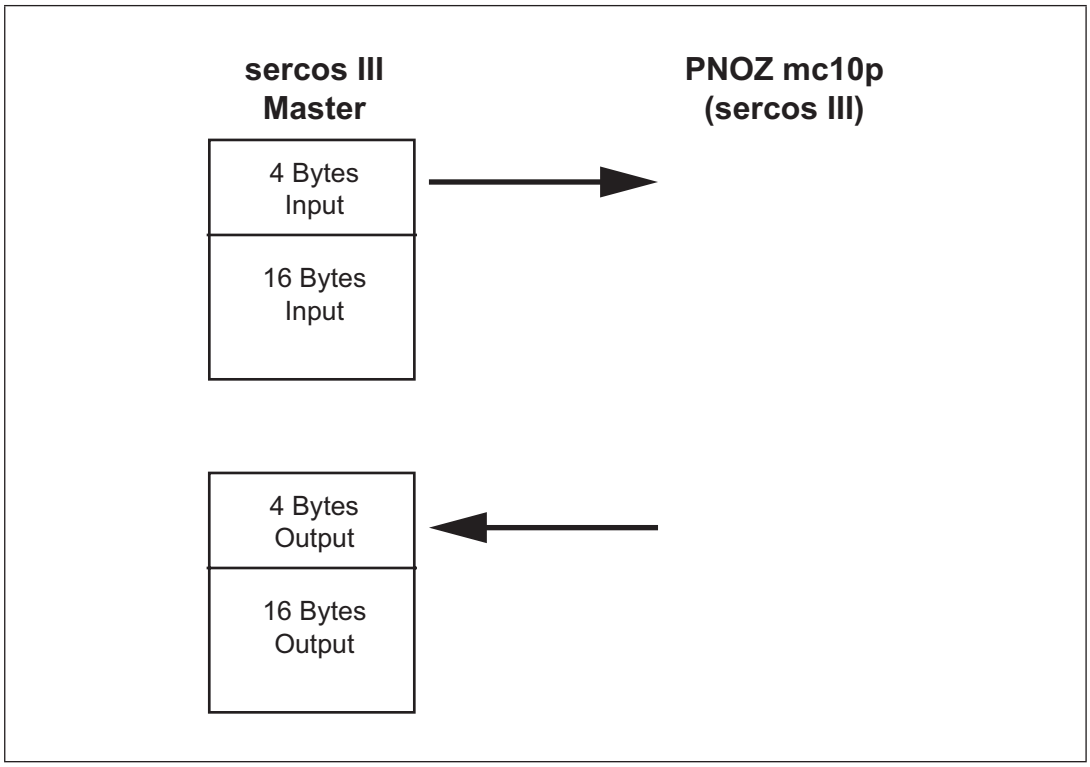


Fig.: Synchronous data exchange

To reduce data traffic, only the first 4 Bytes can be configured (see S-0-1507.0.2 in the chapter entitled Description of IDNs [📖 73]). Then only the 24Bit virtual I/Os / LED status are downloaded from the PNOZmulti.

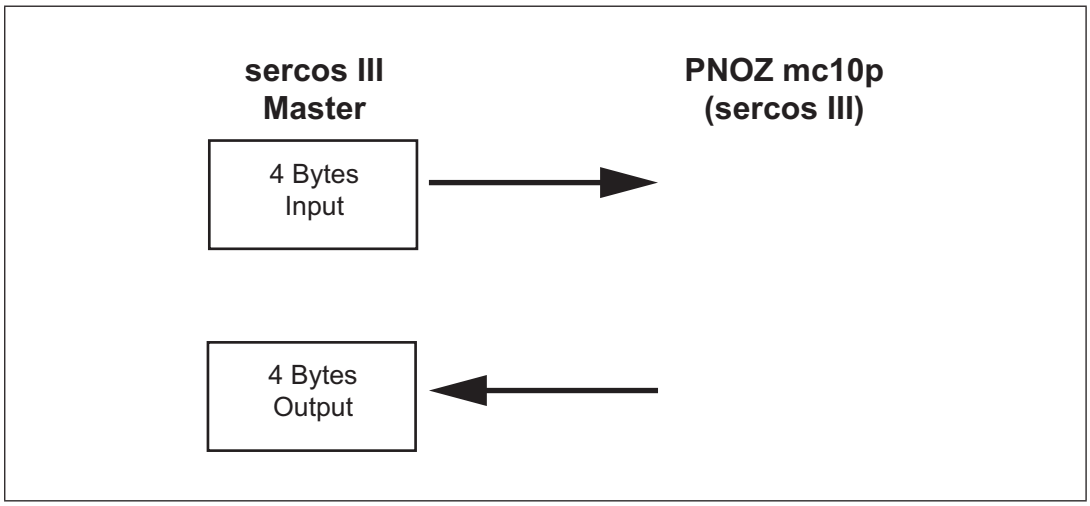


Fig.: Synchronous data exchange 4 Bytes

Both connections are always the same length (S-0-1050.x.5).

4.5.6.2 Asynchronous data access

Data stored in the object buffer can be polled asynchronously. 4 Bytes each can be addressed and polled. The address refers to the first of the 4 Bytes (see S-0-1507.0.19 and S-0-1507.0.20 in the chapter entitled Description of IDNs [📖 73]).

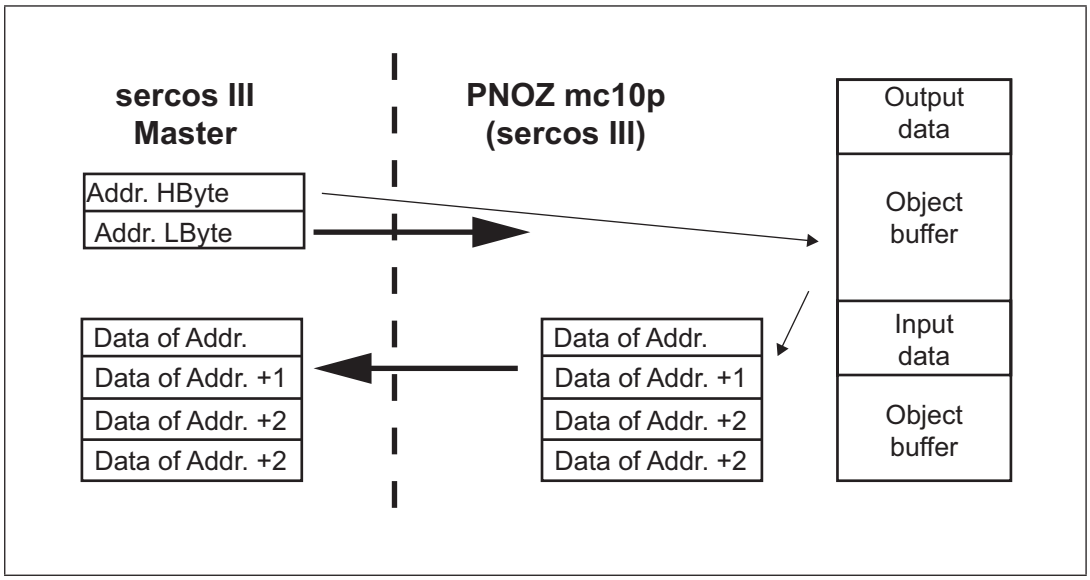


Fig.: Asynchronous data exchange

Data in the input block can also be written from the asynchronous section (see S-0-1507.0.20 in the chapter entitled Description of IDNs [📖 73]).

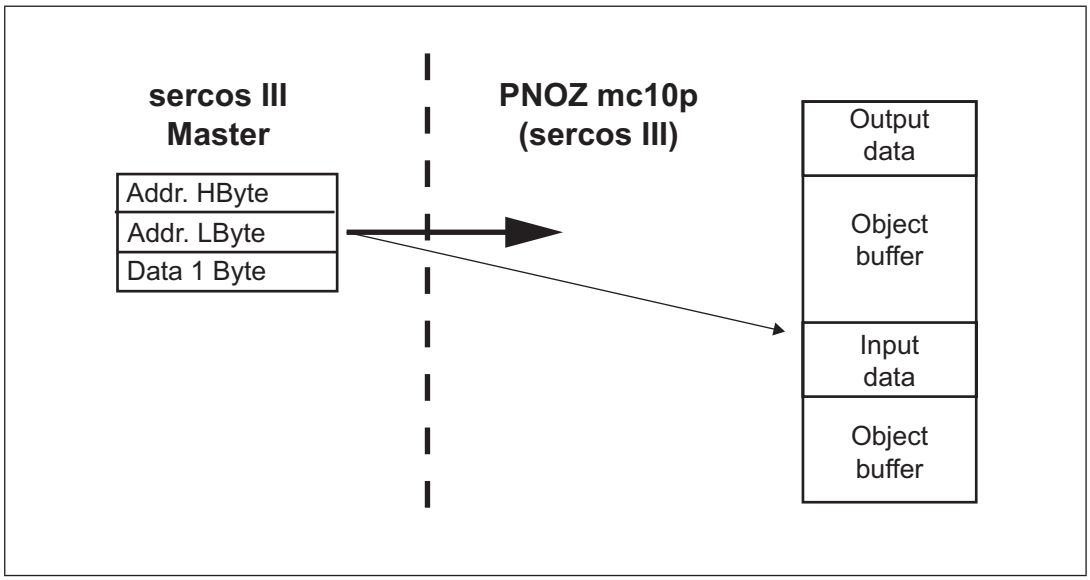


Fig.: Asynchronous data exchange

4.5.7 Sercos Master interface

4.5.7.1 Supported profiles

The fieldbus module PNOZ mc10p is designed as a sercos III IO device in accordance with the sercos specification 1.1.2. The following profiles are supported:

- GDP_Basic
  - S3 LED (in accordance with spec 1.1.3)
- SCP_FixCFG
  - Two master/slave connections, one as consumer and one as producer
  - Two different configurations for connections (with/without table data).
- FSP_IO
  - Compact IO device
  - S-0-1500 IO Bus Coupler
  - S-0-1502 Digital Output

    S-0-1502.0.5 PDOUT: 4 Bytes input/output data
  - S-0-1503 Digital Input

    S-0-1503.0.9 PDIN: 4 Bytes input/output data

    S-0-1503.0.19 Parameter Channel Receive: 6 Bytes diagnostic data from PNOZ-multi
  - S-0-1507 Complex Protocol

    S-0-1507.0.5 PDOUT 16 Bytes table data

    S-0-1507.0.9 PDIN 16 Bytes table data

    S-0-1507.0.19 Parameter Channel Receive: 4 Bytes received from object buffer

    S-0-1507.0.20 Parameter Channel Transmit: Send 2 / 3 Bytes to object buffer.

4.5.7.2 Default settings

- IP address: 192.168.1.64
- Subnet mask: 255.255.255.0
- Gateway address: 0.0.0.0
- Device name: PR100011
- Sercos address: 64

4.5.7.3 Description of IDNs

- **S-0-0128 CP4 Transition Check**

  If there is no communication between the PNOZ mc10p and base unit, IDN S-0-0128 will register an error after 30 seconds. It will not be possible to switch to communication phase 4 (CP4).

- **S-0-1502.0.5 Digital Output PDOUT**

  Contains the first 4 Bytes of input data. Is always configured in a consumer connection.

- **S-0-1503.0.9 Digital Input PDIN**

  Contains the first 4 Bytes of output data. Is always configured in a producer connection.

- **S-0-1503.0.19 Digital Input Parameter Channel Receive**

  Contains 6 Bytes of diagnostic data. Cannot be configured in connections.

- **S-0-1507.0.5 Complex PDOUT**

  Contains 16 Bytes of table data. Is always configured in a consumer connection with S-0-1507.0.2.

- **S-0-1507.0.9 Complex PDIN**

  Contains 16 Bytes of table data. Is always configured in a producer connection with S-0-1507.0.2.

- **S-0-1507.0.2 Configuration of Function Group Complex Protocol**

  Configures whether table data is included in both connections. To insert the table data in both connections, write in CP2 0x0018 before the master reads the connection length via S-0-1550.0.5 (default confguration). To remove the table data from both connections, write in CP2 0x001B. Every other value is ignored, but appears in the service channel (SVC) with error 0x7008.

- **S-0-1507.0.19 Complex Parameter** Channel Receive for Object Buffer

  Reads 4 Bytes from the object buffer (second step of read access). The address must be set in advance with S-0-1507.0.20 (see chapter entitled Asynchronous data access [📖 71]).

- **S-0-1507.0.20 Complex Parameter Channel Transmit for Object Buffer**

  Writes 2 Bytes (first step of object read access) or 3 Bytes (full object write access). If 2 Bytes are written, the address of the object buffer for subsequent read access is set with S-0-1507.0.19. If 3 Bytes are written, the third byte contains the value that is written in the addressed byte of the object buffer (see chapter entitled Asynchronous data access [📖 71]).

4.5.7.4 Communication channels with the PNOZmulti

This chapter describes the communication channels between the sercos III Master and the PNOZmulti, based on the sercos III communication phases (CP) and the selected connection configuration for input/output data and table data.

- **NRT**

  The PNOZmulti input data is set to "0" in NRT status. Communication is only possible via a web interface.

- **Communication phase 0 and 1 (CP0, CP1)**

  The PNOZmulti input data is set to "0" in communication phase 0. No communication is possible.

- **Communication phase 2 and 3 (CP2, CP3)**

  Communication is only possible via the sercos service channel (SVC). Four Bytes of input/output data can be written via IDN S-0-1502.0.5 (Digital PDOUT) and read via the command S-0-1503.0.9 (Digital PDIN). 16 Bytes of table data can be written via the command S-0-1507.0.5 (Complex PDOUT) and read via the command S-0-1507.0.9 (Complex PDIN).

  The full object buffer can be accessed via the service channel SVC Complex transmit/receive (S-0-1507.0.19 and S-0-1507.0.20).

- **Communication phase 4 (CP4)**

  Communication is possible via the sercos service channel (SVC) and also via the real-time channel (RT). It is possible to transmit input/output data only or additional table data via the real-time channel (RT), depending on the configuration. Please note that data disturbances may occur if the service channel and real-time channel are used at the same time.

4.5.7.5 Diagnostics

The IDNs S-0-0095 (diagnostic message) and S-0-039 (diagnostic number) are supported and are always set simultaneously by the PNOZmulti. The diagnostic classes are prioritised in accordance with the sercos specification.

- **sercos diagnostic numbers**

  Various pre-defined diagnostic numbers are used (see sercos specification)

- **PNOZ diagnostic numbers**

  The 48 PNOZ error and status messages (ref.: Chapter 4.6.3.7) from S-0-1503.0.19 are also shown in S-0-0095 and S-0-0390 as manufacturer-specified diagnostics in operational or error class.

  Operational: 0x010A0000 to 0x010A002F

  Error: 0x010F0000 to 0x010F002F

5 RS232/Ethernet interfaces

5.1 Overview

The RS232/Ethernet interfaces on the configurable control system PNOZmulti are used to

- Download the project
- Read the diagnostic data
- Set virtual inputs for standard functions
- Read virtual outputs for standard functions.

The interfaces are integrated into the base units PNOZmulti. The base units PNOZmulti Mini do not have an integrated interface; a communication module with interface can be connected in this case.

Either a serial interface RS232 or Ethernet interface is integrated, depending on the base unit type or communication module.

- **Serial interface RS232**
  - Base units PNOZ mXp
  - Base units PNOZ mmXp + PNOZ mmc2p
- **2 Ethernet interfaces**
  - Base units PNOZ mXp ETH
  - Base units PNOZ mmXp + PNOZ mmc1p

5.2 System requirements

Communication via the integrated interface (protocol, requirements), as described in this document, is supported from the following versions of the base units.

- Base unit PNOZ m0p: From Version 3.1
- Base unit PNOZ m1p: From Version 6.1
- Base unit PNOZ m1p ETH: From Version 2.1
- Base unit PNOZ m2p: From Version 3.1
- Base unit PNOZ m3p: From Version 2.1

Base units that are not listed support the described communication via the integrated interface from Version 1.0.

Please contact Pilz if you have an older version.

5.3 Interface description

5.3.1 Ethernet interfaces

The connection is established via two RJ45 sockets.

The Ethernet interface is configured in the PNOZmulti Configurator (for description see online help for the PNOZmulti Configurator).

All base units that have an Ethernet interface support Modbus/TCP (see chapter entitled Modbus/TCP [93]).

A PNOZmulti base unit can manage up to 8 Modbus/TCP connections and up to 4 PG port (Port 9000) connections.

Transmission rate:

- 10 MBit/s (10BaseT)

  or

- 100 MBit/s (100BaseTX)

5.3.1.1 RJ45 interfaces ("Ethernet")

Two free switch ports are provided as Ethernet interfaces via an internal autosensing switch. The autosensing switch automatically detects whether data transfer is occurring at 10 Mbit/s or 100 Mbit/s.

Information

The connected subscribers must support the autosensing/autonegotiation function. If not, the communication partner must be set permanently to "10 Mbit/s, half duplex".

The switch's automatic crossover function means there is no need to distinguish on the connection cable between patch cable (uncrossed data line connection) and crossover cable (crossover data line connection). The switch automatically creates the correct data line connection internally. Patch cable can therefore be used as the connection cable for end devices as well as cascading.

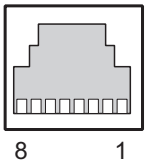
Both Ethernet interfaces use RJ45 technology.

5.3.1.2 Requirements of the connection cable and connector

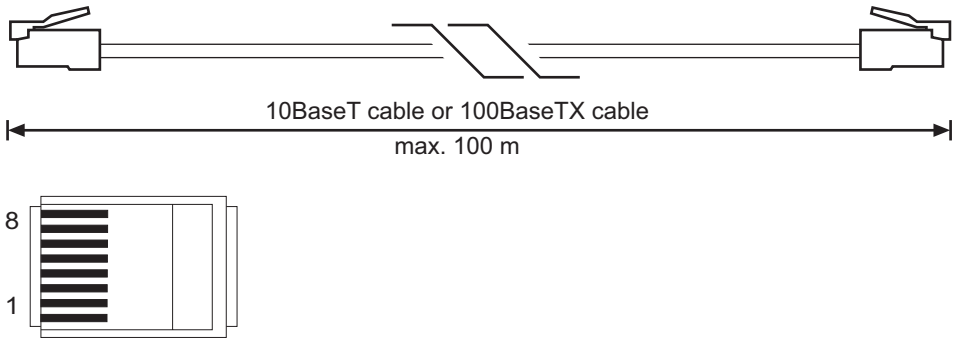
The following minimum requirements must be met:

- Ethernet standards (min. Category 5) 10BaseT or 100BaseTX
- Double-shielded twisted pair cable for industrial Ethernet use
- Shielded RJ45 connectors (industrial connectors)

5.3.1.3 Interface configuration

RJ45 socket 8-pin	PIN	Standard	Crossover
	1	TD+ (Transmit+)	RD+ (Receive+)
	2	TD- (Transmit-)	RD- (Receive-)
	3	RD+ (Receive+)	TD+ (Transmit+)
	4	n.c.	n.c.
	5	n.c.	n.c.
	6	RD- (Receive-)	TD- (Transmit-)
	7	n.c.	n.c.
	8	n.c.	n.c.

5.3.1.4 RJ45 connection cable

RJ45 connector 8-pin


CAUTION!

With the plug-in connection please note that the data cable and connector have a limited mechanical load capacity. Appropriate design measures should be used to ensure that the plug-in connection is insensitive to increased mechanical stress (e.g. through shock, vibration). Such measures include fixed routing with strain relief, for example.

5.3.1.5 Process data exchange

The RJ45 interfaces on the internal autosensing switch enable process data to be exchanged with other Ethernet subscribers within a network.

The PNOZ m ES ETH can also be connected to Ethernet via a hub (hub or switch).

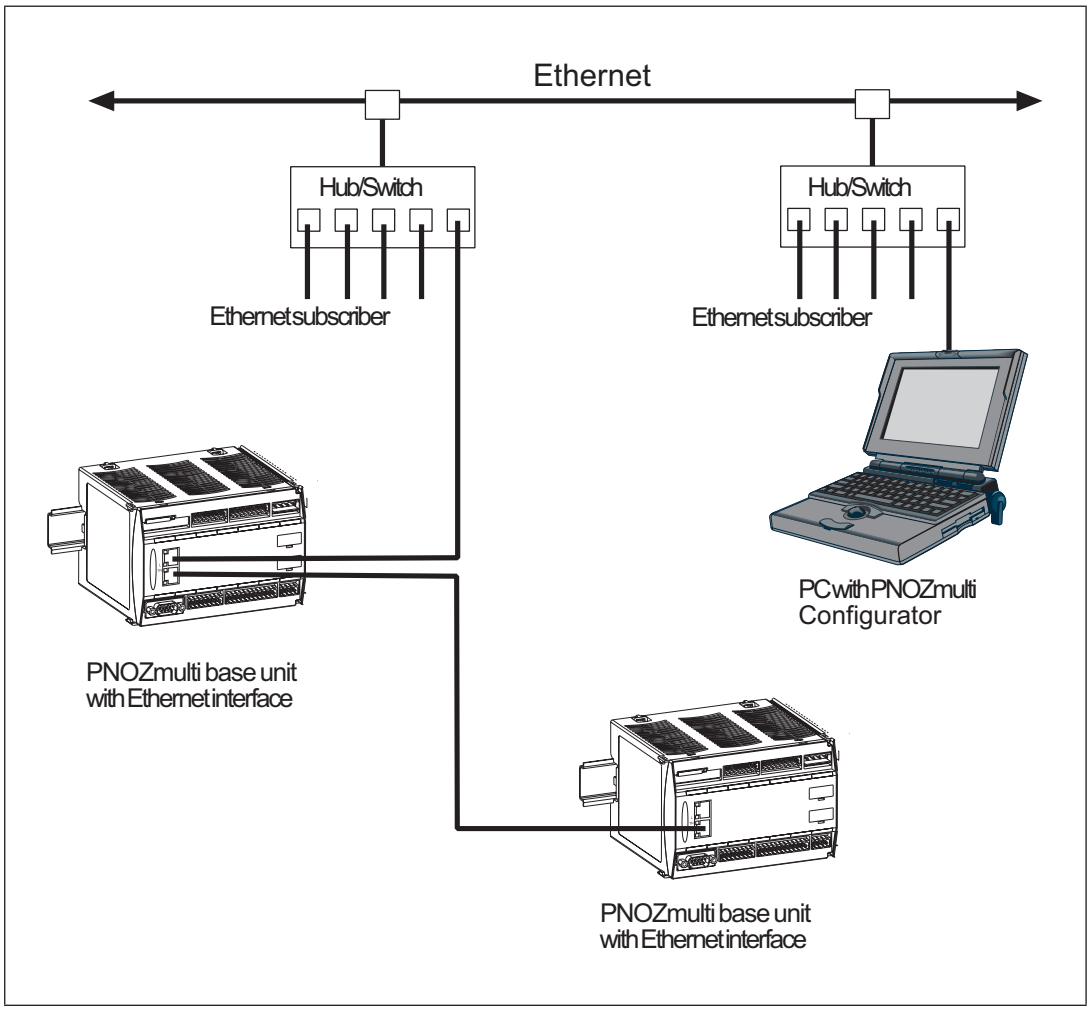


Fig.: PNOZmulti as Ethernet subscriber - possible topologies

5.3.2 Serial interface RS232

The connection between the RS 232 interface on the communications partner and the integrated interface on the base unit is established via a null modem cable.

Transmission rate:
19.2 kBit with

- 8 bits of data
- 1 start bit
- 2 stop bits
- 1 parity bit
- Even parity

5.4 Communication procedure

When communicating via the integrated interface, the PNOZmulti is always the connection's Server; the communications partner (PC, SPS) is the Client.

