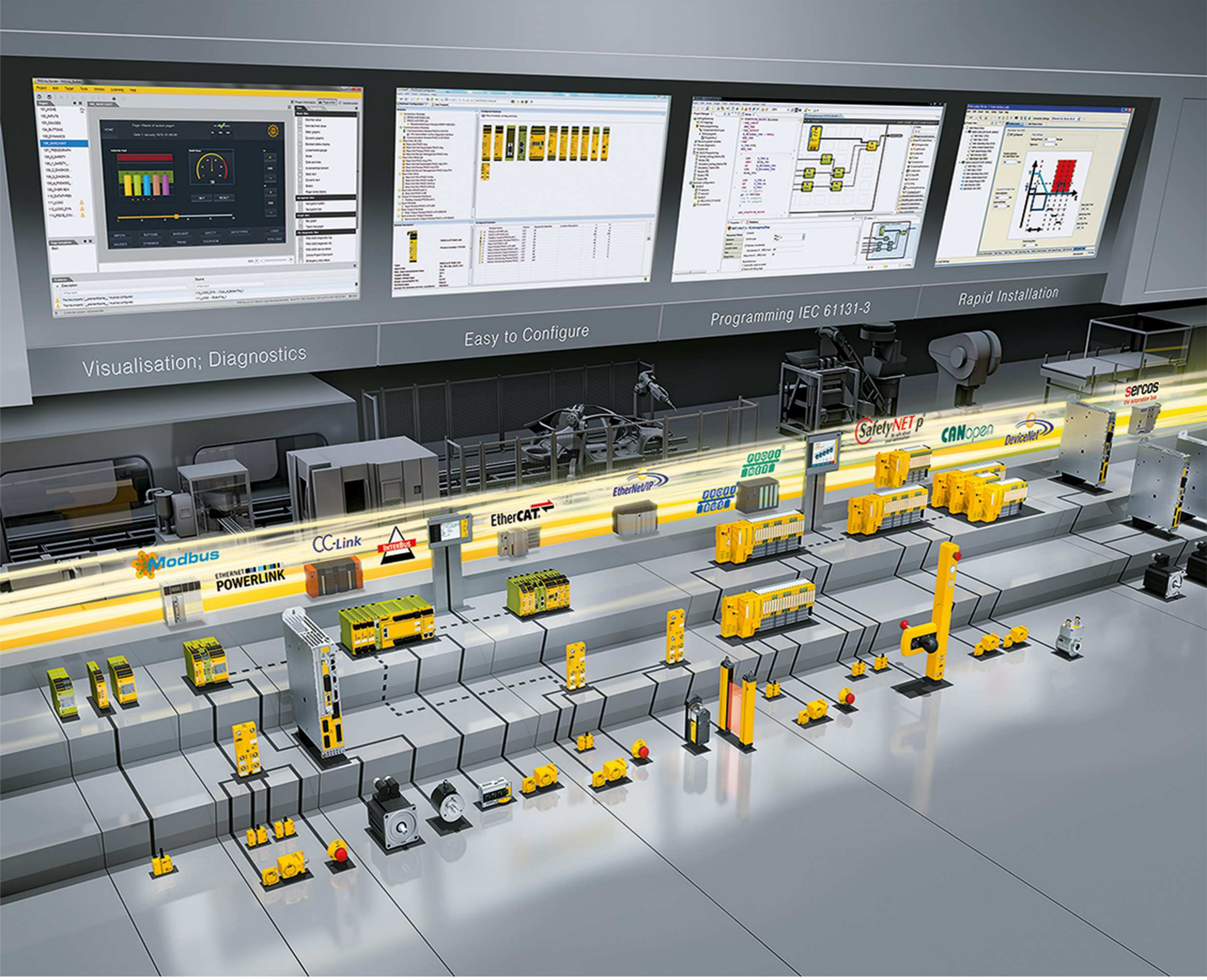


Communication interfaces

PILZ

THE SPIRIT OF SAFETY

- Configurable control systems PNOZmulti 2

Operating Manual 1002971-EN-04

This document is a translation of the original document.

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.

⚠ **CAUTION!**

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.

❗ **NOTICE**

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.

ℹ **INFORMATION**

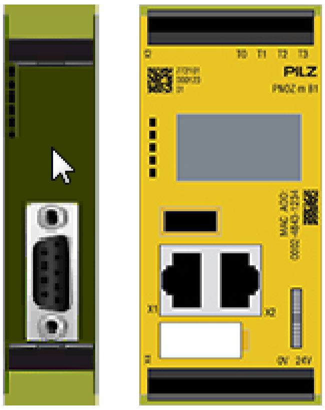
This gives advice on applications and provides information on special features.

2 Overview - Communication options

2.1 Communication via the fieldbus modules

The fieldbus modules provide both cyclical and acyclical data. The chapter entitled Fieldbus Modules [📖 14] describes the structure of the cyclical data. The chapter entitled Service data [📖 73] describes the acyclical data, its structure and addressing.

The following device combinations are possible:

Fieldbus modules		Base units
PNOZ m ES Profibus		PNOZ m B0
PNOZ m ES CANopen		
PNOZ m ES CC-Link		
PNOZ m ES EtherCAT		
PNOZ m ES Powerlink		
PNOZ m ES Ethernet/IP		
PNOZ m ES Profinet		
PNOZ m ES Profibus		PNOZ m B1
PNOZ m ES EtherCAT		

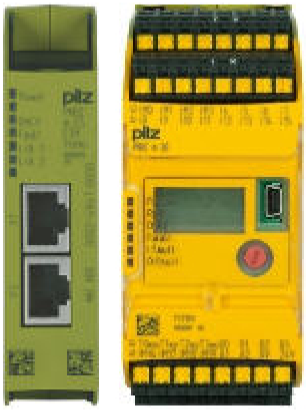
INFORMATION

A maximum of one fieldbus module can be connected to a base unit.

2.2 Communication via the communication modules

When communication takes place via a communication module, data exchange is defined via a Pilz-specific protocol. This protocol is described in more detail in the chapter entitled ETH/RS232 interfaces [📖 41].

The following device combinations are possible:

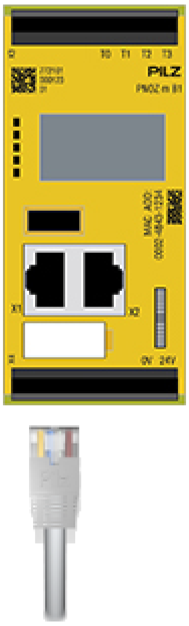
Communication Module		Base units
Communication Module PNOZ m ES RS232		PNOZ m B0
Communication Module PNOZ m ES ETH		

INFORMATION

If virtual inputs/outputs are to be downloaded via the communication module (without using a fieldbus module), 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 the integrated ETH interface

When communication takes place via an integrated ETH interface, data exchange is defined via a Pilz-specific protocol. This protocol is described in more detail in the chapter entitled ETH/RS232 interfaces [📖 41].

Communication		Base units
Integrated Ethernet interface		PNOZ m B1

INFORMATION