Information

For communication via Ethernet, the Ethernet interface must be set up in the PNOZmulti Configurator. The procedure is described in the PNOZmulti Configurator's online help.

Each communication is started by sending a request to the PNOZmulti. Requests are used to receive data from the PNOZmulti or send data to the PNOZmulti :

1. **Request**

   The user sends a request to the PNOZmulti via the communications partner.

2. **Response**

   The PNOZmulti sends a response to the communications partner after approx. 20 to 30 ms, confirming that the request has been received without error. Data is sent in accordance with the request.

5.5 Telegram structure

The telegram used for communication has the following structure:

Byte	Request		Byte	Response
0	0x05		0	0x05
1	0x15		1	0x15
2	0x00		2	0x00
3	Amount of usable data +5		3	Amount of usable data +5
4	Request No.		4	Confirmation/error
5	Segment No. HB		5	Segment No. HB
6	Segment No. LB		6	Segment No. LB
7	0x00		7	Reserved
8	Usable data Byte 0		8	Usable data Byte 0
9	Usable data Byte 1		9	Usable data Byte 1
10	Usable data Byte 2		10	Usable data Byte 2
...	...		...	...
Amount of usable data +7	Usable data Byte n		Amount of usable data +7	Usable data Byte n
Amount of usable data +8	BBC		Amount of usable data +8	BBC
Amount of usable data +9	0x10		Amount of usable data +9	0x10

5.5.1 Header

Byte 0 ... Byte 7 form the data block's header

- Byte 0: Always 0x05
- Byte 1: Always 0x15
- Byte 2: Always 0x00
- Byte 3: Payload amount plus 5
- Byte 4
  - Request: Request number
    A request is defined via the request number
    Requests
  - Response: Confirmation of request
    The request is confirmed: Request number + 0x80 (Bit 7 set).
    If the request cannot be processed, an error message is returned Troubleshooting [📖 91].
- Byte 5: High Byte of segment number
- Byte 6: Low Byte of segment number
- Byte 7
  - Request: Always 0x00
  - Response: Reserved

5.5.2 Usable data

Byte 8 ... Byte "Amount of usable data + 7" contain the requested usable data. The content and amount of usable data depend on the request. 0 - 40 Bytes of usable data can be transmitted. If no usable data is available, the BCC (Block Control Check) will follow directly after Byte 7.

- Bytes 8 ... "Amount of usable data + 7" (request):

  Application data, which is sent to the PNOZmulti
- Bytes 8 ... "Amount of usable data + 7" (response):

  Application data, which is sent from the PNOZmulti

5.5.3 Information data

The Bytes "Amount of usable data + 8 and + 9" contain the information data

- Byte "Amount of usable data + 8": Check sum (Block Control Check = BCC)

  The check sum is calculated as follows:

  BCC = 0 – (Byte 4 + ... Byte "Amount of usable data + 7")
- Byte "Amount of usable data + 9": Last Byte in each telegram

5.6 Usable data

This section describes the payload that can be transmitted as the result of a corresponding request.

5.6.1 Virtual inputs (Input Byte 0 ... Input Byte 15)

The virtual inputs are defined by the communications partner and transferred to the PNOZmulti. Each input has a number, e.g. input bit 4 of input byte 1 has the number i12.

Input Byte								
0	i7	i6	i5	i4	i3	i2	i1	i0
1	i15	i14	i13	i12	i11	i10	i9	i8
2	i23	i22	i21	i20	i19	i18	i17	i16
...	...	...	...	...	...	...	...	...

5.6.1.1 Mask (Mask Byte 0 ... Mask Byte 15)

The mask is used to determine which of the virtual inputs sent in a Byte are to be set. For example, if only inputs i0 to i5 are to be set in Byte 8, 0x3F must be entered in the mask in Byte 24

Send virtual inputs to the PNOZmulti [📖 83].

5.6.1.2 Watchdog

The watchdog is used to monitor the virtual inputs.

If no virtual inputs are sent by a communications partner within a defined watchdog time (watchdog timeout), the PNOZmulti sets the virtual inputs to "0".

The configuration and functionality of the watchdog varies and so is described in the respective requests.

5.6.2 Virtual outputs (Output Byte 0 ... Output Byte 15)

The virtual outputs are defined in the PNOZmulti Configurator. Each output that is used is given a number there, e.g. o0, o5 .... The state of output o0 is stored in bit 0 of output byte 0; the state of output o5 is stored in bit 5 of output byte 0 etc.

Output Byte								
0	o7	o6	o5	o4	o3	o2	o1	o0
1	o15	o14	o13	o12	o11	o10	o9	o8
2	o23	o22	o21	o20	o19	o18	o17	o16
...	...	...	...	...	...	...	...	...

5.6.3 LED status

The status of the LEDs is stored in a byte as follows:

- Bit 0 = 1: LED OFAULT is lit or flashes
- Bit 1 = 1: LED IFAULT is lit or flashes
- Bit 2 = 1: LED FAULT is lit or flashes
- Bit 3 = 1: LED DIAG is lit or flashes
- Bit 4 = 1: LED RUN is lit
- Bit 5-7: Reserved

5.6.4 Tables

Further information can be requested in table form.

A table consists of one or more segments. Each segment is made up of 13 Bytes.

The communications partner requests the required data using the table number and segment number. The PNOZmulti repeats the two numbers and sends the requested data.

There are a total of 10 tables, with the following contents:

Table 1:	Configuration
Table 2:	Reserved
Table 3:	State of inputs
Table 4:	State of outputs
Table 5:	Status of LED
Table 6:	Reserved
Table 7:	Diagnostic word
Table 8:	Element types
Table 9:	Transfer/state of the expanded virtual inputs and outputs
Table 10	State of the virtual inputs and outputs on the integrated link interface on the PNOZ mm0.2p
Table 11	State of the safe inputs and outputs on the safe Ethernet connection
Element types	The element type's byte is entered in Table 8

The content of the tables is described in detail in the Appendix.

5.7 Requests

A request is defined via the request number and segment number.

The following requests are available:

Request No.	Segment No.	Meaning
0x14	0x01	Send virtual inputs to the PNOZmulti
0x14	0x02	Send virtual inputs to the PNOZmulti, request state of the virtual outputs and LED status from the PNOZmulti
0x2C	0x02	Request state of virtual inputs and outputs from PNOZmulti
0x2F		Send data from the PNOZmulti in table form
0x53		Request all input and output data from PNOZmulti

5.7.1 Send virtual inputs to the PNOZmulti

Request 0x14 Segment 0x01

The communication partner uses this request to send virtual inputs to the PNOZmulti .

The mask (Bytes 24 to 39) is used to determine which of the virtual inputs in a Byte are to be set.

Telegram

Byte	Request
0	0x05
1	0x15
2	0x00
3	0x25
4	0x14
5	0x00
6	0x01
7	0x00
8	Virtual inputs Input Byte 0: i7 to i0
...	...
23	Virtual inputs Input Byte 15: i127 to i120
24	Mask Mask Byte 0: i7 to i0
...	...
39	Mask Mask Byte 15: i127 to i120
40	BCC
41	0x10

Byte	Response
0	0x05
1	0x15
2	0x00
3	0x05
4	0x94
5	0x00
6	0x01
7	0x00
8	0x6B
9	0x10

Information

If a fieldbus module is configured, virtual inputs cannot be activated via the integrated interface. In this case the request will be rejected by the PNOZmulti with error message 0x63 (request cannot be executed).

5.7.2 Send virtual inputs to the PNOZmulti, request state of the virtual outputs and LED status from the PNOZmulti

Request 0x14 Segment 0x02

The communication partner uses this request to send virtual inputs to the PNOZmulti in exactly the same way as request 0x14 segment 0x01. It also requests the virtual outputs and the LED status from the PNOZmulti .

The mask (Bytes 24 to 39) is used to determine which of the virtual inputs in a Byte are to be set. For example, if only inputs i0 to i5 are to be set in Byte 8, 0x3F must be entered in the mask in Byte 24.

Telegram

Byte	Request		Byte	Response
0	0x05		0	0x05
1	0x15		1	0x15
2	0x00		2	0x00
3	0x26		3	0x16
4	0x14		4	0x94
5	0x00		5	0x00
6	0x02		6	0x02
7	0x00		7	0x00
8	Virtual inputs Input Byte 0: i7 to i0		8	Virtual outputs Output Byte 0: o7 to o0
...	...		...	...
23	Virtual inputs Input Byte 15: i127 to i120		23	Virtual outputs Output Byte 15: o127 to o120
24	Mask Mask Byte 0: i7 to i0		24	LED status
...	...		25	BCC
39	Mask Mask Byte 15: i127 to i120		26	0x10
40	Control Byte			
41	BCC			
42	0x10			

To LED status LED status [82].

Information

If a fieldbus module is configured, virtual inputs cannot be activated via the integrated interface. In this case the request will be rejected by the PNOZmulti with error message 0x63 (request cannot be executed).

5.7.2.1 Control Byte (Byte 40)

Bits 0 … 2 of the Control Byte contain a watchdog function.

If no virtual inputs are sent by a communications partner within the defined watchdog time (watchdog timeout), the PNOZmulti sets the virtual inputs to "0".

Control Byte 40:

Reserved	Delayed Response	Error Message	Reserved	Reserved	W-Timer Bit2	W-Timer Bit1	W-Timer Bit0

- Bit 0 - 2: Watchdog Timeout

Watchdog Timer Bit 2	Watchdog Timer Bit 1	Watchdog Timer Bit 0	Watchdog Timeout
0	0	0	Timer deactivated
0	0	1	100 ms
0	1	0	200 ms
0	1	1	500 ms
1	0	0	1 s
1	0	1	3 s
1	1	0	5 s
1	1	1	10 s

- Bit 3 and 4: Reserved
- Bit 5 Error Message: Error message
  If the bit is set to "1", an error stack entry is generated when the watchdog is triggered.
- Bit 6 Delayed Response: Delayed response
  If the bit is set to "1", the response (send virtual outputs) is sent after a delay of one cycle.
- Bit 7: Reserved

Information

The watchdog functions of the requests 0x14 segment 0x02 and 0x53 use the same watchdog timer. In other words, the watchdog timer is reset if either request is called.

Information

To test if the watchdog is active, set a virtual input continuously to "1".
If this input becomes "0" after the set watchdog timeout has elapsed, then the watchdog is active.

5.7.3 Request state of virtual inputs and outputs from PNOZmulti

Request 0x2C Segment 0x02

The communication partner uses this request to request the state of the virtual inputs and outputs from the PNOZmulti.

Telegram

Byte	Request		Byte	Response
0	0x05		0	0x05
1	0x15		1	0x15
2	0x00		2	0x00
3	0x05		3	0x26
4	0x2C		4	0xAC
5	0x00		5	0x00
6	0x02		6	0x02
7	0x00		7	0x00
8	0xD2		8	Virtual inputs Input Byte 0: i7 to i0
9	0x10		...	...
			23	Virtual inputs Input Byte 15: i127 to i120
			24	Virtual outputs Output Byte 0: o7 to o0
			...	...
			39	Virtual outputs Output Byte 15: o127 to o120
			40	LED status
			41	BCC
			42	0x10

5.7.4 Send data from the PNOZmulti in table form

Request 0x2F

The communication partner uses this request to request data from the PNOZmulti in table form.

The content of the tables and segments is described in detail in the Appendix.

Telegram

Byte	Request
0	0x05
1	0x15
2	0x00
3	0x07
4	0x2F
5	0x00
6	0x00
7	0x00
8	Table No.
9	Segment No.
10	BCC
11	0x10

Byte	Response
0	0x05
1	0x15
2	0x00
3	0x14
4	0xAF
5	0x00
6	0x00
7	0x00
8	Table No.
9	Segment No.
10	Byte 0 of Table x, Segment y
...	...
22	Byte 12 of Table x, Segment y
23	BCC
24	0x10

- Byte 8: Table number

  Example: 0x15 for Table 21: Process data, right-hand expansion modules
- Byte 9: Segment number

  Example: 0x00 for Segment 0, in Byte 4 state of outputs o0 ... o7 of the right-hand expansion modules

Information

If the requested segment is unavailable, the Segment No. is set to 255.
Example:
Request: Table No. 20 Segment No. 45
Response: Table No. 20 Segment No. 255
Byte 10 ... 22 = 0

5.7.5 Send input and output data (cf. fieldbus communication)

Request 0x53

The communications partner uses this request to send input data to the PNOZmulti and request output data from the PNOZmulti (see section entitled "Fieldbus modules", under "Basics").

As with fieldbus communication, 20 Bytes each are reserved for input and output data (Byte 8 – 27); they are updated approx. every 15 ms.

Byte	Request		Byte	Response
0	0x05		0	0x05
1	0x15		1	0x15
2	0x00		2	0x00
3	0x19		3	0x19
4	0x53		4	0xD3
5	Control Byte		5	Control Byte
6	Reserved		6	Reserved
7	0x00		7	0x00
8	Input Byte 0		8	Output Byte 0
9	Input Byte 1		9	Output Byte 1
10	Input Byte 2		10	Output Byte 2
...	...		...	...
27	Input Byte 19		27	Output Byte 19
28	BCC		28	BCC
29	0x10		29	0x10

5.7.5.1 Input data (to the PNOZmulti)

Input Byte	Content
0	i7 to i0
1	i15 to i8
2	i23 to i16
3	Reserved
4	Table No.
5	Segment No.
6	Byte 0 of Table x, Segment y
7	Byte 1 of Table x, Segment y
8	.
9	.
10	.
11	.
12	.

Input Byte	Content
13	.
14	.
15	.
16	.
17	.
18	Byte 12 of Table x, Segment y
19	Reserved

The virtual inputs are set and a specific table/segment is requested in the input data.

Information

Bytes 6 to 18 are used only for Table 9, Segment 1.

Information

If a fieldbus module is configured, virtual inputs cannot be activated via the integrated interface. In this case the request will be rejected by the PNOZ-multi with error message 0x63 (request cannot be executed).

5.7.5.2 Output data (from the PNOZmulti)

Output Byte	Content
0	o7 to o0
1	o15 to o8
2	o23 to o16
3	LED status
4	Table No.
5	Segment No.
6	Byte 0 of Table x, Segment y
7	Byte 1 of Table x, Segment y
8	.
9	.
10	.
11	.
12	.
13	.
14	.
15	.

Output Byte	Content
16	.
17	.
18	Byte 12 of Table x, Segment y
19	Reserved

The states of the configured outputs and the LED are in Byte 0 ...Byte 3. The content of the tables and segments is described in detail in the section entitled "Usable data"/"Tables".

5.7.5.3 Control Byte (Byte 5)

Bits 0 … 2 of the Control Byte contain a watchdog function.

If no virtual inputs are sent by a communications partner within the defined watchdog time (watchdog timeout), the PNOZmulti sets the virtual inputs to "0".

Control Byte 5:

Read/ Write	Delayed Response	Error Message	Reserved	Reserved	W-Timer Bit 2	W-Timer Bit 1	W-Timer Bit 0

- Bit 0 - 2: Watchdog Timeout

Watchdog Timer Bit 2	Watchdog Timer Bit 1	Watchdog Timer Bit 0	Watchdog Timeout
0	0	0	Timer deactivated
0	0	1	100 ms
0	1	0	200 ms
0	1	1	500 ms
1	0	0	1 s
1	0	1	3 s
1	1	0	5 s
1	1	1	10 s

- Bit 3 and 4: Reserved
- Bit 5 Error Message: Error message
  If the bit is "1", an error stack entry is generated when the watchdog is triggered.
- Bit 6 Delayed Response: Delayed response
  If the bit is "1", the response (send virtual outputs) is sent after a delay of one cycle.
- Bit 7: Read/Write Read/write access

  If the bit is "1", write protection is active; data cannot be overwritten. With read access, the watchdog timer is not reset, Bit 6 Delayed Response is deactivated.

Information

The watchdog functions of the requests 0x14 segment 0x02 and 0x53 use the same watchdog timer. In other words, the watchdog timer is reset if either request is called.

Information

To test if the watchdog is active, set a virtual input continuously to "1".
If this input becomes "0" after the set watchdog timeout has elapsed, then the watchdog is active.

5.8 Troubleshooting

5.8.1 Request format does not meet specifications

If the request format does not meet the specifications, the PNOZmulti sends the following response:

Byte	Response
0	0x05
1	0x02
2	0x00
3	0x02
4	0x00
5	0x02
6	0x10

5.8.2 Error while executing a request

If an error occurs while executing a request, the PNOZmulti sends the following response:

Byte	Response
0	0x05
1	0x15
2	0x00
3	0x05
4	Error Code
5	0x00
6	0x00
7	0x00
8	BCC
9	0x10

Error Codes (Byte 4):

- 0x62: Request's BCC is incorrect
- 0x63: Request cannot be executed
- 0x64: Request unknown
- 0x67: Table or segment number unavailable
- 0x68: PNOZmulti not ready

6 Modbus/TCP

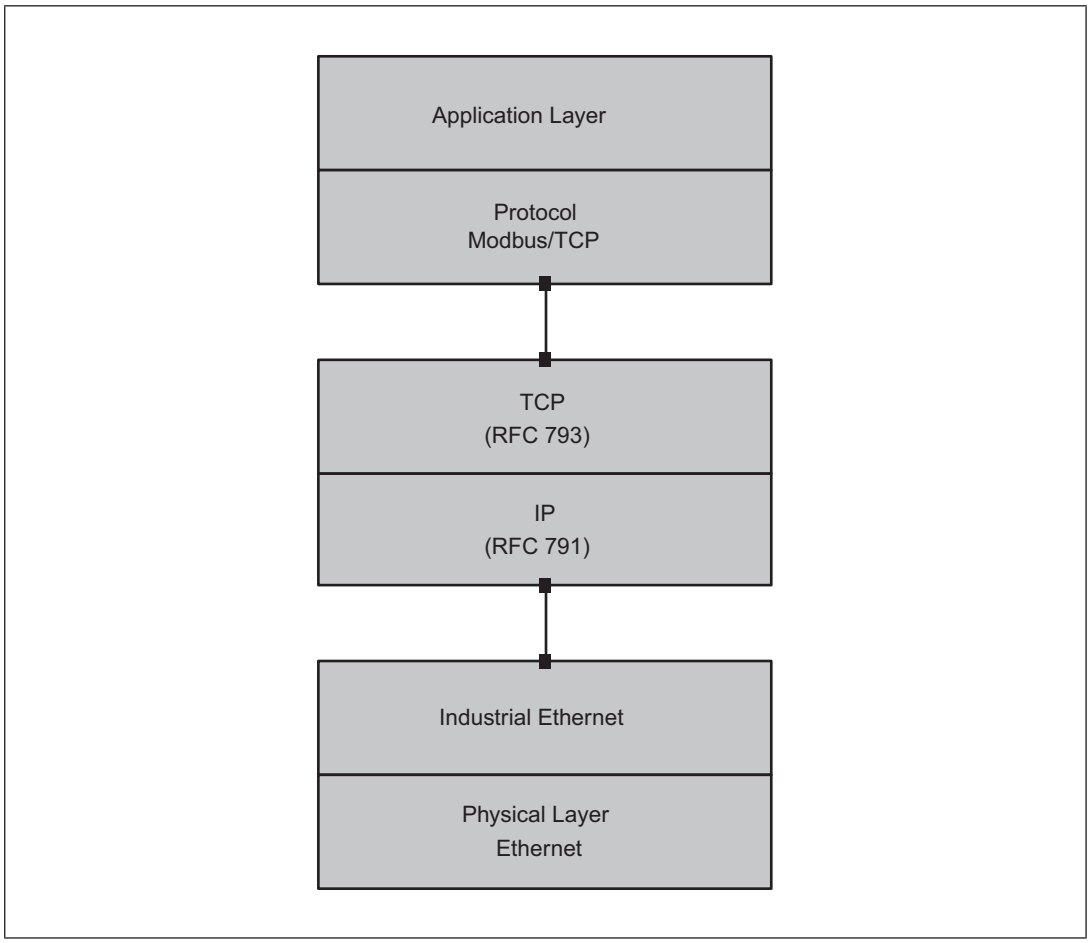
6.1 System requirements

- PNOZmulti Configurator: From Version 7.1.0
- All base units and modules that have an Ethernet interface (exception: PNOZ m1p ETH from V2.1)

Please contact Pilz if you have an older version.

6.2 Modbus/TCP - Basics

Modbus/TCP is an open fieldbus standard published by the User Group MODBUS-IDA (see www.Modbus-IDA.org).



Modbus/TCP is a protocol based on Industrial Ethernet (TCP/IP over Ethernet). It is one of the protocols with Client/Server communication. Data is transferred via a request/response mechanism using function codes (FC).

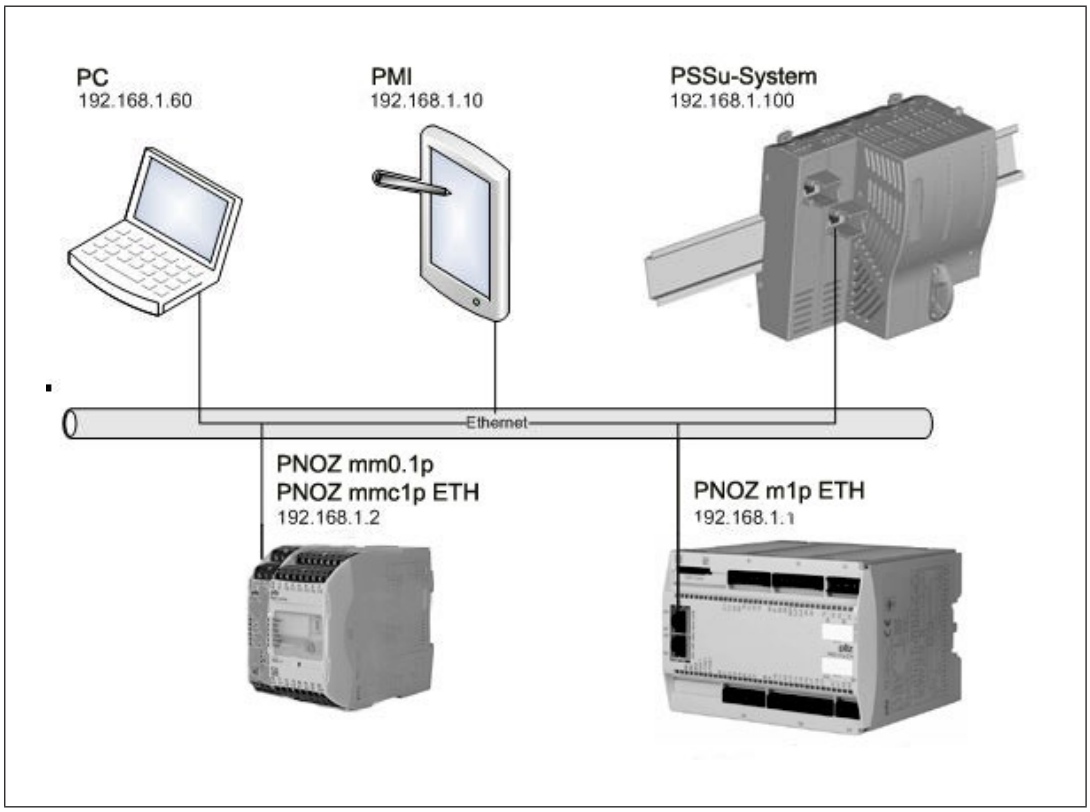
Modbus/TCP is connection-oriented, i.e. before usable data is transferred via Modbus/TCP, a connection must first be established between two Modbus/TCP interfaces. The initiator of the connection is called the Client. The communications partner with which the Client establishes the connection is called the Server. Whether a connection on a device assumes the role of Client or Server is defined when a connection is configured. As a result, the Server/Client role only applies for that specific connection.

6.3 Modbus/TCP with PNOZmulti

All base units from the configurable control system PNOZmulti that have an Ethernet interface (PNOZ m1p ETH from V2.1) support Modbus/TCP. The same applies for the base units PNOZmulti Mini in conjunction with a communication module with Ethernet interface.

A PNOZmulti base unit can manage a max. of 8 Modbus/TCP connections. The PNOZmulti is always the Server in a connection. The connection Clients may be various devices, e.g. PC (PNOZmulti Configurator), control system, display unit. They can access the configurable control system PNOZmulti simultaneously.

The virtual I/Os plus all the information that is polled during fieldbus communication are contained in data areas. The data is accessed directly. It's no longer necessary to switch via table/segment.



The configurations required for Modbus/TCP are fully pre-configured in the PNOZmulti operating system. All that's necessary in the PNOZmulti Configurator is to activate the virtual inputs and outputs (see PNOZmulti Configurator's online help, under "Display and edit module selection").

On a configurable control system PNOZmulti, port number "502" is fixed as the default for data exchange via a Modbus/TCP connection. It is not displayed in the PNOZmulti Configurator and cannot be modified.

6.4 Data areas

6.4.1 Overview

A configurable control system PNOZmulti supports the following Modbus/TCP data areas:

Data area	Modbus syntax	Example
Coils (Bit) 0x00000 … 0x65535 [read/write]	0x[xxxxx]	0x00031 (virtual input i31)
Discrete Inputs (Bit) 1x00000 … 1x65535 [read only]	1x[xxxxx]	1x08193 (virtual output o1)
Input Register (Word/16 Bits) 3x00000 … 3x65535 [read only]	3x[xxxxx]	3x00002 (virtual inputs 32 ... 47)
Holding Register (Wort/16 Bits) 4x00000 … 4x65535 [read/write]	4x[xxxxx]	4x00805 (project name, 1st character)

Information

Addressing for PNOZmulti systems starts at “0”. On devices from other manufacturers, addressing can start at “1”.
Please refer to the operating manual provided by the relevant manufacturer.

6.4.2 Function codes

The following function codes (FC) are available for communication with the PNOZmulti via Modbus/TCP:

Function code	Function	
FC 01	Read Coils	The connection Client reads bit data from the connection Server, data length ≥ 1 Bit, content: input/output data (data received from 0x)
FC 02	Read Discrete Input	The connection Client reads bit data from the connection Server, data length ≥ 1 Bit, content: input/output data (data received from 1x)
FC 03	Read Holding Register	The Client in a connection reads word data from the Server in the connection, data length ≥ 1 Word, content: diagnostic word (data received from 4x)

Function code	Function	
FC 04	Read Input Register	The Client in the connection reads word data from the Server in the connection, data length ≥ 1 Word, content: diagnostic word (data received from 3x)
FC 05	Write Single Coil	The connection Client writes to one bit datum in the connection Server, data length = 1 Bit, content: input data (send data to 0x)
FC 06	Write Single Register	The connection Client writes to one word datum in the connection Server, data length = 1 Word, content: input data (send data to 4x)
FC 15	Write Multiple Coils	The connection Client writes to multiple bit data in the connection Server, data length ≥ 1 Bit, content: input data (send data to 0x)
FC 16	Write Multiple Registers	The connection Client writes to multiple word data in the connection Server, data length = 1 Word, content: input data (send data to 4x)
FC 23	Read/Write Multiple Registers	The connection Client reads and writes multiple word data within a telegram (receive data from 3x and send data to 4x)

6.4.3 Data transfer limits

The following table contains information on the maximum data length per telegram that is supported:

Data transfer		Max. data length per telegram
Read data (Bit)	FC 01 (Read Coils)	1 … 2000
	FC 02 (Read Discrete Inputs)	
Read data (Bit)	FC 05 (Write Single Coil)	1 Bit
	FC 15 (Write Multiple Coils)	1 … 1968
Read data (Word)	FC 03 (Read Holding Registers)	1 … 125
	FC 04 (Read Input Register)	
Write data (Word)	FC 06 (Write Single Register)	1 Word
	FC 16 (Write Multiple Registers)	1 … 123 Words
Read and write data (Word)	FC 23 (Read/Write Multiple Registers)	Read 1 … 125 Words Write 1 … 121 Words

Information

There may be some restrictions in data length, depending on the device that is used. Please refer to the information stated in the operating manual of the device you are using.

6.4.4 Assignment of data areas

Data can be accessed via various Modbus/TCP data areas.

The tables below show the relationship between Modbus/TCP data areas and the content of the data areas.

6.4.4.1 Virtual inputs

The table below describes the Modbus/TCP data areas that contain the current state of the virtual inputs on the PNOZmulti. These are the virtual inputs that can be set by the user.

Relevant areas for the data are defined in each Modbus/TCP data area (Coils (0x), Discrete Inputs (1x), Input Register (3x), Holding Register (4x)). Read/write access will depend on the Modbus/TCP data area.

Register (3x, 4x)	Coil/ Descrete Input (0x, 1x)	Content	High Byte	Low Byte
0	15... 0	State of the inputs 0...15	i15…i8	i7…i0
1	31... 6	State of the inputs 16...31	i31…i24	i23…i16
2	47...32	State of the inputs 32...47	i47…i40	i39…i32
3	63...48	State of the inputs 48...63	i63…i56	i55…i48
4	79... 64	State of the inputs 64...79	i79…i72	i71…i64
5	95...80	State of the inputs 80...95	i95…i88	i87…i80
6	111...96	State of the inputs 96...111	i111…i104	i103…i96
7	127...112	State of the inputs 112...127	i127…i120	i119…i112

6.4.4.2 Control Register

A watchdog can be activated in Control Register 255.
If no input bits are set by a Modbus/TCP subscriber within the preset time, the PNOZmulti will set the input bits to "0".

The table below describes the Modbus/TCP data areas for the watchdog.

A relevant area for the watchdog is defined in each Modbus/TCP data area (Coils (0x), Discrete Inputs (1x), Input Register (3x), Holding Register (4x)). Read/write access will depend on the Modbus/TCP data area.

Register (3x, 4x)	Coil/ Descrete Input (0x, 1x)	Content	High Byte	Low Byte
255	4095...4080	Control Register	See table below	

High Byte	WD-Trigger	Error Message	Reserved	Reserved	Reserved	W-Timer Bit 2	W-Timer Bit 1	W-Timer Bit 0
Low Byte	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved

Bit 15 "Watchdog Trigger": The watchdog can be triggered by setting Bit 15 constantly to "1", or by a Client writing in the input area of the 128 inputs. The state of the bit when reading is undefined. A 1 or a 0 can be read.

Bit 14 "Error Message": If this bit is set, an error stack entry is generated when the watchdog is triggered.

Bits 10 … 8 "WD Timer": If the set time for the watchdog is set, then Bit 15 must be set, or become set, at the same time.

Watchdog Timer Bit 2	Watchdog Timer Bit 1	Watchdog Timer Bit 0	Watchdog Time
0	0	0	Timer deactivated
0	0	1	100 ms
0	1	0	200 ms
0	1	1	500 ms
1	0	0	1 s
1	0	1	3 s
1	1	0	5 s
1	1	1	10 s

Information

To check whether the watchdog has been triggered, set a virtual input permanently to "1".
If the corresponding input in the PNOZmulti is "0", then the watchdog has been triggered.

6.4.4.3 Virtual outputs

The table below describes the Modbus/TCP data areas that contain the state of the virtual outputs on the PNOZmulti.

Relevant areas for the data are defined in the Modbus/TCP data areas Discrete Inputs (1x) and Input Register (3x). Read access is available to these data areas.

Register (3x)	Descrete Input (1x)	Content	High Byte	Low Byte
512	8207...8192	State of the outputs 0...15	o15…o8	o7…o0
513	8223...8208	State of the outputs 16...31	o31…o24	o23…o16
514	8239...8224	State of the outputs 32...47	o47…o40	o39…o32
515	8255...8240	State of the outputs 48...63	o63…o56	o55…o48
516	8271...8256	State of the outputs 64...79	o79…o72	o71…o64
517	8287...8272	State of the outputs 80...95	o95…o88	o87…o80
518	8303...8288	State of the outputs 96...111	o111…o104	o103…o96
519	8319...8304	State of the outputs 112...127	o127…o120	o119…o112

6.4.4.4 LEDs

The table below describes the Modbus/TCP data areas that contain the LED statuses.

Relevant areas for the data are defined in the Modbus/TCP data areas Discrete Inputs (1x) and Input Register (3x). Read access is available to these data areas.

Register (3x)	Discrete Input (1x)	Content	High Byte	Low Byte
520	8335...8320	8 Bit LED status; 8 Bit reserved	Reserved	PNOZmulti LEDs
521…783		Reserved		

Bit 0 = 1: LED OFAULT illuminates or flashes
Bit 1 = 1: LED IFAULT illuminates or flashes
Bit 2 = 1: LED FAULT illuminates or flashes
Bit 3 = 1: LED DIAG illuminates or flashes
Bit 4 = 1: LED RUN illuminates
Bit 5: Reserved
Bit 6: Reserved
Bit 7: Reserved

6.4.4.5 Configuration

The table below describes the Modbus/TCP data areas that contain the device data from the base unit and the project data. The data was defined in the PNOZmulti Configurator.

Relevant areas for the data are defined in the Modbus/TCP data areas Discrete Inputs (1x) and Input Register (3x). Read access is available to these data areas.

Register (3x)	Discrete Input (1x)	Content	High Byte	Low Byte
784	12559... 12544	Product number	HH Byte	HL Byte
785	12575... 12560	Product number	LH Byte	LL Byte
786	12591... 12576	Device version	HH Byte	HL Byte
787	12607... 12592	Device version	LH Byte	LL Byte
788	12623... 12608	Serial number	HH Byte	HL Byte
789	12639... 12624	Serial number	LH Byte	LL Byte
790	12655... 12640	Reserved		
791	12671... 12656	Check sum safe	H Byte	L Byte
792	12687... 12672	Overall project check sum	H Byte	L Byte
793	12703... 12688	Date of project	Day	Month
794	12719... 12704	Date of project	Year (H Byte)	Year (L Byte)
795	12735... 12720	Operating hours	HL Byte	LH Byte
796	12751... 12736	Operating hours / Type of base unit	LL Byte	Type
797	12767... 12752	Reserved		
798	12783... 12768	Configuration, fieldbus modules / RS232 / expansion module, right	Slot 1	Fieldbus
799	12799... 12784	Configuration, expansion module, right	Slot 3	Slot 2
800	12815... 12800	Configuration, expansion module, right	Slot 5	Slot 4
801	12831... 12816	Configuration, expansion module, right	Slot 7	Slot 6
802	12847... 12832	Configuration, expansion module, right	Reserved	Slot 8
803	12863... 12848	Reserved		
804	12879... 12864	Reserved		
805	12895... 12880	Project name	1st character (H Byte)	1st character (L Byte)
806	12911... 12896	Project name	2nd character (H Byte)	2nd character (L Byte)
807	12927... 12912	Project name	3rd character (H Byte)	3rd character (L Byte)

Register (3x)	Discrete Input (1x)	Content	High Byte	Low Byte
808	12943... 12928	Project name	4th character (H Byte)	4th character (L Byte)
809	12959... 12944	Project name	5th character (H Byte)	5th character (L Byte)
810	12975... 12960	Project name	6th character (H Byte)	6th character (L Byte)
811	12991... 12976	Project name	7th character (H Byte)	7th character (L Byte)
812	13007... 12992	Project name	8th character (H Byte)	8th character (L Byte)
813	13023... 13008	Project name	9th character (H Byte)	9th character (L Byte)
814	13039... 13024	Project name	10th character (H Byte)	10th character (L Byte)
815	13055... 13040	Project name	11th character (H Byte)	11th character (L Byte)
816	13071... 13056	Project name	12th character (H Byte)	12th character (L Byte)
817	13087... 13072	Project name	13th character (H Byte)	13th character (L Byte)
818	13103... 13088	Project name	14th character (H Byte)	14th character (L Byte)
819	13119... 13104	Project name	15th character (H Byte)	15th character (L Byte)
820	13135... 13120	Project name	16th character (H Byte)	16th character (L Byte)
821	13151... 13136	Project name	0xFF	0xFF
822	13167... 13152	Reserved		
823	13183... 13168	Reserved		
824	13199... 13184	Reserved		
825	13215... 13200	Reserved		
826	13231... 13216	Project date	Day	Month
827	13247... 13232	Project date	Year (H Byte)	Year (L Byte)
828	13263... 13248	Project date	Hour	Minute
829	13279... 13264	Project date	Time zone	Reserved
830	13295... 13280	Reserved		
831	13311... 13296	Reserved		
832	13327... 13312	Reserved		
833	13343... 13328	Fieldbus type	Fieldbus type (H Byte)	Fieldbus type (L Byte)

Register (3x)	Discrete Input (1x)	Content	High Byte	Low Byte
834	13359... 13344	Fieldbus module software version	Version	Reserved
835	13375... 13360	Reserved		
836	13391... 13376	Reserved		
837	13407... 13392	Reserved		
838	13423... 13408	Reserved		
839	13439... 13424	Reserved		
840	13455... 13440	Configuration, expansion module left	Slot 2	Slot 1
841	13471... 13456	Configuration, expansion module left	Slot 4	Slot 3
842	13487... 13472	Configuration, expansion module left	Slot 6	Slot 5
843	13503... 13488	Reserved		
844	13519... 13504	Reserved		
845	13535... 13520	Reserved		
846	13551... 13536	Reserved		

6.4.4.6 State of the inputs from the base unit and expansion modules

The table below describes the Modbus/TCP data areas that contain the state of the inputs from the base unit and expansion modules.

Relevant areas for the data are defined in the Modbus/TCP data areas Discrete Inputs (1x) and Input Register (3x). Read access is available to these data areas.

Register (3x)	Coil/ Discrete Input (1x)	Content	High Byte	Low Byte
847	13567... 13552	Base unit I0 - I15 Base unit Mini IM0 ... I15	i15 ….i8	i7 ….i0
848	13583... 13568	Base unit I16 – I19 Base unit Mini I16 ... IM19	Reserved	i23...i16
849	13599... 13584	Reserved / Expansion module, right	Right 1 (i7... i0)	Reserved
850	13615... 13600	Expansion module, right	Right 3 (i7... i0)	Right 2 (i7...i0)
851	13631... 13616	Expansion module, right	Right 5 (i7... i0)	Right 4 (i7...i0)
852	13647... 13632	Expansion module, right	Right 7 (i7... i0)	Right 6 (i7...i0)

Register (3x)	Coil/ Discrete Input (1x)	Content	High Byte	Low Byte
853	13663... 13648	Expansion module, right / reserved	Reserved	Right 8 (i7...i0)
854	13679... 13664	Expansion module, left	Left 1 (i15… i8)	Left 1 (i7…i0)
855	13695... 13680	Expansion module, left	Left 1 (i31… i24)	Left 1 (i23… i16)
856	13711... 13696	Expansion module, left	Left 2 (i15… i8)	Left 2 (i7…i0)
857	13727... 13712	Expansion module, left	Left 2 (i31… i24)	Left 2 (i23… i16)
858	13743... 13728	Expansion module, left	Left 3 (i15… i8)	Left 3 (i7…i0)
859	13759... 13744	Expansion module, left	Left 3 (i31… i24)	Left 3 (i23… i16)
860	13775... 13760	Reserved		
861	13791... 13776	Expansion module, left	Left 4 (i15… i8)	Left 4 (i7…i0)
862	13807... 13792	Expansion module, left	Left 4 (i31… i24)	Left 4 (i23… i16)
863	13823... 13808	Expansion module, left	Left 5 (i15… i8)	Left 5 (i7…i0)
864	13839... 13824	Expansion module, left	Left 5 (i31… i24)	Left 5 (i23… i16)
865	13855... 13840	Expansion module	Left 6 (i15… i8)	Left 6 (i7…i0)
866	13871... 13856	Expansion module, left	Left 6 (i31… i24)	Left 6 (i23… i16)
867	13887... 13872	Reserved		

Register 854 to 866 "Expansion module, left"
Please note: for analogue input modules, the content for "High Byte" and "Low Byte" is transposed.

6.4.4.7 State of the outputs from the base unit and expansion modules

The table below describes the Modbus/TCP data areas that contain the state of the outputs from the base unit and expansion modules.

Relevant areas for the data are defined in the Modbus/TCP data areas Discrete Inputs (1x) and Input Register (3x). Read access is available to these data areas.

Register (3x)	Coil/ Discrete Input (1x)	Content	High Byte	Low Byte
868	13903... 13888	Base unit Mini IM0 … IM3	Reserved	4Bit reserved M3… M0
869	13919... 13904	Base unit Mini IM16 ... IM19, TM20 … TM23 Base unit o0 - o3	4 Bit reserved..o3… o0	M23 … M16
870	13935... 13920	Base unit o4 - o5 / expansion module, right	Right 1 o7… o0	6 Bit reserved o5,o4
871	13951... 13936	Expansion module, right	Right 3 o7… o0	Right 2 o7… o0
872	13967... 13952	Expansion module, right	Right 5 o7… o0	Right 4 o7… o0
873	13983... 13968	Expansion module, right	Right 7 o7… o0	Right 6 o7… o0
874	13999... 13984	Expansion module, right / res	Reserved	Right 8 o7… o0
875	14015... 14000	0	0	0
876	14031... 14016	0	0	0
877	14047... 14032	0 / Expansion module, right	Right 1 o15…o8	0
878	14063... 14048	Expansion module, right	Right 3 o15…o8	Right 2 o15… o8
879	14079... 14064	Expansion module, right	Right 5 o15…o8	Right 4 o15… o8
880	14095... 14080	Expansion module, right	Right 7 o15…o8	Right 6 o15… o8
881	14111... 14096	Expansion module, right / res	Reserved	Right 8 o15… o8
882	14127... 14112	Expansion module, left	Left 1 (o15… o8)	Left 1 (o7… o0)
883	14143... 14128	Expansion module, left	Left 1 (o31… o24)	Left 1 (o23… o16)
884	14159... 14144	Expansion module, left	Left 2 (o15… o8)	Left 2 (o7… o0)
885	14175... 14160	Expansion module, left	Left 2 (o31… o24)	Left 2 (o23… o16)

Register (3x)	Coil/ Discrete Input (1x)	Content	High Byte	Low Byte
886	14191... 14176	Expansion module, left	Left 3 (o15… o8)	Left 3 (o7… o0)
887	14207... 14192	Expansion module, left	Left 3 (o31… o24)	Left 3 (o23… o16)
888	14223... 14208	0		
889	14239... 14224	Expansion module, left	Left 4 (o15… o8)	Left 4 (o7… o0)
890	14255... 14240	Expansion module, left	Left 4 (o31… o24)	Left 4 (o23… o16)
891	14271... 14256	Expansion module, left	Left 5 (o15… o8)	Left 5 (o7… o0)
892	14287... 14272	Expansion module, left	Left 5 (o31… o24)	Left 5 (o23… o16)
893	14303... 14288	Expansion module, left	Left 6 (o15… o8)	Left 6 (o7… o0)
894	14319... 14304	Expansion module, left	Left 6 (o31… o24)	Left 6 (o23… o16)
895	14335... 14320	0		

6.4.4.8 LED status

The table below describes the Modbus/TCP data areas that contain the LED status.

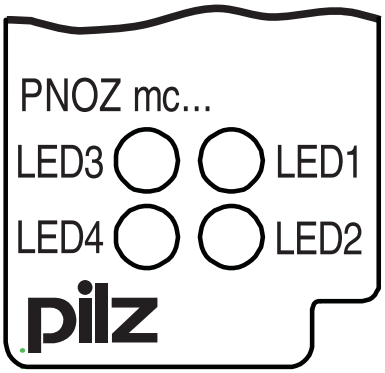
Relevant areas for the data are defined in the Modbus/TCP data areas Discrete Inputs (1x) and Input Register (3x). Read access is available to these data areas.

Register (3x)	Discrete Input (1x)	Content	High Byte	Low Byte
896	14351... 14336	LEDs RUN / DIAG	Diag	Run
897	14367... 14352	LEDs FAULT/IFAULT	I Fault	Fault
898	14383... 14368	LEDs OFAULT / Expansion module, right	Right 1	O Fault
899	14399... 14384	LEDs expansion module, right	Right 3	Right 2
900	14415... 14400	LEDs	Right 5	Right 4
901	14431... 14416	LEDs	Right 7	Right 6
902	14447... 14432	LEDs / res	Reserved	Right 8
903	14463... 14448	LED base unit i0 - i15	LED i15…i8	LED i7…i0
904	14479... 14464	LED base unit i16-i19 / 0	0	LED i19…i16
905	14495... 14480	0 / LEDs expansion module, right	LED right 1	0
906	14511... 14496	LEDs expansion module, right	LED right 3	LED right 2

Register (3x)	Discrete Input (1x)	Content	High Byte	Low Byte
907	14527... 14512	LEDs expansion module, right	LED right 5	LED right 4
908	14543... 14528	LEDs expansion module, right	LED right 7	LED right 6
909	14559... 14544	LED right 8 / res	Reserved	LED right 8
910	14575... 14560	LEDs fieldbus status	LED 2	LED 1
911	14591... 14576	LEDs fieldbus status	LED 4	LED 3
912	14607... 14592	0		
913	14623... 14608	0		
914	14639... 14624	0		
915	14655... 14640	0		
916	14671... 14656	0		
917	14687... 14672	LEDs speed monitor 1	Axis 2	Axis 1
918	14703... 14688	LEDs speed monitor 2	Axis 2	Axis 1
919	14719... 14704	LEDs speed monitor 3	Axis 2	Axis 1
920	14735... 14720	LEDs speed monitor 4	Axis 2	Axis 1
921	14751... 14736	0		
922	14767... 14752	0		
923	14783... 14768	0		
924	14799... 14784	LEDs expansion module, left	Left 2	Left 1
925	14815... 14800	LEDs expansion module, left	Left 4	Left 3
926	14831... 14816	LEDs expansion module, left	Left 6	Left 5
927	14847... 14832	0		
928	14863... 14848	0		
929	14879... 14864	0		
930	14895... 14880	0		

Register 896 "LEDs" and Register 924 to 926 “LEDs expansion module, left"
0x00 = LED off
0xFF = LED on
0x30 = LED flashes

Register 910 to 911 "LEDs fieldbus"
Position of LED1 ... LED4 PNOZmulti:



Position of LED1 ... LED4 PNOZmulti Mini:

0x00 = LED off
0x01 = LED green
0x02 = LED red

The LED functions are described in the relevant operating manual.

Register 917 to 920 "LEDs speed monitor 1 ... 4"
State of the LEDs on the speed monitors
PNOZ ms1p, PNOZ ms2p:
I10, I11, I20, I21, X12, X22
PNOZ ms3p:
X12, X22
PNOZ ms4p:
X12

Bit	7	6	5	4	3	2	1	0
Axis 1	0	0	I11	I11	I10	I10	0	X12
Axis 2	0	0	I21	I21	I20	I20	0	X22

LEDs for proximity switch: I10, I11, I20, I21:
If the LED is lit, the corresponding Bit will contain "1". The proximity switch is energised.
LEDs for incremental encoder: X12, X22:
If the LED is lit, the corresponding Bit will contain "1". The incremental encoder is connected correctly.
The LED functions are described in the operating manuals for the speed monitors.

6.4.4.9 Diagnostic word, element types

The table below describes the Modbus/TCP data areas that contain information on the elements in the PNOZmulti Configurator and the diagnostic word.

Relevant areas for the data are defined in the Modbus/TCP data areas Discrete Inputs (1x) and Input Register (3x). Read access is available to these data areas.

Register (3x)	Coil/ Discrete Input (1x)	Content	High Byte	Low Byte
931	14911... 14896	Number of elements that can store a state	0	Number
932	14927... 14912	Reserved		
933	14943... 14928	Reserved		
934	14959... 14944	Reserved		
935	14975... 14960	Reserved		
936	14991... 14976	Reserved		
937	15007... 14992	Reserved		
938	15023... 15008	Element enable 1-16	16…9	8…1
939	15039... 15024	Element enable 17-32	32…25	24…16
940	15055... 15040	Element enable 33-48	48…41	40…33
941	15071... 15056	Element enable 49-64	64…57	56…49
942	15087... 15072	Element enable 65-80	80…73	72…65
943	15103... 15088	Element enable 81-96	96…89	88…81
944	15119... 15104	Element enable 96-100 / reserved	Reserved	100…96
945	15135... 15120	Reserved		
946	15151... 15136	Reserved		
947	15167... 15152	Reserved		
948	15183... 15168	Reserved		
949	15199... 15184	Reserved		
950	15215... 15200	Reserved		
951	15231... 15216	Reserved		

Register (3x)	Coil/ Discrete Input (1x)	Content	High Byte	Low Byte
952	15247... 15232	Diagnostic word 1	Bit 15… 8	Bit 7… 0
953	15263... 15248	Diagnostic word 2	Bit 15… 8	Bit 7… 0
954	15279... 15264	Diagnostic word 3	Bit 15… 8	Bit 7… 0
955	15295... 15280	Diagnostic word 4	Bit 15… 8	Bit 7… 0
956	15311... 15296	Diagnostic word 5	Bit 15… 8	Bit 7… 0
957	15327... 15312	Diagnostic word 6	Bit 15… 8	Bit 7… 0
958	15343... 15328	Diagnostic word 7	Bit 15… 8	Bit 7… 0
959	15359... 15344	Diagnostic word 8	Bit 15… 8	Bit 7… 0
960	15375... 15360	Diagnostic word 9	Bit 15… 8	Bit 7… 0
961	15391... 15376	Diagnostic word 10	Bit 15… 8	Bit 7… 0
962	15407... 15392	Diagnostic word 11	Bit 15… 8	Bit 7… 0
963	15423... 15408	Diagnostic word 12	Bit 15… 8	Bit 7… 0
964	15439... 15424	Diagnostic word 13	Bit 15… 8	Bit 7… 0
965	15455... 15440	Diagnostic word 14	Bit 15… 8	Bit 7… 0
966	15471... 15456	Diagnostic word 15	Bit 15… 8	Bit 7… 0
967	15487... 15472	Diagnostic word 16	Bit 15… 8	Bit 7… 0
968	15503... 15488	Diagnostic word 17	Bit 15… 8	Bit 7… 0
969	15519... 15504	Diagnostic word 18	Bit 15… 8	Bit 7… 0
970	15535... 15520	Diagnostic word 19	Bit 15… 8	Bit 7… 0
971	15551... 15536	Diagnostic word 20	Bit 15… 8	Bit 7… 0
972	15567... 15552	Diagnostic word 21	Bit 15… 8	Bit 7… 0
973	15583... 15568	Diagnostic word 22	Bit 15… 8	Bit 7… 0
974	15599... 15584	Diagnostic word 23	Bit 15… 8	Bit 7… 0
975	15615... 15600	Diagnostic word 24	Bit 15… 8	Bit 7… 0
976	15631... 15616	Diagnostic word 25	Bit 15… 8	Bit 7… 0
977	15647... 15632	Diagnostic word 26	Bit 15… 8	Bit 7… 0
978	15663... 15648	Diagnostic word 27	Bit 15… 8	Bit 7… 0
979	15679... 15664	Diagnostic word 28	Bit 15… 8	Bit 7… 0
980	15695... 15680	Diagnostic word 29	Bit 15… 8	Bit 7… 0
981	15711... 15696	Diagnostic word 30	Bit 15… 8	Bit 7… 0
982	15727... 15712	Diagnostic word 31	Bit 15… 8	Bit 7… 0
983	15743... 15728	Diagnostic word 32	Bit 15… 8	Bit 7… 0
984	15759... 15744	Diagnostic word 33	Bit 15… 8	Bit 7… 0
985	15775... 15760	Diagnostic word 34	Bit 15… 8	Bit 7… 0
986	15791... 15776	Diagnostic word 35	Bit 15… 8	Bit 7… 0
987	15807... 15792	Diagnostic word 36	Bit 15… 8	Bit 7… 0

Register (3x)	Coil/ Discrete Input (1x)	Content	High Byte	Low Byte
988	15823... 15808	Diagnostic word 37	Bit 15… 8	Bit 7… 0
989	15839... 15824	Diagnostic word 38	Bit 15… 8	Bit 7… 0
990	15855... 15840	Diagnostic word 39	Bit 15… 8	Bit 7… 0
991	15871... 15856	Diagnostic word 40	Bit 15… 8	Bit 7… 0
992	15887... 15872	Diagnostic word 41	Bit 15… 8	Bit 7… 0
993	15903... 15888	Diagnostic word 42	Bit 15… 8	Bit 7… 0
994	15919... 15904	Diagnostic word 43	Bit 15… 8	Bit 7… 0
995	15935... 15920	Diagnostic word 44	Bit 15… 8	Bit 7… 0
996	15951... 15936	Diagnostic word 45	Bit 15… 8	Bit 7… 0
997	15967... 15952	Diagnostic word 46	Bit 15… 8	Bit 7… 0
998	15983... 15968	Diagnostic word 47	Bit 15… 8	Bit 7… 0
999	15999... 15984	Diagnostic word 48	Bit 15… 8	Bit 7… 0
1000	16015... 16000	Diagnostic word 49	Bit 15… 8	Bit 7… 0
1001	16031... 16016	Diagnostic word 50	Bit 15… 8	Bit 7… 0
1002	16047... 16032	Diagnostic word 51	Bit 15… 8	Bit 7… 0
1003	16063... 16048	Diagnostic word 52	Bit 15… 8	Bit 7… 0
1004	16079... 16064	Diagnostic word 53	Bit 15… 8	Bit 7… 0
1005	16095... 16080	Diagnostic word 54	Bit 15… 8	Bit 7… 0
1006	16111... 16096	Diagnostic word 55	Bit 15… 8	Bit 7… 0
1007	16127... 16112	Diagnostic word 56	Bit 15… 8	Bit 7… 0
1008	16143... 16128	Diagnostic word 57	Bit 15… 8	Bit 7… 0
1009	16159... 16144	Diagnostic word 58	Bit 15… 8	Bit 7… 0
1010	16175... 16160	Diagnostic word 59	Bit 15… 8	Bit 7… 0
1011	16191... 16176	Diagnostic word 60	Bit 15… 8	Bit 7… 0
1012	16207... 16192	Diagnostic word 61	Bit 15… 8	Bit 7… 0
1013	16223... 16208	Diagnostic word 62	Bit 15… 8	Bit 7… 0
1014	16239... 16224	Diagnostic word 63	Bit 15… 8	Bit 7… 0
1015	16255... 16240	Diagnostic word 64	Bit 15… 8	Bit 7… 0
1016	16271... 16256	Diagnostic word 65	Bit 15… 8	Bit 7… 0
1017	16287... 16272	Diagnostic word 66	Bit 15… 8	Bit 7… 0
1018	16303... 16288	Diagnostic word 67	Bit 15… 8	Bit 7… 0
1019	16319... 16304	Diagnostic word 68	Bit 15… 8	Bit 7… 0
1020	16335... 16320	Diagnostic word 69	Bit 15… 8	Bit 7… 0
1021	16351... 16336	Diagnostic word 70	Bit 15… 8	Bit 7… 0
1022	16367... 16352	Diagnostic word 71	Bit 15… 8	Bit 7… 0
1023	16383... 16368	Diagnostic word 72	Bit 15… 8	Bit 7… 0

Register (3x)	Coil/ Discrete Input (1x)	Content	High Byte	Low Byte
1024	16399... 16384	Diagnostic word 73	Bit 15… 8	Bit 7… 0
1025	16415... 16400	Diagnostic word 74	Bit 15… 8	Bit 7… 0
1026	16431... 16416	Diagnostic word 75	Bit 15… 8	Bit 7… 0
1027	16447... 16432	Diagnostic word 76	Bit 15… 8	Bit 7… 0
1028	16463... 16448	Diagnostic word 77	Bit 15… 8	Bit 7… 0
1029	16479... 16464	Diagnostic word 78	Bit 15… 8	Bit 7… 0
1030	16495... 16480	Diagnostic word 79	Bit 15… 8	Bit 7… 0
1031	16511... 16496	Diagnostic word 80	Bit 15… 8	Bit 7… 0
1032	16527... 16512	Diagnostic word 81	Bit 15… 8	Bit 7… 0
1033	16543... 16528	Diagnostic word 82	Bit 15… 8	Bit 7… 0
1034	16559... 16544	Diagnostic word 83	Bit 15… 8	Bit 7… 0
1035	16575... 16560	Diagnostic word 84	Bit 15… 8	Bit 7… 0
1036	16591... 16576	Diagnostic word 85	Bit 15… 8	Bit 7… 0
1037	16607... 16592	Diagnostic word 86	Bit 15… 8	Bit 7… 0
1038	16623... 16608	Diagnostic word 87	Bit 15… 8	Bit 7… 0
1039	16639... 16624	Diagnostic word 88	Bit 15… 8	Bit 7… 0
1040	16655... 16640	Diagnostic word 89	Bit 15… 8	Bit 7… 0
1041	16671... 16656	Diagnostic word 90	Bit 15… 8	Bit 7… 0
1042	16687... 16672	Diagnostic word 91	Bit 15… 8	Bit 7… 0
1043	16703... 16688	Diagnostic word 92	Bit 15… 8	Bit 7… 0
1044	16719... 16704	Diagnostic word 93	Bit 15… 8	Bit 7… 0
1045	16735... 16720	Diagnostic word 94	Bit 15… 8	Bit 7… 0
1046	16751... 16736	Diagnostic word 95	Bit 15… 8	Bit 7… 0
1047	16767... 16752	Diagnostic word 96	Bit 15… 8	Bit 7… 0
1048	16783... 16768	Diagnostic word 97	Bit 15… 8	Bit 7… 0
1049	16799... 16784	Diagnostic word 98	Bit 15… 8	Bit 7… 0
1050	16815... 16800	Diagnostic word 99	Bit 15… 8	Bit 7… 0
1051	16831... 16816	Diagnostic word 100	Bit 15… 8	Bit 7… 0
1052	16847... 16832	Reserved		
1053	16863... 16848	Reserved		
1054	16879... 16864	Reserved		
1055	16895... 16880	Reserved		
1056	16911... 16896	Reserved		
1057	16927... 16912	Reserved		
1058	16943... 16928	Reserved		
1059	16959... 16944	Reserved		

Register (3x)	Coil/ Discrete Input (1x)	Content	High Byte	Low Byte
1060	16975... 16960	Reserved		
1061	16991... 16976	Reserved		
1062	17007... 16992	Reserved		
1063	17023... 17008	Reserved		
1064	17039... 17024	Reserved		
1065	17055... 17040	Reserved		
1066	17071... 17056	Reserved		
1067	17087... 17072	Reserved		
1068	17103... 17088	Reserved		
1069	17119... 17104	Reserved		
1070	17135... 17120	Reserved		
1071	17151... 17136	Element type	Element ID = 2	Element ID = 1
1072	17167... 17152	Element type	Element ID = 4	Element ID = 3
1073	17183... 17168	Element type	Element ID = 6	Element ID = 5
1074	17199... 17184	Element type	Element ID = 8	Element ID = 7
1075	17215... 17200	Element type	Element ID = 10	Element ID = 9
1076	17231... 17216	Element type	Element ID = 12	Element ID = 11
1077	17247... 17232	Element type	0	Element ID = 13
1078	17263... 17248	Element type	Element ID = 15	Element ID = 14
1079	17279... 17264	Element type	Element ID = 17	Element ID = 16
1080	17295... 17280	Element type	Element ID = 19	Element ID = 18
1081	17311... 17296	Element type	Element ID = 21	Element ID = 20
1082	17327... 17312	Element type	Element ID = 23	Element ID = 22
1083	17343... 17328	Element type	Element ID = 25	Element ID = 24
1084	17359... 17344	Element type	0	Element ID = 26
1085	17375... 17360	Element type	Element ID = 15	Element ID = 27
1086	17391... 17376	Element type	Element ID = 17	Element ID = 29
1087	17407... 17392	Element type	Element ID = 19	Element ID = 31
1088	17423... 17408	Element type	Element ID = 21	Element ID = 33
1089	17439... 17424	Element type	Element ID = 23	Element ID = 35
1090	17455... 17440	Element type	Element ID = 25	Element ID = 37
1091	17471... 17456	Element type	0	Element ID = 39
1092	17487... 17472	Element type	Element ID = 41	Element ID = 40
1093	17503... 17488	Element type	Element ID = 43	Element ID = 42
1094	17519... 17504	Element type	Element ID = 45	Element ID = 44
1095	17535... 17520	Element type	Element ID = 47	Element ID = 46

Register (3x)	Coil/ Discrete Input (1x)	Content	High Byte	Low Byte
1096	17551... 17536	Element type	Element ID = 49	Element ID = 48
1097	17567... 17552	Element type	Element ID = 51	Element ID = 50
1098	17583... 17568	Element type	0	Element ID = 52
1099	17599... 17584	Element type	Element ID = 54	Element ID = 53
1100	17615... 17600	Element type	Element ID = 56	Element ID = 55
1101	17631... 17616	Element type	Element ID = 58	Element ID = 57
1102	17647... 17632	Element type	Element ID = 60	Element ID = 59
1103	17663... 17648	Element type	Element ID = 62	Element ID = 61
1104	17679... 17664	Element type	Element ID = 64	Element ID = 63
1105	17695... 17680	Element type	0	Element ID = 65
1106	17711... 17696	Element type	Element ID = 67	Element ID = 66
1107	17727... 17712	Element type	Element ID = 69	Element ID = 68
1108	17743... 17728	Element type	Element ID = 71	Element ID = 70
1109	17759... 17744	Element type	Element ID = 73	Element ID = 72
1110	17775... 17760	Element type	Element ID = 75	Element ID = 74
1111	17791... 17776	Element type	Element ID = 77	Element ID = 76
1112	17807... 17792	Element type	0	Element ID = 78
1113	17823... 17808	Element type	Element ID = 80	Element ID = 79
1114	17839... 17824	Element type	Element ID = 82	Element ID = 81
1115	17855... 17840	Element type	Element ID = 84	Element ID = 83
1116	17871... 17856	Element type	Element ID = 86	Element ID = 85
1117	17887... 17872	Element type	Element ID = 88	Element ID = 87
1118	17903... 17888	Element type	Element ID = 90	Element ID = 89
1119	17919... 17904	Element type	0	Element ID = 91
1120	17935... 17920	Element type	Element ID = 93	Element ID = 92
1121	17951... 17936	Element type	Element ID = 95	Element ID = 94
1122	17967... 17952	Element type	Element ID = 97	Element ID = 96
1123	17983... 17968	Element type	Element ID = 99	Element ID = 98
1124	17999... 17984	Element type	Reserved	Element ID = 100
1125	18015... 18000	Element type	Reserved	Reserved
1126	18031... 18016	Element type	Reserved	Reserved

Register 938 to 944 "Element enable 1 ... 100"

Each element is assigned an ID in the PNOZmulti Configurator. If the element's output = 0 (no enable), the corresponding bit is set.

Byte 0	8	7	6	5	4	3	2	1
Byte 1	16	15	14	13	12	11	10	9

Byte 2	24	23	22	21	20	19	18	17
...								
Byte 10	88	87	86	85	84	83	82	81
Byte 11	96	95	94	93	92	91	90	89
Byte 12	-	-	-	-	100	99	98	97

Register 1071 to 1126 "Element type"

Please refer to the chapter entitled Element types [📖 173] in the Appendix

6.4.4.10 Current state of the virtual inputs

The table below describes the Modbus/TCP data areas that contain the current state of the virtual inputs. These are the virtual inputs that can be set by various subscribers (e.g. fieldbus).

Relevant areas for the data are defined in the Modbus/TCP data areas Discrete Inputs (1x) and Input Register (3x). Read access is available to these data areas.

Register (3x)	Coil/ Discrete Input (1x)	Content	High Byte	Low Byte
1127	18047... 18032	Inputs, feedback status 0...15	i15…i8	i7…i0
1128	18063... 18048	Inputs, feedback status 16...31	i31…i24	i23…i16
1129	18079... 18064	Inputs, feedback status 32...47	i47…i40	i39…i32
1130	18095... 18080	Inputs, feedback status 48...63	i63…i56	i55…i48
1131	18111... 18096	Inputs, feedback status 64...79	i79…i72	i71…i64
1132	18127... 18112	Inputs, feedback status 80...95	i95…i88	i87…i80
1133	18143... 18128	Inputs, feedback status 96...111	i111…i104	i103…i96
1134	18159... 18144	Inputs, feedback status 112...127	i127…i120	i119…i112
1135	18175... 18160	Reserved		
1136	18191... 18176	Reserved		
1137	18207... 18192	Reserved		
1138	18223... 18208	Reserved		
1139	18239... 18224	Reserved		
1140- 2047		Reserved		

6.4.4.11 Current states of the virtual inputs for the safe Ethernet connection

The table below describes the Modbus/TCP data areas that contain the current states of the virtual inputs and outputs for the ***Safe Ethernet connection***. These are virtual inputs and outputs that are transmitted via the safe Ethernet connection.

Relevant areas for the data are defined in the Modbus/TCP data areas Discrete Inputs (1x) and Input Register (3x). Read access is available to these data areas.

Register (3x)	Coil/ Discrete Input (1x)	Content	High Byte	Low Byte
1141	18271... 18256	Inputs i0...i15		
1142	18287... 18272	Inputs i16...i31		
1143	18303... 18288	Inputs i32...i47		
1144	18319... 18304	Outputs o0...o15		
1145	18335... 18320	Outputs O16...o31		
1146	18351... 18336	Outputs O32...o47		

6.4.4.12 State of process data

The table below describes the Modbus/TCP data areas that contain the status info register. The general state of the data is reproduced in the status info register.

Relevant areas for the data are defined in the Modbus/TCP data areas Discrete Inputs (1x) and Input Register (3x). Read access is available to these data areas.

Register (3x)	Coil/ Discrete Input (1x)	Content	High Byte	Low Byte
2048	32783... 32768	Status information	See below	

High Byte	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Low Byte	Reserved	Reserved	WD Timeout	Reserved	Reserved	Reserved	Reserved	Global Error Bit

Bit 5 "WD Timeout": The set watchdog for the input bits has been triggered and the inputs set to "0".
Bit 0 "Error Bit": The content of the data areas is not current or the set watchdog for the input bits has been triggered.

6.4.4.13 Safe Ethernet connection

The table below describes the Modbus/TCP data areas that contain the data for the safe Ethernet connection. A description of how to use this data is provided in the chapter entitled "Safe Ethernet Connection".

Register (3x)	Coil/ Discrete Input (1x)	Content	High Byte	Low Byte
20000 - 20017	-	Safe Ethernet connection send data		
21000 - 21017	-	Safe Ethernet connection receive data		

6.4.5 Updating the data areas

Data is updated with varying priority.

The table below shows the typical update cycles for the various data.

Content	Typ. update cycle
Virtual inputs/outputs	20 ms
Configuration	Once during initialisation
State of the inputs/outputs from the base unit and expansion modules	320 ms
LED status	1000 ms
Number of elements that can store a state	Once during initialisation
Element enable	320 ms
Diagnostic words	1000 ms
Element types	Once during initialisation
Current state of the virtual inputs	1000 ms

Information

The update time may increase if there are additional TCP/IP connections (e.g. PNOZmulti Configurator, PMI, control system) on the PG port (Port 9000).

6.4.6 Bit addressing in a Register

Addressing the virtual inputs (coils) on the PNOZmulti

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Register 0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Register 1	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
Register 2	Bit 47	Bit 46	Bit 45	Bit 44	Bit 43	Bit 42	Bit 41	Bit 40	Bit 39	Bit 38	Bit 37	Bit 36	Bit 35	Bit 34	Bit 33	Bit 32
Register 3	Bit 63	Bit 62	Bit 61	Bit 60	Bit 59	Bit 58	Bit 57	Bit 56	Bit 55	Bit 54	Bit 53	Bit 52	Bit 51	Bit 50	Bit 49	Bit 48
Register 4	Bit 79	Bit 78	Bit 77	Bit 76	Bit 75	Bit 74	Bit 73	Bit 72	Bit 71	Bit 70	Bit 69	Bit 68	Bit 67	Bit 66	Bit 65	Bit 64
Register 5	Bit 95	Bit 94	Bit 93	Bit 92	Bit 91	Bit 90	Bit 89	Bit 88	Bit 87	Bit 86	Bit 85	Bit 84	Bit 83	Bit 82	Bit 81	Bit 80
Register 6	Bit 111	Bit 110	Bit 109	Bit 108	Bit 107	Bit 106	Bit 105	Bit 104	Bit 103	Bit 102	Bit 101	Bit 100	Bit 99	Bit 98	Bit 97	Bit 96
Register 7	Bit 127	Bit 126	Bit 125	Bit 124	Bit 123	Bit 122	Bit 121	Bit 120	Bit 119	Bit 118	Bit 117	Bit 116	Bit 115	Bit 114	Bit 113	Bit 112

Addressing the virtual outputs (discrete inputs) on the PNOZmulti

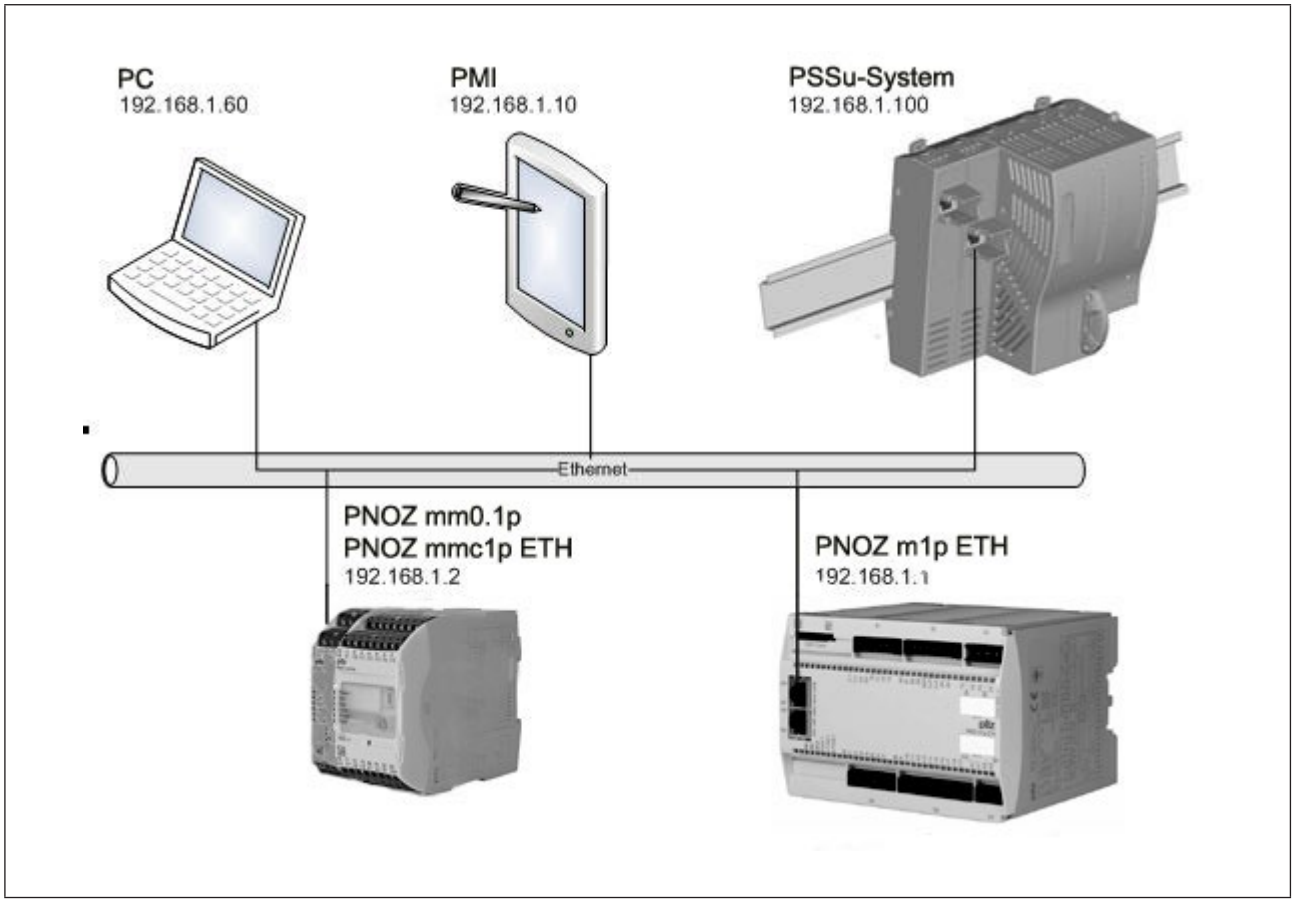
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Register 512	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Register 513	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
Register 514	Bit 47	Bit 46	Bit 45	Bit 44	Bit 43	Bit 42	Bit 41	Bit 40	Bit 39	Bit 38	Bit 37	Bit 36	Bit 35	Bit 34	Bit 33	Bit 32
Register 515	Bit 63	Bit 62	Bit 61	Bit 60	Bit 59	Bit 58	Bit 57	Bit 56	Bit 55	Bit 54	Bit 53	Bit 52	Bit 51	Bit 50	Bit 49	Bit 48
Register 516	Bit 79	Bit 78	Bit 77	Bit 76	Bit 75	Bit 74	Bit 73	Bit 72	Bit 71	Bit 70	Bit 69	Bit 68	Bit 67	Bit 66	Bit 65	Bit 64
Register 517	Bit 95	Bit 94	Bit 93	Bit 92	Bit 91	Bit 90	Bit 89	Bit 88	Bit 87	Bit 86	Bit 85	Bit 84	Bit 83	Bit 82	Bit 81	Bit 80
Register 518	Bit 111	Bit 110	Bit 109	Bit 108	Bit 107	Bit 106	Bit 105	Bit 104	Bit 103	Bit 102	Bit 101	Bit 100	Bit 99	Bit 98	Bit 97	Bit 96
Register 519	Bit 127	Bit 126	Bit 125	Bit 124	Bit 123	Bit 122	Bit 121	Bit 120	Bit 119	Bit 118	Bit 117	Bit 116	Bit 115	Bit 114	Bit 113	Bit 112

6.5 Example

The following subscribers communicate via Modbus/TCP or Ethernet:

- Devices with server role:
  - PNOZmulti base unit PNOZ m1p ETH
  - PNOZmulti Mini base unit PNOZ mm0.1p with the expansion module PNOZ mmc1p ETH
- Devices with client role:
  - PSSu system in the automation system PSS 4000
  - Operator terminal PMI
- PC as programming device for PNOZmulti, PSSu system and PMI

The PSSu system and the operator terminal PMI access both configurable control systems PNOZmulti (server roles) simultaneously.



7 Safe Ethernet connection

7.1 Overview

The safe Ethernet connection enables a point-to-point connection between a PNOZmulti base unit and a PSS 4000 device. Up to 48 safe virtual inputs and outputs can be transmitted via this connection.

7.2 System requirements

- PNOZmulti Configurator: from Version 9.3.0
- All base units from the configurable control system PNOZmulti (PNOZ mxp) that have an Ethernet interface, from V6.7.

Please contact Pilz if you have an older version.

7.3 Function description

The Safe Ethernet connection is used for safe communication between a PNOZmulti base unit and a PSS 4000 device, based on Industrial Ethernet. The underlying protocol is Modbus/TCP.

A point-to-point connection (1:1 communication relationship) can be implemented via the safe Ethernet connection. Information can be exchanged between a PNOZmulti and a PSS4000 via 48 safe virtual inputs and 48 safe virtual outputs.

Data exchange via the non-safety-related communication medium (Ethernet, Modbus/TCP) achieves safety in two ways: by using safety-related blocks in both communication partners and by the fact that the blocks are executed in the safety-related part of the control system.

This procedure corresponds to the black channel principle in accordance with EN/IEC 61784-3. Depending on the application area and its respective regulations, the block can be used in accordance with EN/IEC 61508 up to SIL3 and up to PLe (Cat.4) in accordance with EN ISO 13849-1.

A safe Ethernet connection between a PNOZmulti and a PSS 4000 device can be regarded as a connection that provides communication in both directions. The communication partners continually try to send, even if the connection is broken. If the connection is error-free, then it can be restored via a reset on the receiver side.
To establish a full connection in both communication directions, the connection must be reset on both sides.

7.4 Configuration in the PNOZmulti Configurator

The connection settings for the PNOZmulti are made in the PNOZmulti Configurator in the ***Safe Ethernet Connection Status*** element. This is where the local address, remote address and timeout are set. For details of the configuration please refer to the online help for the PNOZmulti Configurator.

- **Local address**

  Own connection address, must be different from the remote address.

- **Remote address**

  Connection address of the communication partner, must be different from the local address. (Local address of the communication partner.)

- **Timeout**

  Timeout is the monitoring time for a telegram's runtime. The monitoring time is jointly responsible for the safety function's reaction time and should therefore be set as low as possible. However, if the rated monitoring time is too short, frequent connection failures may result.
  Recommended configuration when connecting PSS 4000 and PNOZmulti:

  Necessary condition:
  $t_{SecTimeout} \geq (2 \times t_{MultiProcessing}) + (4 \times t_{PssTask})$,
  Sufficient condition:
  $t_{SecTimeout} = (k \times t_{MultiProcessing})$; where k=1, 2, 3…

  Please refer to the operating instructions of the PNOZmulti base unit for the maximum processing time for data communication $t_{MultiProcessing}$.

  The task cycle time $t_{PssTask}$ is the task cycle time configured in PAS 4000 and is the time needed to call the FS_SafeEthernetConnection block in the PSS 4000.

  The timeout value $t_{SecTimeout}$ can only accept an integer multiple value of $t_{MultiProcessing}$ and must be rounded up if necessary.

  The same value should be configured as the timeout value $t_{SecTimeout}$ for both communication partners.

WARNING!

Loss of safety function due to brevity of signals!

The payload must be present for at least the monitoring time Timeout, otherwise certain communication errors in the receiver cannot be detected. Make sure that the payload is available in the transmitter for at least the monitoring time $t_{SecTimeout}$ to enable the receiver to evaluate it safely.

7.5 Modbus configuration

Data exchange is based on Ethernet. The underlying protocol is Modbus/TCP.

The PNOZmulti is always the Server for the Modbus/TCP with a fixed pre-configuration for the safe Ethernet connection.

Information regarding the configuration of Modbus/TCP for PSS 4000

Data exchange is defined by the possibilities and requirements of Modbus/TCP. The blocks uses the holding registers (4x) for data exchange. FC 23 (Read/Write Multiple Registers) must be configured as the function code (FC) for Client connections.

For communication between two devices, a Modbus/TCP connection must be configured for each device. Modbus/TCP requires that one of the communication partners is configured as the Client connection and the other as the Server connection. The PNOZmulti can only act as Server. The PSS4000 must be configured as the Client.

The Modbus address at which the PNOZmulti makes the send/receive data available as Server is not configurable in the PNOZmulti.

The PNOZmulti's send data (18 Register) can be found from start address 20000 (signifies HoldingRegister 4x20000)

The PNOZmulti's receive data (18 Register) can be found from start address 21000 (signifies HoldingRegister 4x21000)

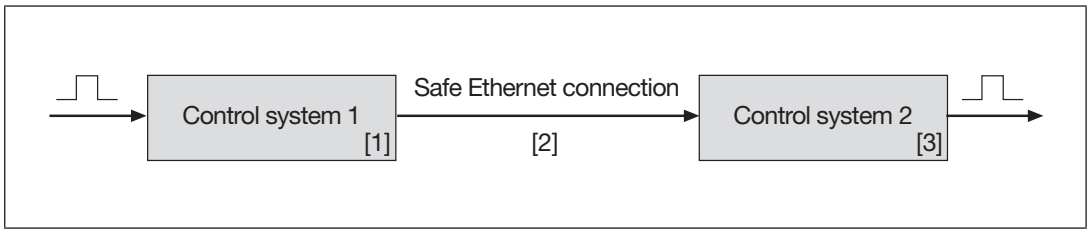
These addresses must be configured accordingly in the PSS 4000.

7.6 Reaction time

The safety function's safe reaction time is composed of the reaction times of the control systems and the monitoring time for a telegram's runtime.

Composition of the entire data path

The entire data path is composed of the data subpaths of a control system 1, the safe Ethernet connection and the data subpaths of a control system 2.



Data path 1: Control system 1 (transmitter)

Data path 1 describes the time between the signal changing at the input of control system 1 and the signal being present in the ***Safe Ethernet connection*** output area of control system 1.

If control system 1 (transmitting system) is an automation system PSS 4000, you can calculate the reaction time of this data path as described in the online help for PAS4000.

If control system 1 (transmitting system) is a control system PNOZmulti, you can calculate the reaction time as follows:

- Max. input delay (see Technical details in the operating instructions for the respective input) + Max. cycle time of the device (see base unit's operating instructions)

Data path 2: Safe Ethernet connection (transmission)

Data path 2 describes the time between the signal being present in the ***Safe Ethernet connection*** output area of control system 1 and the signal being present in the ***Safe Ethernet connection*** input area of control system 2.

The reaction time of data path 2 corresponds to the configured timeout time $t_{SecTimeout}$ of the receiver system.

Data path 3: Control system 2 (transmitter)

Data path 3 describes the time between the signal being present in the ***Safe Ethernet connection*** output area of control system 2 and the output switching in control system 2.

If control system 2 (receiving system) is an automation system PSS 4000, you can calculate the reaction time of this data path as described in the online help for PAS4000.

If control system 2 (receiving system) is a control system PNOZmulti, you can calculate the reaction time as follows:

- Max. switch-off delay of the output (see Technical details in the operating instructions for the respective output)

Overall reaction time

The reaction time t_{React_max} from the signal changing at the input on control system 1 to the output switching on control system 2 is calculated by adding the reaction times of the three individual data paths.

Series connection

If several control systems are connected in series and information is channeled via several ***Safe Ethernet connections***, each transmission must be calculated as a standalone connection (consisting of the three data subpaths) and the reaction times are added together.

Example: Input PNOZmulti base unit – Output PSS 4000 PLC

Data path	PNOZmulti	PSS 4000
1	Max. input delay + Max. cycle time of the device	
2	Calculated timeout time $t_{SecTimeout}$: (2 x processing time for data communication $t_{MultiProcessing}$) + (4 x task cycle time $t_{PssTask}$)	
3		Cycle time with ext. communication $t_{extCo\_Task2\_max}$ + Reaction time of module bus $t_{Task2\_MBUS\_max}$

Max. input delay PNOZmulti (see base unit's operating instructions)	4 ms
Max. cycle time of device (see base unit's operating instructions)	15 ms
Max. processing time for data communication ($t_{MultiProcessing}$) (see operating instructions for PNOZmulti base unit)	*50 ms*
Configured task cycle time in the PSS 4000 ($t_{PSSTask}$)	*10 ms*
Calculated timeout time (see Configuration in the PNOZmulti Configurator [119])	150 ms
$t_{extCo\_Task2\_max}$ (see example in the online help for PAS4000)	100 ms
$t_{Task2\_MBUS\_max}$ (see example in the online help for PAS4000)	15 ms

Reaction time t_{React_max} = 4 ms + 15 ms + 150 ms + 100 ms + 15 ms

Reaction time t_{React_max} = 284 ms

Example: Input PNOZ ml2p – Output PSS 4000 PLC

Data path	PNOZmulti	PSS 4000
1	Max. input delay + Max. cycle time of the device	
2	Calculated timeout time $t_{SecTimeout}$: (2 x processing time for data communication $t_{MultiProcessing}$) + (4 x task cycle time $t_{PssTask}$)	
3		Cycle time with ext. communication $t_{extCo\_Task2\_max}$ + Reaction time of module bus $t_{Task2\_MBUS\_max}$

Max. input delay PNOZmulti (see operating instructions for PNOZ ml2p)	15 ms
Max. cycle time of device (see base unit's operating instructions)	15 ms
Max. processing time for data communication ($t_{MultiProcessing}$) (see operating instructions for PNOZmulti base unit)	*50 ms*
Configured task cycle time in the PSS 4000 ($t_{PSSTask}$)	*10 ms*
Calculated timeout time (see Configuration in the PNOZmulti Configurator [119])	150 ms
$t_{extCo\_Task2\_max}$ (see example in the online help for PAS4000)	100 ms
$t_{Task2\_MBUS\_max}$ (see example in the online help for PAS4000)	15 ms

Reaction time t_{React_max} = 15 ms + 15 ms + 150 ms + 100 ms + 15 ms

Reaction time t_{React_max} = 295 ms

Example: Input PSS 4000 PLC – Output PNOZ mo4p

Data path	PNOZmulti	PSS 4000
1		Reaction time of module bus $t_{MBUS\_Task1\_max}$ + Cycle time with ext. communication $t_{Task1\_ExtCo\_max}$
2	Calculated timeout time $t_{SecTimeout}$: (2 x processing time for data communication $t_{MultiProcessing}$) + (4 x task cycle time $t_{PssTask}$)	
3	Max. switch-off delay	

$t_{MBUS\_Task1\_max}$ (see example in the online help for PAS4000)	42 ms
$t_{Task1\_ExtCo\_max}$ (see example in the online help for PAS4000)	20 ms
Max. processing time for data communication ($t_{MultiProcessing}$) (see base unit's operating instructions)	*50 ms*
Configured task cycle time in the PSS 4000 ($t_{PSSTask}$)	*10 ms*
Calculated timeout time (see Configuration in the PNOZmulti Configurator [119])	150 ms
Max. switch-off delay PNOZmulti (see operating instructions for PNOZ m04p)	50 ms

Reaction time t_{React_max} = 42 ms + 20 ms + 150 ms + 50 ms

Reaction time t_{React_max} = 262 ms

7.7 Application guidelines

Connection status

The output on the ***Safe Ethernet Connection Status*** element in the user program indicates whether the data is received without error and whether there is a connection for receiving data (error-free data receipt).

If the output = "0", the connection is broken. All virtual inputs on the ***Safe Ethernet Connection*** are switched to "0". The base unit remains in a RUN condition.

When the PNOZmulti is restarted, communication is started with a falling edge on the element's input.

The cause of the error can be evaluated via the expanded PVIS diagnostic configuration (see section entitled Diagnostic word [📖 128]).

A break in the connection for receiving data has no direct effect on the connection for sending data.

Connection addresses

The connection authenticity of a point-to-point connection is tested using the connection addresses that are configured on the respective status - block/element as ***Local Address*** and ***Remote Address***.

Make sure that the connection addresses in a point-to-point connection is only used on precisely this connection within a network.

ATTENTION!

Loss of safety function due to the use of a connection address for more than one point-to-point connection in a network!

If a connection address is used in a network for more than one point-to-point connection, an unintended connection to a communication partner may result. This cannot be detected. Make sure that the connection addresses in a point-to-point connection is only used on precisely this connection within a network. It is essential to use the **Check list for connection addresses**.

Example 1: Connection addresses on a point-to-point connection with safe Ethernet connection

- Two different connection addresses are needed per point-to-point connection. In the example, connection addresses 20 and 21 are to be used.
- Other potential connections in the network may no longer use connection addresses 20 and 21.

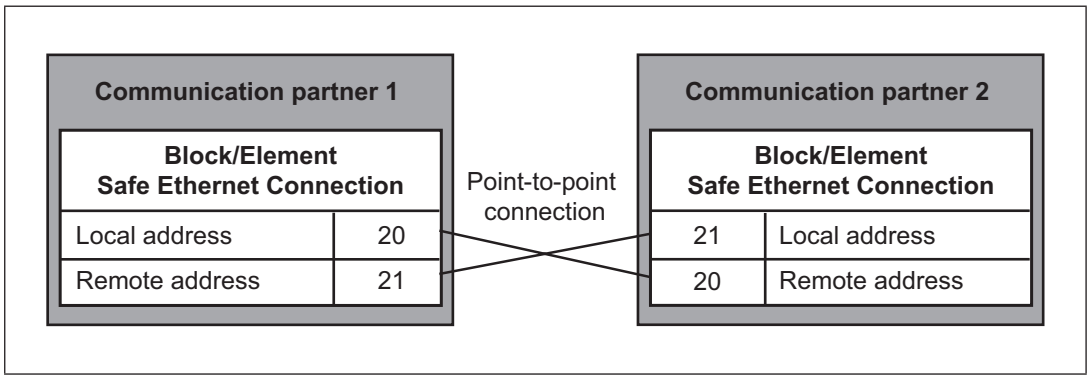


Fig.: Connection addresses on a point-to-point connection

Example 2: Connection addresses on multiple point-to-point connections with safe Ethernet connection

- Communication partner 1 maintains a point-to-point connection with communication partners 2 and 3. A total of four different connection addresses are needed for the two point-to-point connections. In the example, connection addresses 30 and 31 are to be used for point-to-point connection 1 and addresses 40 and 41 for point-to-point connection 2.
- Other potential connections in the network may no longer use connection addresses 30, 31, 40 and 41.

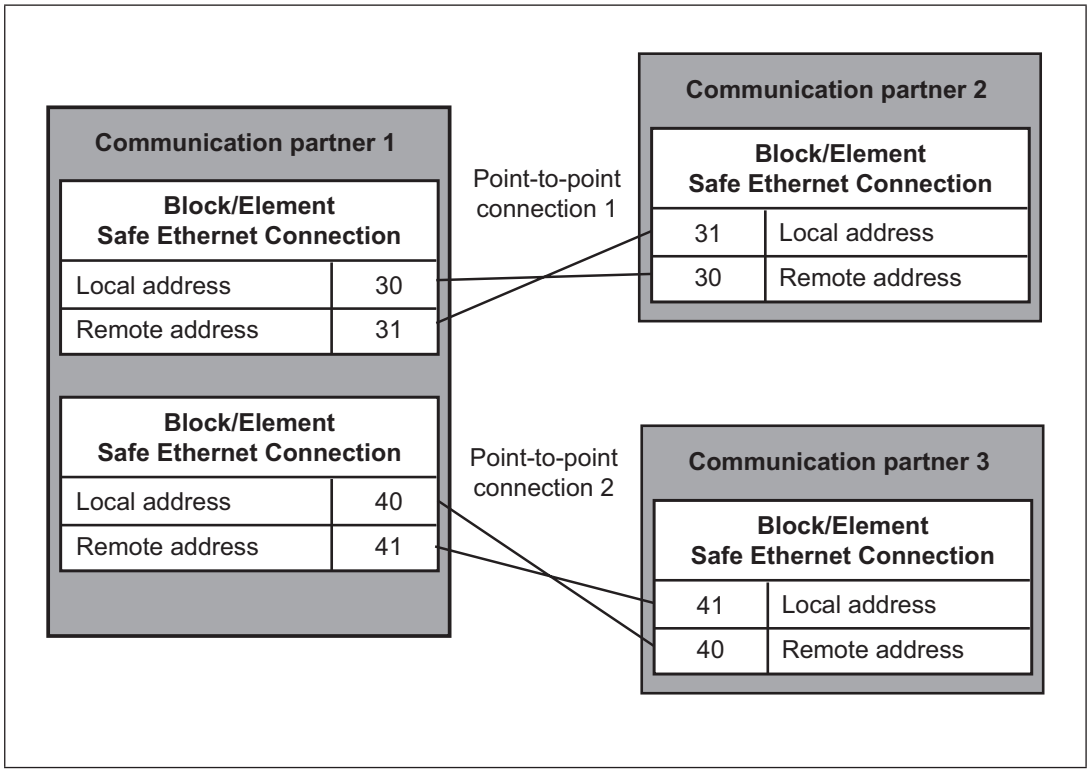


Fig.: Connection addresses on two point-to-point connections

Check list for connection addresses

As the multiple use of connection addresses cannot be avoided by technical measures, this must be achieved using organisational measures, implemented by the user.

Proceed as follows:

1. Calculate the total number of block calls

For each individual device in the overall network, calculate the number of block calls with ***Safe Ethernet connection***. We recommend using a table as shown in the following example:

Number of device	Number of block calls
1	2
2	1
3	1
4	-
Total number of block calls	**4**

2. Calculate connection addresses

Calculate the connection addresses for all point-to-point connections. You should also calculate how often a connection address has been configured. We recommend using a table as shown in the following example:

Connection address	Configuration as local address	Configuration as remote address
1	I	I
2	I	I
3	I	I
4	I	I
5 … 255	Not used	
Total number of configured connection addresses	**8**	

3. Complete check list

The following check list must be completed and the results documented:

Question	Yes	No
Have all devices in the network been recorded?	☐	☐
Have all the block calls on all devices in the network been recorded?	☐	☐
Does each configured connection address occur 1x only as a "local address"?	☐	☐
Does each configured connection address occur 1x only as a "remote address"?	☐	☐

Question	Yes	No
Does the total number of block calls calculated from step 1 and the overall number of configured connection addresses calculated from step 2 correspond to the following equation? (Total number of configured connection addresses) = 2 x (Total number of block calls)	☐	☐

Date ……..……	Signature ……..……

CAUTION!

Please note:
All questions in the check list must be answered with "Yes". If you **cannot** answer "Yes" to one of the questions, the relevant situation must be rectified. Then run through all the steps documented here again.

8 Diagnostic word

8.1 Introduction

A diagnostic word can be read out for those elements in the PNOZmulti Configurator that can store a status. The diagnostic word contains information on a particular element, such as:

- Operating states (e.g. safety gate was opened)
- Error messages (e.g. N/C contact failed to switch or switched too late)

8.2 Elements with diagnostic word

The diagnostic word is accessed by activating an Element ID. The permitted value range for the Element ID is 1 ... 100. Elements with an element ID include:

- Function elements
  - E-STOP and safety gate
  - Safety gate with guard locking
  - Light curtain
  - Enabling switch
  - Foot switch
  - Pressure sensitive mat
  - Two-hand button
  - Operating mode selector switch
  - Analogue input module
- Cascading
  - Cascading input
  - Cascading output
- Logic elements
  - RS flip-flop
  - Start element
- Speed monitor
- Press elements
  - Run monitoring
  - Rotary cam arrangement
  - Monitoring
  - Set-up mode
  - Single-stroke
  - Automatic
  - Light curtain
- Burner element

- Muting elements
  - Sequential muting
  - Parallel muting
  - Cross muting
- Safe Ethernet Connection Status element
- Output elements
  - Output elements with feedback loop
  - Safety valve

8.3 Structure of the diagnostic word

The diagnostic word has 16 Bits:

Bit	15	14	...	2	1	0

If the diagnostic word = 0, the output of the respective element = 1. The element was enabled. (Exceptions: On various function elements the states of the inputs are evaluated (see Chapter Compilation of the diagnostic words [131]).
Otherwise, at least one of the bits 0 ... 15 of the diagnostic word is set and can be evaluated,
e.g.: Bit 1 =1:00000000 00000010
Key: Safety gate has been opened

8.4 Evaluate diagnostic word

- **Evaluation in the user program**

  One bit from the diagnostic word can be linked further within the PNOZmulti user program. The user selects a bit within a diagnostic word and polls it. An LED can be driven in this way, for example.

- **Evaluation using PVIS expanded diagnostics**

  The bits of a diagnostic word can be configured for PVIS expanded diagnostics in the PNOZmulti Configurator. A "Safety Device" diagnostic type is assigned to an element. It contains the diagnostic word as an event message. An event message including remedies (actions) is defined in the diagnostic type for each event, i.e. for every potential element state. The event messages and actions can also be supplemented through additional information, which is helpful during diagnostics (equipment identifier, location description). The event messages can be displayed on the PMImicro diag, for example.

Information

Detailed information on PVIS expanded diagnostics can be found in the PNOZmulti Configurator's online help.

Evaluation via the RS232/Ethernet interfaces

The diagnostic word is requested via the interface on the base unit/communication module using the element's ID.

Information

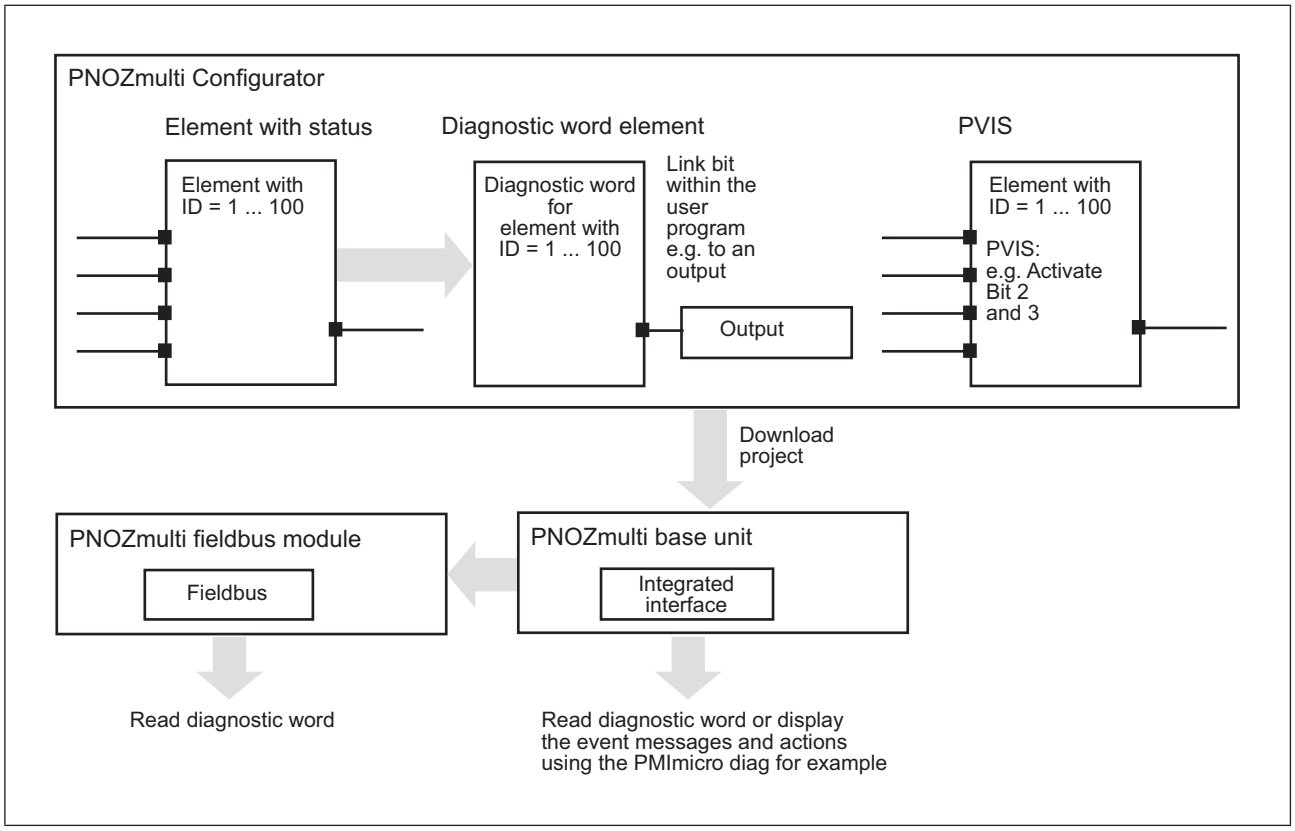
Detailed information can be found in the section entitled "RS232/Ethernet interfaces".

Evaluation via a fieldbus

The diagnostic word is requested via a connected fieldbus module, via the element's ID.

Information

Detailed information can be found in the section entitled "Fieldbus modules".



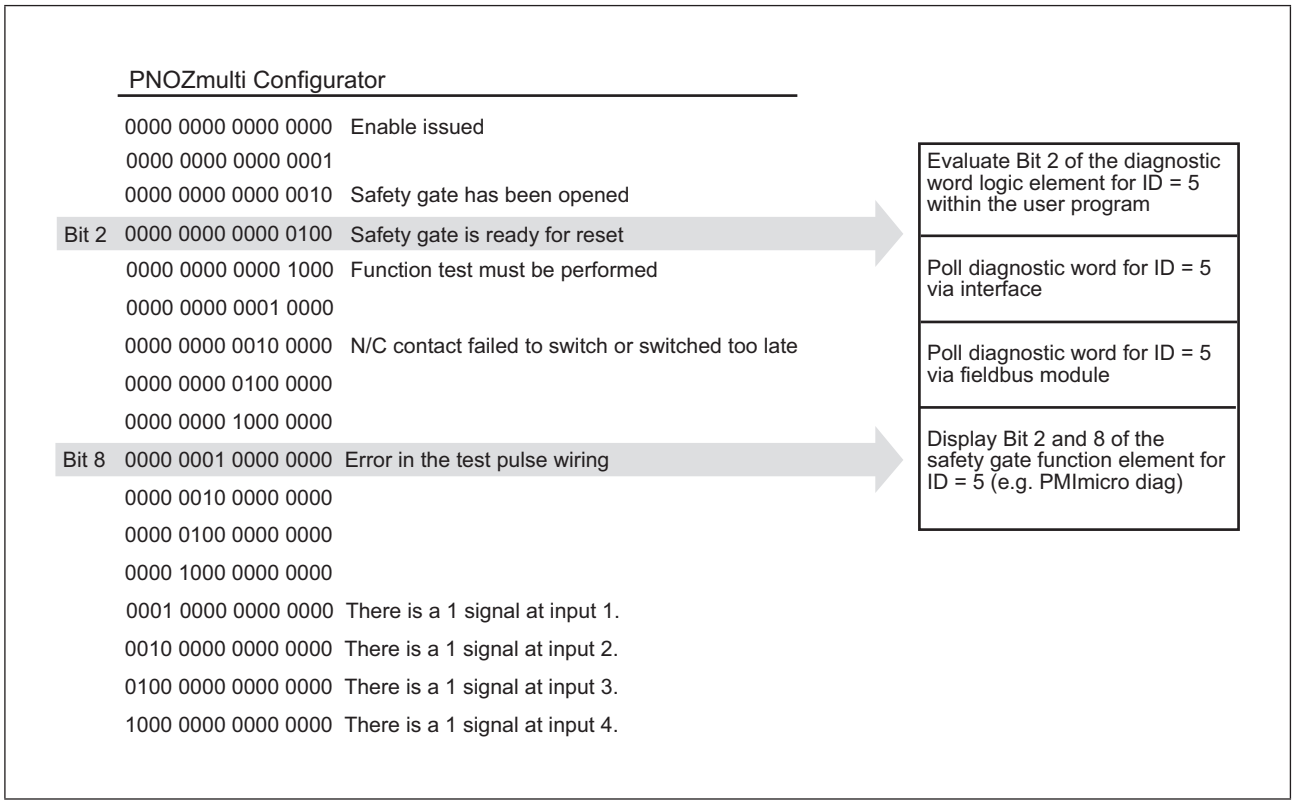
8.4.1 Example

Safety gate with Element ID = 5:

- Dual-channel
- Manual reset
- Start-up test

Evaluation of the following bits:

- Bit 2 = 1: Safety gate is ready for reset. The reset button for manual reset must be operated.
- Bit 8 = 1: Error in the test pulse wiring



8.5 Compilation of the diagnostic words

In the following tables, the respective bit =1 when the corresponding message applies. If no bit = 1, i.e. data word DW = 0, then no error is present.
Exception: On some function elements the state of the input signals is polled. The corresponding bit =1 without an error being present.

Information

If you are using PVIS expanded diagnostics, you will receive additional information (actions) on your display unit alongside the diagnostic word. Please also refer to the explanations regarding element configuration in the PNOZmulti Configurator's online help.

8.5.1 Function elements

- E-STOP
- Safety gate
- Safety gate with guard locking
- Light curtain
- Enabling switch
- Foot switch

Bit	Message	Comment
1	E-STOP: E-STOP pushbutton was operated	The safety device was triggered (E-STOP operated, safety gate opened,...)
	Safety gate, safety gate with guard locking: Safety gate was opened	
	Light curtain: Light curtain was interrupted	
	Enabling switch: Enabling switch has not been operated or is fully depressed	
	Foot switch: Foot switch must be operated	
2	- E-STOP pushbutton - Safety gate - Light curtain - Enabling switch - Foot switch is ready for reset	Manual or monitored reset was configured. The reset button has not yet been operated.
3	Function test must be performed.	A start-up test was configured, but has not yet been run.
5	N/C contact 1 or 2 failed to switch or switched too late	Simultaneity is monitored on many switch types.
8	Error in the test pulse wiring or bus error	
12	There is a 1 signal at input 1.	For information only
13	There is a 1 signal at input 2.	For information only
14	There is a 1 signal at input 3.	For information only
15	There is a 1 signal at input 4.	For information only

- Safety mat

Bit	Message	Comment
1	Safety mat was activated.	
2	Safety mat is ready for reset.	Manual reset/restart has been configured. A reset/restart is only possible if the safety mat has not been activated.
3	Start-up test must be performed.	A start-up test was configured, but has not yet been run.
5	Error caused by safety mat.	Open circuit, signal error, wiring error detected

- Two-hand pushbuttons

Bit	Message	Comment
1	Two-hand button must be operated.	Switches are in their start position.
4	Pushbutton 1 or 2 was operated too late.	Simultaneity was exceeded.
5	Pushbutton 1 or 2 was not operated.	One of the pushbuttons was operated too late or was not operated at all. Or one of the pushbuttons was operated and then released.
6	Two-hand button is deactivated.	Deactivation input is configured and =1
8	Error in the test pulse wiring.	

- Operating mode selector switch

Bit	Message	Comment
5	The input signals at the operating mode selector switch are faulty	No input is “1”.
8	Error in the test pulse wiring.	

- Analogue input module

Bit	Message	Comment
2	Analogue input module is ready for reset	Manual or monitored reset was configured. The reset button has not yet been operated.
3	Tolerance between input I0 and I1 has been exceeded	The configured permitted deviation between the measured values for i0 and i1 has been exceeded.
4	Range limit R1 has been violated.	The value is above or below the configured range limit.
5	Range limit R2 has been violated.	
6	Range limit R3 has been violated.	
7	Range limit R4 has been violated.	
8	Threshold value monitoring for the threshold L1 has been activated (Status =1).	
9	Threshold value monitoring for the threshold L2 has been activated (Status =1).	
10	Threshold value monitoring for the threshold L3 has been activated (Status =1).	
11	Threshold value monitoring for the threshold L4 has been activated (Status =1).	
12	Threshold value monitoring for the threshold L5 has been activated (Status =1).	
13	Threshold value monitoring for the threshold L6 has been activated (Status =1).	

Bit	Message	Comment
14	Threshold value monitoring for the threshold L7 has been activated (Status =1).	
15	Threshold value monitoring for the threshold L8 has been activated (Status =1).	

8.5.2 Cascading

- Cascading output

Bit	Message	Comment
8	The signal at the CO output is faulty.	For example: Fault, short circuit at CO cascading output

- Cascading input

Bit	Message	Comment
8	The signal at the CI input is faulty.	CI input is not connected to a CO output.

8.5.3 Logic elements

- RS flip-flop

Bit	Message	Comment
2	Input S is ready to set.	Input S is “0” after reset
8	There is a 1 signal at input R.	Input R =1

- Reset element

Bit	Message	Comment
2	Reset button is ready for reset.	Input signal is present, reset button can be operated.
3	Reset button is waiting for the input signal.	No input signal is present.

- Speed monitors PNOZ ms1p, PNOZ ms2p <= V 1.9

Bit	Message	Comment
2	Speed monitor is ready for reset.	Manual or monitored reset was configured. The reset button has not yet been operated.
3	Cannot monitor the speed because no speed has been selected.	With inputs n1 to n8, standstill or speed monitoring is initialised by a “1” signal. Only one input may occupy the "1" signal status.
8	Selected speed has been exceeded.	The speed at one of the active inputs n1 to n8 has been exceeded.

- Speed monitors PNOZ ms1p, PNOZ ms2p with proximity switch > V 2.0

Bit	Message	Comment
2	Speed monitor is ready for reset.	Manual or monitored reset was configured. The reset button has not yet been operated.
3	Cannot monitor the speed because no speed has been selected.	With inputs n1 to n8, standstill or speed monitoring is initialised by a “1” signal. Only one input may occupy the "1" signal status.
8	Selected speed has been exceeded.	The speed at one of the active inputs n1 to n8 has been exceeded.
9	No signal from the proximity switches.	
10	The proximity switches are measuring different speeds.	The bit is set if the speed differential exceeds the configured standstill frequency.

- Speed monitors PNOZ ms1p, PNOZ ms2p with incremental encoder > V 2.0

Bit	Message	Comment
2	Speed monitor is ready for reset.	Manual or monitored reset was configured. The reset button has not yet been operated.
3	Cannot monitor the speed because no speed has been selected.	With inputs n1 to n8, standstill or speed monitoring is initialised by a “1” signal. Only one input may occupy the "1" signal status.
8	Selected speed has been exceeded.	The speed at one of the active inputs n1 to n8 has been exceeded.
9	No signal from the incremental encoder.	
10	Different speeds have been measured for track A and track B.	The bit is set if the speed differential exceeds the configured standstill frequency.
11	Cannot determine the direction of rotation.	The speed monitor has detected a different direction of rotation on tracks A and B

- Speed monitors PNOZ ms1p, PNOZ ms2p with proximity switch and incremental encoder on one axis > V 2.0

Bit	Message	Comment
2	Speed monitor is ready for reset.	Manual or monitored reset was configured. The reset button has not yet been operated.
3	Cannot monitor the speed because no speed has been selected.	With inputs n1 to n8, standstill or speed monitoring is initialised by a “1” signal. Only one input may occupy the "1" signal status.
8	Selected speed has been exceeded.	The speed at one of the active inputs n1 to n8 has been exceeded.
9	No signal from the incremental encoder.	
10	Different speeds have been measured for track A and track B.	The bit is set if the speed differential exceeds the configured standstill frequency.
11	Cannot determine the direction of rotation.	The speed monitor has detected a different direction of rotation on tracks A and B

Bit	Message	Comment
12	The incremental encoder is reporting standstill and the proximity switch is reporting movement.	The mechanical connection between incremental encoder and shaft has been broken.
13	The incremental encoder is reporting movement and the proximity switch is reporting standstill.	The speed monitor has detected a different direction of rotation on tracks A and B

- Speed monitor PNOZ ms3p

Bit	Message	Comment
2	Speed monitor is ready for reset.	Manual or monitored reset was configured. The reset button has not yet been operated.
3	Cannot monitor the speed because no speed has been selected.	With inputs n1 to n8, standstill or speed monitoring is initialised by a “1” signal. Only one input may occupy the "1" signal status.
8	Selected speed has been exceeded.	The speed at one of the active inputs n1 to n8 has been exceeded.
9	No signal from the incremental encoder.	
10	Unfeasible or single-channel signal from the incremental encoder	
11	Cannot determine the direction of rotation.	The speed monitor has detected a different direction of rotation on tracks A and B
14	Speed monitoring is deactivated.	Deactivation input is configured and =1

- Speed monitor PNOZ ms4p

Bit	Message	Comment
2	Speed monitor is ready for reset.	Manual or monitored reset was configured. The reset button has not yet been operated.
3	New speed must be accepted	
8	Selected speed has been exceeded.	The speed at one of the active inputs n1 to n8 has been exceeded.
9	No signal from the incremental encoder.	
10	Unfeasible or single-channel signal from the incremental encoder	
11	Cannot determine the direction of rotation.	The speed monitor has detected a different direction of rotation on tracks A and B
14	Speed monitoring is deactivated.	Deactivation input is configured and =1

- Sequential muting, parallel muting, cross muting

Bit/ DW	Message	Comment
DW = 0	Enable triggered	
Bit 0	Optical safety device was triggered, although muting is inactive.	Lght curtain interrupted (without active muting), resets muting after error or starts muting
Bit 2	Safety device is ready for reset.	Waiting for reset (reset)
Bit 3	There is an object in the muting zone or the optical safety device is faulty.	Sensor status implausible, override required
Bit 8	Cannot switch on because the start enable (“EN2”) has not been triggered.	Muting time exceeded, only one sensor operated
Bit 9	Cannot switch on because the static enable (“EN1”) has not been triggered.	Feasibility error, muting sensors 1 and 2
Bit 10	Press was stopped because the static enable (“EN1”) is missing.	Feasibility error, muting sensors 3 and 4, not in the case of cross muting

- Group diagnostic message

Bit/ DW	Message	Comment
Bit 1	Stored status of the first configured diagnostic bit available to the OR connection	
Bit 2	Stored status of the second configured diagnostic bit available to the OR connection	
Bit 3	Stored status of the third configured diagnostic bit available to the OR connection	
Bit 4	Stored status of the fourth configured diagnostic bit available to the OR connection	
Bit 5	Stored status of the fifth configured diagnostic bit available to the OR connection	

- Press element: Run monitoring

Bit	Message	Comment
2	Run monitoring is ready for reset.	Apply 1/0 pulse edge at Reset input parameter.
8	Start-up time was exceeded.	The set start-up time has elapsed.
9	Shaft is broken	- The camshaft is no longer mechanically connected to the shaft - Open circuit in the encoder circuit

- Press element: Monitoring of rotary cam arrangement

Bit	Message	Comment
2	Monitoring of the rotary cam arrangement is ready for reset.	1/0 pulse edge at reset input parameter
3	Overrun was exceeded.	
8	The run-up cam failed to switch off when the overrun cam was switched off.	NL: Overrun cam, HL: Run-up cam Plausibility error 1: NL = 1/0 pulse edge and HL = 1
9	The run-up cam failed to switch on when the overrun cam was switched on.	Plausibility error 2: NL = 0/1 pulse edge and HL = 0
10	The overrun cam failed to switch off when the run-up cam was switched on.	Plausibility error 3: HL = 0/1 pulse edge and NL = 1
10	The overrun cam failed to switch on when the run-up cam was switched off.	Plausibility error 4: HL = 1/0 pulse edge and NL = 0

- Press element: Set-up mode

Bit/ DW	Message	Comment
DW = 0	“Set-up” mode has been enabled	
Bit 0	"Set-up" mode is not active.	Enable has not been triggered, input parameter *MODE* = 0
Bit 2	Press is ready for reset.	1/0 pulse edge at reset input parameter
Bit 8	Cannot switch on because the start enable (“EN2”) has not been triggered.	No enable because start enable *EN2* = 0
Bit 9	Cannot switch on because the static enable (“EN1”) has not been triggered.	No enable because static enable EN1 = 0
Bit 11	Press was stopped because the static enable (“EN1”) is missing.	No enable because static enable EN1 = 0 during operation

- Press element: Single-stroke

Bit/ DW	Message	Comment
DW = 0	“Single-stroke” mode has been enabled	
Bit 0	"Single stroke" mode is not active.	Enable has not been triggered, input parameter *MODE* = 0
Bit 2	Press is ready for reset.	1/0 pulse edge at *Reset* input parameter
Bit 8	Cannot switch on because the start enable (“EN2”) has not been triggered.	No enable because start enable *EN2* = 0
Bit 9	Cannot switch on because the static enable (“EN1”) has not been triggered.	No enable because static enable *EN1* = 0

Bit/ DW	Message	Comment
Bit 10	Cannot switch on because the safety enable (“EN3”) has not been triggered.	No enable because there is no safety enable *EN3* = 0
Bit 11	Press was stopped because the static enable (“EN1”) is missing.	No enable because static enable EN1 = 0 during operation
Bit 12	Safety enable (“EN3”) is missing.	No enable because safety enable *EN3* = 0 during operation

- Press element: Automatic mode

Bit/ DW	Message	Comment
DW = 0	“Automatic” mode has been enabled	
Bit 0	"Automatic" mode is not active.	Enable has not been triggered, input parameter *MODE* = 0
Bit 2	Press is ready for reset.	1/0 pulse edge at *Reset* input parameter
Bit 8	Cannot switch on because the start enable (“EN2”) has not been triggered.	No enable because start enable *EN2* = 0
Bit 9	Cannot switch on because the static enable (“EN1”) has not been triggered.	No enable because static enable *EN1* = 0
Bit 11	Press was stopped because the static enable (“EN1”) is missing.	No enable because static enable *EN1* = 0 during operation
Bit 13	Cannot switch on because the stop button has been operated.	No enable because input parameter *STOP* = 0

- Press element: Light curtain

Bit/ DW	Message	Comment
DW = 0	“Break” mode has been enabled	
Bit 0	"Break" mode is inactive.	Enable has not been triggered, input parameter *MODE* = 0
Bit 2	Light curtain is ready for break mode.	Break mode is active, waiting for break
Bit 8	Enable must be performed.	1/0 pulse edge at *Reset* input parameter, wait for reset

- Burner part 1

Bit	Message	Comment
2	Burner is ready for reset.	
4	Stop (Signal=1 during start-up test)	
5	Reset (Signal=1 during start-up test)	
6	Safety chain 1 broken (CHA1)	

Bit	Message	Comment
7	Safety chain 2 broken (CHA2)	
8	Ignition and operation safety chain broken (CHAI)	
9	Air pressure error (AIRP)	
10	Flame error, main flame (FLAM)	
11	Flame error, ignition flame (FLAI)	
12	Error during compound controller to pre-purge position (PUR)	
13	Error during compound controller to ignition position (IGNI)	
14	Tightness control error	

- Burner part 2

Bit	Message	Comment
0	Step 0 enabled	Step 0: Burner switched off
1	Step 1 enabled	Step 1: Check start conditions
2	Step 2 enabled	Step 2: Start-up of combustion air blower
3	Step 3 enabled	Step 3: Compound controller to pre-purge position
4	Step 4 enabled	Step 4: Only relevant internally
5	Step 5 enabled	Step 5: Pre-purge/tightness control: Vent
6	Step 6 enabled	Step 6: Pre-purge/tightness control: Test air pressure
7	Step 7 enabled	Step 7: Pre-purge/tightness control: Fill
8	Step 8 enabled	Step 8: Pre-purge/tightness control: Test fuel pressure
9	Step 9 enabled	Step 9: Continue prepurge
10	Step 10 enabled	Step 10: Compound controller to ignition position
11	Step 11 enabled	Step 11: Only relevant internally
12	Step 12 enabled	Step 12: Pre-ignition
13	Step 13 enabled	Step 13: Ignite ignition flame/1st safety time
14	Step 14 enabled	Step 14: Stabilise ignition flame
15	Step 15 enabled	Step 15: Ignite main flame/2nd safety time

- Burner part 3

Bit	Message	Comment
0	Step 16 enabled	Step 16: Stabilise main flame
1	Step 17 enabled	Step 17: Burner in operation/start position
2	Step 18 enabled	Step 18: Only relevant internally
3	Step 19 enabled	Step 19: Only relevant internally
4	Step 20 enabled	Step 20: Afterburn
5	Step 21 enabled	Step 21: Post-purge
6	Step 22 enabled	Step 22: Run down combustion air blower
7	Step 23 enabled	Step 23: Only relevant internally
8	Step 24 enabled	Step 24: Tightness control, vent
9	Step 25 enabled	Step 25: Tightness control, test air pressure
10	Step 26 enabled	Step 26: Tightness control, filling
11	Step 27 enabled	Step 27: Tightness control, test fuel pressure
12	Step 28 enabled	Step 28: Only relevant internally
13	Step 29 enabled	Step 29: Only relevant internally
14	Step 30 enabled	Step 30: Only relevant internally
15	Step 31 enabled	Step 31: Only relevant internally

- Safe Ethernet connection

Bit	Message	Comment
1	Data corrution detected when data is received.	Receive channel
2	Connection failure or timeout during connection monitoring	Receive channel
3	Address conflict detected when data is received	Receive channel
7	No data received on the communication partner	Send channel

8.5.4 Output elements

- Output elements with feedback loop

Bit/ DW	Message	Comment
DW = 0	Enable triggered	
Bit 8	Feedback loop monitoring is registering an error.	- When the output was switched on, the feedback loop was not closed (= 1). - After the output was switched on, the feedback loop was not opened within 3 s (= 0)

- Safety valve

Bit/ DW	Message	Comment
Bit 0	Valve is not activated.	
Bit 2	Valve is ready for reset.	Reset error messages on the reset input
Bit 8	Cannot switch on because, according to the feedback loop, the valve is already switched on.	Start attempt for open feedback loop
Bit 11	When the valve was switched on, the feedback loop failed to open or opened too late.	Power-up monitoring time TOn exceeded, feedback loop did not open during TOn
Bit 12	When the valve was switched off, the feedback loop failed to close or closed too late.	Switch off monitoring time T0ff exceeded, feedback loop did not close during TOff
Bit 13	Error caused by valve or feedback loop	Feedback loop closes when the valve is activated

9 Appendix

9.1 Table assignment

There are a total of 10 tables, with the following contents:

Table 1:	Configuration
Table 2:	Reserved
Table 3:	State of inputs
Table 4:	State of outputs
Table 5:	Status of LED
Table 6:	Reserved
Table 7:	Diagnostic word
Table 8:	Element types
Table 9:	Transfer/state of the expanded virtual inputs and outputs
Table 10	State of the virtual inputs and outputs on the integrated link interface on the PNOZ mm0.2p
Table 11	State of the safe inputs and outputs on the safe Ethernet connection
Element types	The element type's byte is entered in Table 8

The content of the tables is described in detail in the Appendix.

9.2 Table 1

Table 1 consists of 9 segments, each of which has 13 Bytes. It contains device data from the base unit and the project data defined in the PNOZmulti Configurator.

Segment	Byte	Content	Example/Comment
0	0	Product number (hex)	Product number 733 100: 000BCBEC hex Byte 0: 00, Byte 1: 0B, Byte 2: CB, Byte 3: EC
	1		
	2		
	3		
	4	Unit version (hex)	Unit version 20: 14 hex Byte 4: 00, Byte 5, Byte 6: 00, Byte 7: 14
	5		
	6		
	7		
	8	Serial number (hex)	Serial number 123 456: 0001E240 hex. Byte 8: 00, Byte 9: 01, Byte 10: E2, Byte 11: 40
	9		
	10		
	11		
	12	Free	

Segment	Byte	Content	Example/Comment
1	0	Check sum safe (hex)	Check sum A1B2 hex: Byte 0: A1, Byte 1: B2
	1		
	2	Overall project check sum (hex)	Check sum 3C5A hex: Byte 2: 3C, Byte 3: 5A
	3		
	4	Project creation date	Creation date: 28.11.2003 Byte 4: 1C, Byte 5: 0B, Byte 6: 07, Byte 7: D3
	5		
	6		
	7		
	8	Operating hours counter (hex)	Byte 8: x 10000 hex Byte 9: x 100 hex Byte 10: x 1 hex Operating hours: 106786 Byte 8: 01, Byte 9: A1, Byte 10: 22
	9		
	10		
	11	Type of base unit (hex)	PNOZ m1p: 00 PNOZ m0p: 02 PNOZ m2p: 04 PNOZ m3p: 03 PNOZ m1p ETH: 20 PNOZ m0p ETH: 22 PNOZ m2p ETH: 24 PNOZ m3p ETH: 23 PNOZ mm0p: 50 PNOZ mm0.1p: 51 PNOZ mm0.2p: 52
	12	Free	Free

Segment	Byte	Content	Example/Comment
2	0	Configuration, fieldbus module / Integrated interface	Byte 0 contains the Hex code for a fieldbus module (installed on the left) or for inputs and outputs via the integr. interface: Fieldbus module PNOZ mc / PNOZ mmc ... :30 Communication module PNOZ mmc1p: 02 Communication module PNOZ mmc2p: 01 Communication module PNOZ mmc1p plus fieldbus module: 32 Communication module PNOZ mmc2p plus fieldbus module: 31 No fieldbus module and no communication module: FF Virtual inputs and outputs via integr. interface: 40 Communication module PNOZ mmc1p and virtual inputs and outputs via integr. interface: 42 Communication module PNOZ mmc2p and virtual inputs and outputs via integr. interface: 41 Additional input modules on the left: PNOZml1p: See Table 1, Segment 8
	1	Configuration, 1st expansion module, right	Byte 1 ... 8 contains the Hex code for the expansion modules on the right: PNOZ mi1p: 08 PNOZ mi2p: 38 PNOZ mo1p: 18 PNOZ mo2p: 10 PNOZ mo3p: 30 PNOZ mo4p: 28 PNOZ mo5p: 48 PNOZ mc1p: 20 PNOZ ms3p: 68 PNOZ ms4p: 78 PNOZ ms1p/PNOZ ms2p: 88 PNOZ ms2p HTL: 58 PNOZ ms3p HTL: 64 PNOZsigma with one output: 11 PNOZsigma with two outputs: 22 No expansion module: 00
	2	Configuration, 2nd expansion module, right	
	3	Configuration, 3rd expansion module, right	
	4	Configuration, 4th expansion module, right	
	5	Configuration, 5th expansion module, right	
	6	Configuration, 6th expansion module, right	
	7	Configuration, 7th expansion module, right	
	8	Configuration, 8th expansion module right	
	9	Free	
	10	Free	
	11	Free	
	12	Free	

Segment	Byte	Content	Example/Comment
3	0	1st character	Byte 0 ... 12 of the project name, which was defined in the PNOZmulti Configurator under "Enter project data"; this is stored in UNICODE format. 2 Bytes contain the Hex code of the individual UNI-CODE characters
	1		
	2	2nd character	
	3		
	4	3rd character	
	5		
	6	4th character	
	7		
	8	5th character	
	9		
	10	6th character	
	11		
	12	7th character (High Byte)	
4	0	7th character (Low Byte)	Project name Byte 13 ... 25
	1	8th character	
	2		
	3	9th character	
	4		
	5	10th character	
	6		
	7	11th character	
	8		
	9	12th character	
	10		
	11	13th character	
	12		

Segment	Byte	Content	Example/Comment
5	0	14th character	Project name Byte 26 ... 31
	1		
	2	15th character	
	3		
	4	16th character	
	5		
	6	End character FF	The end of the character string is signalled with "FFFF".
	7	End character FF	
	8	Free	
	9	Free	
	10	Free	
	11	Free	
	12	Free	
6	0	Day	Date on which the program on the chip card was last modified Date modified : 28.11.2003 Byte 4: 1C, Byte 5: 0B, Byte 6: 07, Byte 7: D3 Time: 14 hours 25 minutes Byte 4: 0E, Byte 5: 19 Time zone 1: Byte 6: 01
	1	Month	
	2	Year	
	3		
	4	Hour	
	5	Minute	
	6	Time zone	
	7	Reserved	
	8	Reserved	
	9	Reserved	
	10	Reserved	
	11	Reserved	
	12	Reserved	

Segment	Byte	Content	Example/Comment
7	0	Fieldbus type	Profibus: 0x0001 Interbus: 0x0010 Interbus 2M: 0x0011 DeviceNet: 0x0025 CanOpen: 0x0020 Ethernet IP/Modbus TCP: 0x0083 PROFINET: 0x0084 CC Link: 0x0090 EtherCAT: 0x0087 Sercos III: 0x0095 Powerlink: 0x0098
	1		
	2	Software version	5 Bits for version, 3 Bits for sub-number e.g.: Version:1.2 Byte 2: 0 0 0 0 1 0 1 0
	3	Reserved	
	...		
	12		
8	0	Configuration, 1st expansion module, left	Byte 0 ... 5 contains the Hex code for the expansion modules to the left of the base unit. Fieldbus modules are not considered in this segment (see Table 1, Segment 2). PNOZ ml1p: A8 PNOZ ml2p: C8 PNOZ ma1p: B8
	1	Configuration, 2nd expansion module, left	
	2	Configuration, 3rd expansion module, left	
	3	Configuration, 4th expansion module, left	
	4	Configuration, 5th expansion module, left	
	5	Configuration, 6th expansion module left	
	6	Free	
	...		
	12		

9.3 Table 3

Table 3 consists of 3 segments, each of which has 13 Bytes. It contains the status of the inputs

Information

On the base units PNOZmulti Mini, the status of the configurable inputs/outputs is only displayed if they are configured as inputs in the PNOZmulti Configurator.

Segment	Byte	Content	Example/Comment									
0	0	I0 ... I7 base unit, IM0 ... I7 base unit Mini	For example: The safety system consists of one base unit PNOZ m1p and one expansion module PNOZ mi1p									
	1	I8 ... I15 base unit I8 ... I15 base unit Mini										
	2	I16 ... I19 base unit IM16 ... IM19 base unit Mini	Byte 0	I7	I6	I5	I4	I3	I2	I1	I0	PNOZ m1p
			Byte 1	I15	I14	I13	I12	I11	I10	I9	I8	PNOZ m1p
	3	0	Byte 2	0	0	0	0	I19	I18	I17	I16	PNOZ m1p
	4	0	Byte 3	0	0	0	0	0	0	0	0	
	5	I0 ... I7 1st expansion module right	Byte 4	0	0	0	0	0	0	0	0	
0	6	I0 ... I7 2nd expansion module right	Byte 5	I7	I6	I5	I4	I3	I2	I1	I0	PNOZ mi1p
0	7	I0 ... I7 3rd expansion module right	If an input has a high signal, the corresponding bit will be "1"; if an input has a low signal, the bit will be "0". Virtual inputs on the 2nd link module PNOZ ml1p									
	8	I0 ... I7 4th expansion module right										
	9	I0 ... I7 5th expansion module right										
	10	I0 ... I7 6th expansion module right										
	11	I0 ... I7 7th expansion module right										
	12	I0 ... I7 8th expansion module right										
1	0	I0 ... I7 1st expansion module, left										
	1	I8 ... I15 1st expansion module, left										
	2	I16 ... I23 1st expansion module, left										

Segment	Byte	Content	Example/Comment									
1	3	I24 ... I31 1st expansion module, left	Byte 4	I7	I6	I5	I4	I3	I2	I1	I0	
	4	I0 ... I7 2nd expansion module, left	Byte 5	I15	I14	I13	I12	I11	I10	I9	I8	
	5	I8 ... I15 2nd expansion module, left	Byte 6	I23	I22	I21	I20	I19	I18	I17	I16	
	6	I16 ... I23 2nd expansion module, left	Byte 7	I31	I30	I29	I28	I27	I26	I25	I24	
	7	I24 ... I31 2nd expansion module, left										
	8	I0 ... I7 3rd expansion module, left	If an input has a high signal, the corresponding bit will be "1"; if an input has a low signal, the bit will be "0". Analogue inputs on the analogue input module PNOZ ma1p: Byte 0: Analogue Input 0 Analogue Value High Byte Byte 1: Analogue Input 0 Analogue Value Low Byte Byte 2: Analogue Input 1 Analogue Value High Byte Byte 3: Analogue Input 1 Analogue Value Low Byte Bytes 0 and 1 are to be interpreted as words and will be displayed as a scaled value. A distinction is made between voltage and current measurement. The following applies when current is measured: 1 Bit = 6.25 µA e.g.: Byte 0 = 0x01; Byte 1 = 0xff -> 0x01ff*6.25 µA = 3.19 mA The following applies for voltage measurement: 1 Bit = 2.5 mV Please note: With voltage measurement, negative values are also valid. The negative value is formed via the two's complement. e.g.: Byte 0 = 0x01; Byte 1 = 0xff -> 0x01ff * 2.5 mV = 1.28 V e.g.: Byte 0 = 0x0F8; Byte 1 = 0x30 -> 0xF830 = -5 V Assignment of Bytes on the **base units PNOZmulti Mini:**
	9	I8 ... I15 3rd expansion module, left										
	10	I16 ... I23 3rd expansion module, left										
	11	I24 ... I31 3rd expansion module, left										
	12	Free										
2	0	I0 ... I7 4th expansion module, left										
	1	I8 ... I15 4th expansion module, left										
	2	I16 ... I23 4th expansion module, left										
	3	I24 ... I31 4th expansion module, left										
	4	I0 ... I7 5th expansion module, left										
	5	I8 ... I15 5th expansion module, left										
	6	I16 ... I23 5th expansion module, left										
	7	I24 ... I31 5th expansion module, left										
	8	I0 ... I7 6th expansion module, left										
	9	I8 ... I15 6th expansion module, left										
	10	I16 ... I23 6th expansion module, left										
	11	I24 ... I31 6th expansion module, left										
	12	Free										

Segment	Byte	Content	Example/Comment								
			Byte 0	I7	I6	I5	I4	IM3	IM2	IM1	IM0	PNOZ mmxp
			Byte 1	I15	I14	I13	I12	I11	I10	I9	I8	PNOZ mmxp
			Byte 2	0	0	0	0	IM 19	IM 18	IM 17	IM 16	PNOZ mmxp

9.4 Table 4

Table 4 consists of 4 segments, each of which has 13 Bytes. It contains the status of the outputs

Information

On the base units PNOZmulti Mini, the status of the configurable inputs/outputs is only displayed if they are configured as outputs in the PNOZmulti Configurator.

Segment	Byte	Content	Example/Comment							
0	0	IM0 ... IM3 base unit PNOZmulti Mini	Assignment of Bytes depends on the unit:							
	1	0	**Base units PNOZmulti Mini**							
	2	IM16 ... T3M23 base unit PNOZmulti Mini	Segment 0, Byte 0:							
	3	O0 ... O3 base unit PNOZmulti	0	0	0	0	IM3	IM2	IM1	IM0
	4	O4 ... O5 base unit PNOZmulti	Segment 0, Byte 2:							
	5	O0 ... O7 1st expansion module, right	T3 M23	T2 M22	T1 M21	T0 M20	IM19	IM18	IM17	IM16 ...
	6	O0 ... O7 2nd expansion module, right	**Base units PNOZmulti** Segment 0, Byte 3:							
	7	O0 ... O7 3rd expansion module, right								
	8	O0 ... O7 4th expansion module, right	0	0	1	1	O3	O2	O1	O0
	9	O0 ... O7 5th expansion module, right	Segment 0, Byte 4:							
	10	O0 ... O7 6th expansion module, right	0	0	0	0	0	0	O5	O4
	11	O0 ... O7 7th expansion module, right	**PNOZ mo1p** Segment 0, Byte 5 ... 12:							
	12	O0 ... O7 8th expansion module, right								

Segment	Byte	Content	Example/Comment							
1	0	0	0	0	0	0	O3	O2	O1	O0
	1	0	Segment 1, Byte 5 ... 12:							
	2	0	0	0	0	0	0	0	0	0
	3	0	**PNOZ mo2p, PNOZ mo3p**							
	4	0	Segment 0, Byte 5 ... 12:							
	5	O8 ... O15 1st expansion module, right	0	0	0	0	0	0	O1	O0
	6	O8 ... O15 2nd expansion module, right	Segment 1, Byte 5 ... 12							
	7	O8 ... O15 3rd expansion module, right	0	0	0	0	0	0	0	0
	8	O8 ... O15 4th expansion module, right	**PNOZ mo4p, PNOZ mo5p** Segment 0, Byte 5 ... 12:							
	9	O8 ... O15 5th expansion module, right								
	10	O8 ... O15 6th expansion module, right	0	0	0	0	O3	O2	O1	O0
	11	O8 ... O15 7th expansion module, right	Segment 1, Byte 5 ... 12							
	12	O8 ... O15 8th expansion module, right	0	0	0	0	0	0	0	0
			PNOZ mc1p Segment 0, Byte 5 ... 12:							
			A7	A6	A5	A4	A3	A2	A1	A0
			Segment 1, Byte 5 ... 12:							
			A15	A14	A13	A12	A11	A10	A9	A8
			If an output has a high signal, the corresponding Bit will contain "1"; if the output is open (low signal), the Bit will contain "0".							

Segment	Byte	Content	Example/Comment							
2	0	O0 ... O7 1st expansion module, left	**PNOZ mc1p** Segment 0, Byte 5 ... 12:							
	1	O8 ... O15 1st expansion module, left	A7	A6	A5	A4	A3	A2	A1	A0
	2	O16 ... O23 1st expansion module, left	Segment 1, Byte 5 ... 12:							
	3	O24 ... O31 1st expansion module, left	A15	A14	A13	A12	A11	A10	A9	A8
	4	O0 ... O7 2nd expansion module, left	If an output has a high signal, the corresponding Bit will contain "1"; if the output is open (low signal), the Bit will contain "0".							
	5	O8 ... O15 2nd expansion module, left								
	6	O16 ... O23 2nd expansion module, left	**Virtual outputs on the 3rd link module** **PNOZ ml1p**: Segment 2							
	7	O24 ... O31 2nd expansion module, left								
	8	O0 ... O7 3rd expansion module, left								
	9	O8 ... O15 3rd expansion module, left	Byte							
	10	O16 ... O23 3rd expansion module, left	8	O7	O6	O5	O4	O3	O2	O1	O0
	11	O24 ... O31 3rd expansion module, left	9	O15	O14	O13	O12	O11	O10	O9	O8
	12	Free	10	O23	O22	O21	O20	O19	O18	O17	O16

Segment	Byte	Content	Example/Comment								
3	0	O0 ... O7 4th expansion module, left	11	O31	O30	O29	O28	O27	O26	O25	O24
	1	O8 ... O15 4th expansion module, left	If an output has a high signal, the corresponding Bit will contain "1"; if the output is open (low signal), the Bit will contain "0".								
	2	O16 ... O23 4th expansion module, left									
	3	O24 ... O31 4th expansion module, left									
	4	O0 ... O7 5th expansion module, left									
	5	O8 ... O15 5th expansion module, left									
	6	O16 ... O23 5th expansion module, left									
	7	O24 ... O31 5th expansion module, left									
	8	O0 ... O7 6th expansion module, left									
	9	O8 ... O15 6th expansion module, left									
	10	O16 ... O23 6th expansion module, left									
	11	O24 ... O31 6th expansion module, left									
	12	Free									

9.5 Table 5

Table 5 consists of 5 segments. It contains the LED status.

Segment	Byte	Content	Example/Comment
0	0	RUN	Depending on the LED status, the following Hex code will be in Byte 0 ... 12: 00 hex: LED off FF hex: LED on 30 hex: LED flashes
	1	DIAG	
	2	FAULT	
	3	IFAULT	
	4	OFAULT	
	5	FAULT 1: Expansion module, right	
	6	FAULT 2: Expansion module, right	
	7	FAULT 3: Expansion module, right	
	8	FAULT 4: Expansion module, right	
	9	FAULT 5: Expansion module, right	
	10	FAULT 6: Expansion module, right	
	11	FAULT 7: Expansion module, right	
	12	FAULT 8: Expansion module, right	

<table>
<tr><th>Seg-ment</th><th>Byte</th><th>Content</th><th colspan="9">Example/Comment</th></tr>
<tr><td rowspan="13">1</td><td>0</td><td>LED I0 ... I7 base unit</td><td colspan="9">PNOZ mi1p</td></tr>
<tr><td>1</td><td>LED I8 ... I15 base unit</td><td colspan="9">Bytes 5 ... 12</td></tr>
<tr><td>2</td><td>LED I16 ... I19 base unit</td><td>Input</td><td>I7</td><td>I6</td><td>I5</td><td>I4</td><td>I3</td><td>I2</td><td>I1</td><td>I0</td></tr>
<tr><td>3</td><td>0</td><td colspan="9" rowspan="2">Example: The safety system consists of a base unit and one PNOZ mi1p.</td></tr>
<tr><td>4</td><td>0</td></tr>
<tr><td>5</td><td>LED 1: Expansion module, right</td><td>Byte 0</td><td>I7</td><td>I6</td><td>I5</td><td>I4</td><td>I3</td><td>I2</td><td>I1</td><td>I0</td></tr>
<tr><td>6</td><td>LED 2: Expansion module, right</td><td>Byte 1</td><td>I15</td><td>I14</td><td>I13</td><td>I12</td><td>I11</td><td>I10</td><td>I9</td><td>I8</td></tr>
<tr><td>7</td><td>LED 3: Expansion module, right</td><td>Byte 2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>I19</td><td>I18</td><td>I17</td><td>I16</td></tr>
<tr><td>8</td><td>LED 4: Expansion module, right</td><td>Byte 3</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>9</td><td>LED 5: Expansion module, right</td><td>Byte 4</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>10</td><td>LED 6: Expansion module, right</td><td>Byte 5</td><td>I7</td><td>I6</td><td>I5</td><td>I4</td><td>I3</td><td>I2</td><td>I1</td><td>I0</td></tr>
<tr><td>11</td><td>LED 7: Expansion module, right</td><td colspan="9" rowspan="2">If the LED on an input is flashing, the corresponding bit contains "1"; if the LED is not flashing, the bit contains "0".

PNOZ ms1p, PNOZ ms2p from Version 2.0, PNOZm-s3p, PNOZ ms4p
LED axis 1 = "SHAFT 1"
LED axis 2 = "SHAFT 2" (not on PNOZ ms4p)
Byte 5 ... 12</td></tr>
<tr><td>12</td><td>LED 8: Expansion module, right</td></tr>
<tr><td colspan="2" rowspan="7"></td><td></td><td colspan="4">Axis 2</td><td colspan="4">Axis 1</td></tr>
<tr><td>Bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>LED off</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>LED on</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>LED flashes</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>LED flashes briefly</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td colspan="9">The LED functions are described in the relevant speed monitor operating manual.</td></tr>
</table>

<table>
<tr><th>Seg-ment</th><th>Byte</th><th>Content</th><th colspan="9">Example/Comment</th></tr>
<tr><td rowspan="13">2</td><td>0</td><td>LED1: Status of fieldbus module</td><td colspan="9" rowspan="12">Position of LED1 - LED4 of the fieldbus modules PNOZ-multi:
 

Position of LED1 - LED4 of the fieldbus modules PNOZ-multi Mini:
 
</td></tr>
<tr><td>1</td><td>LED2: Status of fieldbus module</td></tr>
<tr><td>2</td><td>LED3: Status of fieldbus module</td></tr>
<tr><td>3</td><td>LED4: Status of fieldbus module</td></tr>
<tr><td>4</td><td>Free</td></tr>
<tr><td>5</td><td>Free</td></tr>
<tr><td>6</td><td>Free</td></tr>
<tr><td>7</td><td>Free</td></tr>
<tr><td>8</td><td>Free</td></tr>
<tr><td>9</td><td>Free</td></tr>
<tr><td>10</td><td>Free</td></tr>
<tr><td>11</td><td>Free</td></tr>
<tr><td>12</td><td>Free</td><td>LED off</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td></td><td></td><td>Green LED</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td></td><td></td><td>Red LED</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td></td><td></td><td colspan="9">The LED functions are described in the relevant fieldbus module operating manual.</td></tr>
</table>

Seg-ment	Byte	Content	Example/Comment								
3	0	Speed monitor 1 Encoder on axis 1	Status of the LEDs on the speed monitors								
	1	Speed monitor 1 Encoder on axis 2	PNOZ ms1p/PNOZ ms2p:								
	2	Speed monitor 2 Encoder on axis 1	I10, I11, I20, I21, X12, X22								
	3	Speed monitor 2 Encoder on axis 2	PNOZ ms3p: X12 and X22								
	4	Speed monitor 3 Encoder on axis 1	PNOZ ms4p: X12								
	5	Speed monitor 3 Encoder on axis 2	Bit	7	6	5	4	3	2	1	0
	6	Speed monitor 4 Encoder on axis 1	Axis 1	0	0	I11	I11	I10	I10	0	X12
	7	Speed monitor 4 Encoder on axis 2	Axis 2	0	0	I21	I21	I20	I20	0	X22
	8	Free	LEDs for proximity switch: I10, I11, I20,I21: If the LED is lit, the corresponding Bit will contain “1”. The proximity switch is energised. LEDs for incremental encoder: X12 and X22: If the LED is lit, the corresponding Bit will contain “1”. The incremental encoder is connected correctly. The LED functions are described in the speed monitor operating manuals.								
	9	Free									
	10	Free									
	11	Free									
	12	Free									
4	0	FAULT 1: Expansion module, left	Depending on the LED status, the following Hex code will be in Byte 0 ... 5: 00 hex: LED off FF hex: LED on 30 hex: LED flashes								
	1	FAULT 2: Expansion module, left									
	2	FAULT 3: Expansion module, left									
	3	FAULT 4: Expansion module, left									
	4	FAULT 5: Expansion module, left									
	5	FAULT 6: Expansion module, left									
	6	Free									
	7	Free									
	8	Free									
	9	Free									
	10	Free									
	11	Free									
	12	Free									

9.6 Table 7

Table 7 consists of 20 segments. It contains information on the elements within the PNOZmulti Configurator and on the diagnostic word.

Segment	Byte	Content	Example/Comment
0	0	Number of elements that can store a status	
	1	Reserved	
	2	Reserved	
	3	Reserved	
	4	Reserved	
	5	Reserved	
	6	Reserved	
	7	Reserved	
	8	Reserved	
	9	Reserved	
	10	Reserved	
	11	Reserved	
	12	Reserved	
1	0	Element ID = 1 ... 8	Each element is assigned an ID in the PNOZmulti Configurator. If the element's output = 0 (no enable), the corresponding bit is set.
	1	Element ID = 9 ... 16	
	2	Element ID = 17 ... 24	
	3	Element ID = 25 ... 32	Element ID
	4	Element ID = 33 ... 40	Byte 0 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1
	5	Element ID = 41 ... 48	Byte 1 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9
	6	Element ID = 49 ... 56	Byte 2 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17
	7	Element ID = 57 ... 64	.....
	8	Element ID = 65 ... 72	Byte 10 | 88 | 87 | 86 | 85 | 84 | 83 | 82 | 81
	9	Element ID = 73 ... 80	Byte 11 | 96 | 95 | 94 | 93 | 92 | 91 | 90 | 89
	10	Element ID = 81 ... 88	Byte 12 | - | - | - | - | 100 | 99 | 98 | 97
	11	Element ID = 89 ... 96	
	12	Element ID = 97 ... 100	

Segment	Byte	Content	Example/Comment
2	0	Reserved	
	1	Reserved	
	2	Reserved	
	3	Reserved	
	4	Reserved	
	5	Reserved	
	6	Reserved	
	7	Reserved	
	8	Reserved	
	9	Reserved	
	10	Reserved	
	11	Reserved	
	12	Reserved	
3	0, 1	Diagnostic word. Element ID = 1	The diagnostic word is displayed in the PNOZmulti Configurator and on the PVIS expanded diagnostics (see Chapter 6, Diagnostic word, and the online help for the PNOZmulti Configurator) Element ID = 1, e.g. diagnostic word of switch type 6 (element type 1C hex):
	2, 3	Diagnostic word. Element ID = 2	
	4, 5	Diagnostic word. Element ID = 3	
	6, 7	Diagnostic word. Element ID = 4	
	8, 9	Diagnostic word. Element ID = 5	
	10, 11	Diagnostic word. Element ID = 6	Byte 0 (High Byte): 0 0 0 0 0 0 0 1
	12	Reserved	Byte 1 (Low Byte): 0 0 0 0 0 0 0 0
4	0, 1	Diagnostic word. Element ID = 7	Message: Wiring error, clock error
	2, 3	Diagnostic word. Element ID = 8	
	4, 5	Diagnostic word. Element ID = 9	
	6, 7	Diagnostic word. Element ID = 10	
	8, 9	Diagnostic word. Element ID = 11	
	10, 11	Diagnostic word. Element ID = 12	
	12	Reserved	

Segment	Byte	Content	Example/Comment
5	0, 1	Diagnostic word. Element ID = 13	
	2, 3	Diagnostic word. Element ID = 14	
	4, 5	Diagnostic word. Element ID = 15	
	6, 7	Diagnostic word. Element ID = 16	
	8, 9	Diagnostic word. Element ID = 17	
	10, 11	Diagnostic word. Element ID = 18	
	12	Reserved	
6	0, 1	Diagnostic word. Element ID = 19	
	2, 3	Diagnostic word. Element ID = 20	
	4, 5	Diagnostic word. Element ID = 21	
	6, 7	Diagnostic word. Element ID = 22	
	8, 9	Diagnostic word. Element ID = 23	
	10, 11	Diagnostic word. Element ID = 24	
	12	Reserved	
7	0, 1	Diagnostic word. Element ID = 25	
	2, 3	Diagnostic word. Element ID = 26	
	4, 5	Diagnostic word. Element ID = 27	
	6, 7	Diagnostic word. Element ID = 28	
	8, 9	Diagnostic word. Element ID = 29	
	10, 11	Diagnostic word. Element ID = 30	
	12	Reserved	

Segment	Byte	Content	Example/Comment
8	0, 1	Diagnostic word. Element ID = 31	
	2, 3	Diagnostic word. Element ID = 32	
	4, 5	Diagnostic word. Element ID = 33	
	6, 7	Diagnostic word. Element ID = 34	
	8, 9	Diagnostic word. Element ID = 35	
	10, 11	Diagnostic word. Element ID = 36	
	12	Reserved	
9	0, 1	Diagnostic word. Element ID = 37	
	2, 3	Diagnostic word. Element ID = 38	
	4, 5	Diagnostic word. Element ID = 39	
	6, 7	Diagnostic word. Element ID = 40	
	8, 9	Diagnostic word. Element ID = 41	
	10, 11	Diagnostic word. Element ID = 42	
	12	Reserved	
10	0, 1	Diagnostic word. Element ID = 43	
	2, 3	Diagnostic word. Element ID = 44	
	4, 5	Diagnostic word. Element ID = 45	
	6, 7	Diagnostic word. Element ID = 46	
	8, 9	Diagnostic word. Element ID = 47	
	10, 11	Diagnostic word. Element ID = 48	
	12	Reserved	

Segment	Byte	Content	Example/Comment
11	0, 1	Diagnostic word. Element ID = 49	
	2, 3	Diagnostic word. Element ID = 50	
	4, 5	Diagnostic word. Element ID = 51	
	6, 7	Diagnostic word. Element ID = 52	
	8, 9	Diagnostic word. Element ID = 53	
	10, 11	Diagnostic word. Element ID = 54	
	12	Reserved	
12	0, 1	Diagnostic word. Element ID = 55	
	2, 3	Diagnostic word. Element ID = 56	
	4, 5	Diagnostic word. Element ID = 57	
	6, 7	Diagnostic word. Element ID = 58	
	8, 9	Diagnostic word. Element ID = 59	
	10, 11	Diagnostic word. Element ID = 60	
	12	Reserved	
13	0, 1	Diagnostic word. Element ID = 61	
	2, 3	Diagnostic word. Element ID = 62	
	4, 5	Diagnostic word. Element ID = 63	
	6, 7	Diagnostic word. Element ID = 64	
	8, 9	Diagnostic word. Element ID = 65	
	10, 11	Diagnostic word. Element ID = 66	
	12	Reserved	

Segment	Byte	Content	Example/Comment
14	0, 1	Diagnostic word. Element ID = 67	
	2, 3	Diagnostic word. Element ID = 68	
	4, 5	Diagnostic word. Element ID = 69	
	6, 7	Diagnostic word. Element ID = 70	
	8, 9	Diagnostic word. Element ID = 71	
	10, 11	Diagnostic word. Element ID = 72	
	12	Reserved	
15	0, 1	Diagnostic word. Element ID = 73	
	2, 3	Diagnostic word. Element ID = 74	
	4, 5	Diagnostic word. Element ID = 75	
	6, 7	Diagnostic word. Element ID = 76	
	8, 9	Diagnostic word. Element ID = 77	
	10, 11	Diagnostic word. Element ID = 78	
	12	Reserved	
16	0, 1	Diagnostic word. Element ID = 79	
	2, 3	Diagnostic word. Element ID = 80	
	4, 5	Diagnostic word. Element ID = 81	
	6, 7	Diagnostic word. Element ID = 82	
	8, 9	Diagnostic word. Element ID = 83	
	10, 11	Diagnostic word. Element ID = 84	
	12	Reserved	

Segment	Byte	Content	Example/Comment
17	0, 1	Diagnostic word. Element ID = 85	
	2, 3	Diagnostic word. Element ID = 86	
	4, 5	Diagnostic word. Element ID = 87	
	6, 7	Diagnostic word. Element ID = 88	
	8, 9	Diagnostic word. Element ID = 89	
	10, 11	Diagnostic word. Element ID = 90	
	12	Reserved	
18	0, 1	Diagnostic word. Element ID = 91	
	2, 3	Diagnostic word. Element ID = 92	
	4, 5	Diagnostic word. Element ID = 93	
	6, 7	Diagnostic word. Element ID = 94	
	8, 9	Diagnostic word. Element ID = 95	
	10, 11	Diagnostic word. Element ID = 96	
	12	Reserved	
19	0, 1	Diagnostic word. Element ID = 97	
	2, 3	Diagnostic word. Element ID = 98	
	4, 5	Diagnostic word. Element ID = 99	
	6, 7	Diagnostic word. Element ID = 100	
	8, 9	Reserved	
	10, 11	Reserved	
	12	Reserved	

9.7 Table 8

Table 8 consists of 8 segments. It contains the element type with the corresponding Element ID. The available element types are listed after this table.

Segment	Byte	Content	Example/Comment
0	0	Element type. Element ID = 1	
	1	Element type. Element ID = 2	
	2	Element type. Element ID = 3	
	3	Element type. Element ID = 4	
	4	Element type. Element ID = 5	
	5	Element type. Element ID = 6	
	6	Element type. Element ID = 7	
	7	Element type. Element ID = 8	
	8	Element type. Element ID = 9	
	9	Element type. Element ID = 10	
	10	Element type. Element ID = 11	
	11	Element type. Element ID = 12	
	12	Element type. Element ID = 13	
1	0	Element type. Element ID = 14	
	1	Element type. Element ID = 15	
	2	Element type. Element ID = 16	
	3	Element type. Element ID = 17	
	4	Element type. Element ID = 18	
	5	Element type. Element ID = 19	
	6	Element type. Element ID = 20	
	7	Element type. Element ID = 21	
	8	Element type. Element ID = 22	
	9	Element type. Element ID = 23	
	10	Element type. Element ID = 24	
	11	Element type. Element ID = 25	
	12	Element type. Element ID = 26	

Segment	Byte	Content	Example/Comment
2	0	Element type. Element ID = 27	
	1	Element type. Element ID = 28	
	2	Element type. Element ID = 29	
	3	Element type. Element ID = 30	
	4	Element type. Element ID = 31	
	5	Element type. Element ID = 32	
	6	Element type. Element ID = 33	
	7	Element type. Element ID = 34	
	8	Element type. Element ID = 35	
	9	Element type. Element ID = 36	
	10	Element type. Element ID = 37	
	11	Element type. Element ID = 38	
	12	Element type. Element ID = 39	
3	0	Element type. Element ID = 40	
	1	Element type. Element ID = 41	
	2	Element type. Element ID = 42	
	3	Element type. Element ID = 43	
	4	Element type. Element ID = 44	
	5	Element type. Element ID = 45	
	6	Element type. Element ID = 46	
	7	Element type. Element ID = 47	
	8	Element type. Element ID = 48	
	9	Element type. Element ID = 49	
	10	Element type. Element ID = 50	
	11	Element type. Element ID = 51	
	12	Element type. Element ID = 52	

Segment	Byte	Content	Example/Comment
4	0	Element type. Element ID = 53	
	1	Element type. Element ID = 54	
	2	Element type. Element ID = 55	
	3	Element type. Element ID = 56	
	4	Element type. Element ID = 57	
	5	Element type. Element ID = 58	
	6	Element type. Element ID = 59	
	7	Element type. Element ID = 60	
	8	Element type. Element ID = 61	
	9	Element type. Element ID = 62	
	10	Element type. Element ID = 63	
	11	Element type. Element ID = 64	
	12	Element type. Element ID = 65	
5	0	Element type. Element ID = 66	
	1	Element type. Element ID = 67	
	2	Element type. Element ID = 68	
	3	Element type. Element ID = 69	
	4	Element type. Element ID = 70	
	5	Element type. Element ID = 71	
	6	Element type. Element ID = 72	
	7	Element type. Element ID = 73	
	8	Element type. Element ID = 74	
	9	Element type. Element ID = 75	
	10	Element type. Element ID = 76	
	11	Element type. Element ID = 77	
	12	Element type. Element ID = 78	

Segment	Byte	Content	Example/Comment
6	0	Element type. Element ID = 79	
	1	Element type. Element ID = 80	
	2	Element type. Element ID = 81	
	3	Element type. Element ID = 82	
	4	Element type. Element ID = 83	
	5	Element type. Element ID = 84	
	6	Element type. Element ID = 85	
	7	Element type. Element ID = 86	
	8	Element type. Element ID = 87	
	9	Element type. Element ID = 88	
	10	Element type. Element ID = 89	
	11	Element type. Element ID = 90	
	12	Element type. Element ID = 91	
7	0	Element type. Element ID = 92	
	1	Element type. Element ID = 93	
	2	Element type. Element ID = 94	
	3	Element type. Element ID = 95	
	4	Element type. Element ID = 96	
	5	Element type. Element ID = 97	
	6	Element type. Element ID = 98	
	7	Element type. Element ID = 99	
	8	Element type. Element ID = 100	
	9	Reserved	
	10	Reserved	
	11	Reserved	
	12	Reserved	

9.8 Table 9

Table 9 consists of 3 segments. It contains the data of the expanded virtual inputs and outputs 24 – 127. Each input is assigned a bit in segment Bytes 0 ... 12 of the input data; each output is assigned a bit in segment Bytes 0... 12 of the output data.

ATTENTION!

The expanded input Bits are only updated when table 9 segment 1 is accessed. In the event of a fieldbus error, input bits i24 ... i127 are frozen!

Table 9 segment 1

The inputs are set and the outputs are uploaded in segment 1. Contrary to the other tables, not only is a request made of the PNOZmulti by the communications parter, but input data is also sent.

Input data

Segment	Byte	Content	Example/explanation
1	0	Inputs i24 – i31	The communications partner sends the expanded virtual inputs to the PNOZmulti.
	1	Inputs i32 – i39	
	2	Inputs i40 – i47	
	3	Inputs i48 – i55	
	4	Inputs i56 – i63	
	5	Inputs i64 – i71	
	6	Inputs i72 – i79	
	7	Inputs i80 – i87	
	8	Inputs i88 – i95	
	9	Inputs i96 – i103	
	10	Inputs i104 – i111	
	11	Inputs i112 – i119	
	12	Inputs i120 – i127	

Output data

Segment	Byte	Content	Example/explanation
1	0	Outputs o24 – o31	The output data contains the values that are uploaded by the PNOZmulti (see chapter entitled "Basics"/Assignment of Byte 4 ... Byte 18 [📖 17]/"Exception table 9, segment 1").
	1	Outputs o32 – o39	
	2	Outputs o40 – o47	
	3	Outputs o48 – o55	
	4	Outputs o56 – o63	
	5	Outputs o64 – o71	
	6	Outputs o72 – o79	
	7	Outputs o80 – o87	
	8	Outputs o88 – o95	
	9	Outputs o96 – o103	
	10	Outputs o104 – o111	
	11	Outputs o112 – o119	
	12	Outputs o120 – o127	

Table 9 segment 2

Table 9 segment 2 includes the state of the expanded outputs.

Segment	Byte	Content	Example/explanation
2	0	Outputs o24 – o31	
	1	Outputs o32 – o39	
	2	Outputs o40 – o47	
	3	Outputs o48 – o55	
	4	Outputs o56 – o63	
	5	Outputs o64 – o71	
	6	Outputs o72 – o79	
	7	Outputs o80 – o87	
	8	Outputs o88 – o95	
	9	Outputs o96 – o103	
	10	Outputs o104 – o111	
	11	Outputs o112 – o119	
	12	Outputs o120 – o127	

Table 9 segment 3

Table 9 segment 3 includes the state of the expanded inputs.

Segment	Byte	Content	Example/explanation
3	0	Inputs i24 – i31	
	1	Inputs i32 – i39	
	2	Inputs i40 – i47	
	3	Inputs i48 – i55	
	4	Inputs i56 – i63	
	5	Inputs i64 – i71	
	6	Inputs i72 – i79	
	7	Inputs i80 – i87	
	8	Inputs i88 – i95	
	9	Inputs i96 – i103	
	10	Inputs i104 – i111	
	11	Inputs i112 – i119	
	12	Inputs i120 – i127	

9.9 Table 10

Table 10 consists of one segment. It contains the status of the virtual inputs and outputs of the integrated interface to link 2 base units on the base unit PNOZ mm0.2p.

Segment	Byte	Content	Example/Comment
1	0	i0 ... i7 link interface	Virtual inputs of the link interface on the PNOZ mm0.2p
	1	i8 ... i15 link interface	
	2	i16 ... i23 link interface	
	3	i24 ... i31 link interface	
	4	o0 ... o7 link interface	Virtual outputs of the link interface on the PNOZ mm0.2p
	5	o8 ... o15 link interface	
	6	o16 ... o23 link interface	
	7	o24 ... o31 link interface	
	8	Reserved	
	9	Reserved	
	10	Reserved	
	11	Reserved	
	12	Reserved	

9.10 Table 11

Table 11 consists of one segment. It contains the status of the safe inputs and outputs on the safe Ethernet connection.

Segment	Byte	Content	Example/explanation
0	0	i0 ... i7 Safe Ethernet connection	Safe inputs on the safe Ethernet connection
	1	i8 ... i15 Safe Ethernet connection	
	2	i16 ... i23 Safe Ethernet connection	
	3	i24 ... i31 Safe Ethernet connection	
	4	i32 … i39 Safe Ethernet connection	
	5	i40 … i47 Safe Ethernet connection	
	6	o0 ... o7 Safe Ethernet connection	Safe outputs on the safe Ethernet connection
	7	o8 ... o15 Safe Ethernet connection	
	8	o16 ... o23 Safe Ethernet connection	
	9	o24 ... o31 Safe Ethernet connection	
	10	o24 ... o31 Safe Ethernet connection	
	11	O32 ... o39 Safe Ethernet connection	
	12	O40 ... o47 Safe Ethernet connection	

9.11 Element types

The available element types are listed below. Details of the element type's byte are given in Table 8.

Element type (Byte)	Element
	Function elements
01	Switch type 1: N/C
02	Switch type 1: N/C, monitored reset
03	Switch type 1: N/C, manual reset
04	Switch type 1: N/C, start-up test
05	Switch type 1: N/C, start-up test, monitored reset
06	Switch type 1: N/C, start-up test, manual reset
07	Switch type 2: N/C, N/O
08	Switch type 2: N/C, N/O, monitored reset
09	Switch type 2: N/C, N/O, manual reset
0A	Switch type 2: N/C, N/O, start-up test
0B	Switch type 2: N/C, N/O, start-up test, monitored reset
0C	Switch type 2: N/C, N/O, start-up test, manual reset
0D	Switch type 3: N/C, N/C
0E	Switch type 3: N/C, N/C, monitored reset
0F	Switch type 3: N/C, N/C, manual reset
10	Switch type 3: N/C, N/C, start-up test
11	Switch type 3: N/C, N/C, start-up test, monitored reset
12	Switch type 3: N/C, N/C, start-up test, manual reset
13	Switch type 4: N/C, N/C, N/O
14	Switch type 4: N/C, N/C, N/O, monitored reset
15	Switch type 4: N/C, N/C, N/O, manual reset
16	Switch type 4: N/C, N/C, N/O, start-up test
17	Switch type 4: N/C, N/C, N/O, start-up test, monitored reset
18	Switch type 4: N/C, N/C, N/O, start-up test, manual reset
19	Switch type 5: N/C, N/C, N/C
1A	Switch type 5: N/C, N/C, N/C, monitored reset
1B	Switch type 5: N/C, N/C, N/C, manual reset
1C	Switch type 6: Two-hand, N/C
1D	Switch type 7: Two-hand, N/O
1E	Operating mode selector switch 1 from 2
1F	Operating mode selector switch 1 from 3
20	Operating mode selector switch 1 from 4
21	Operating mode selector switch 1 from 5

Element type (Byte)	Element
22	Safety mat, with automatic reset
23	Safety mat, with start-up test
24	Safety mat, with reset button
25	Cascading input
26	Switch type 5: N/C, N/C, N/C, start-up test
27	Switch type 5: N/C, N/C, N/C, start-up test, monitored reset
28	Switch type 5: N/C, N/C, N/C, start-up test, manual reset
2A	Link module status PNOZ ml2p
2B	Link module status PNOZ ml1p
2C	Pulse detection
2D	Operating mode selector switch 1 from 6
2E	Operating mode selector switch 1 from 7
2F	Operating mode selector switch 1 from 8
	Output elements
51	Single-pole semiconductor output with feedback loop
53	Single-pole, redundant semiconductor output with feedback loop
55	Single-pole relay output with feedback loop
57	Single-pole, redundant relay output with feedback loop
59	Cascading output
5A	Single valve
5B	Dual valve
5C	Directional valve
5E	Dual-pole semiconductor output with feedback loop
60	Dual-pole, redundant semiconductor output with feedback loop
	Logic elements
80	Muting sensor: Cross muting
81	Muting sensor: Parallel muting
82	Muting sensor: Sequential muting
90	Reset element, manual reset
91	Reset element, monitored reset
92	RS flip-flop
94	Reset element, non-safety-related reset button, manual reset
B1	Press element, set-up mode
B2	Press element, single-stroke
B3	Press element, automatic mode
A9	Burner element
87	Group diagnostic message

Element type (Byte)	Element
95	Reset module
96	Reset module
C0	Analogue input module
E4	RS flip-flop with negation

1001154-EN-13, 2013-05 Printed in Germany

...

In many countries we are represented by our subsidiaries and sales partners.

Please refer to our homepage for further details or contact our headquarters.

Technical support

+49 711 3409-444
support@pilz.com

Pilz GmbH & Co. KG
Felix-Wankel-Straße 2
73760 Ostfildern, Germany
Telephone: +49 711 3409-0
Telefax: +49 711 3409-133
E-Mail: pilz.gmbh@pilz.de
Internet: www.pilz.com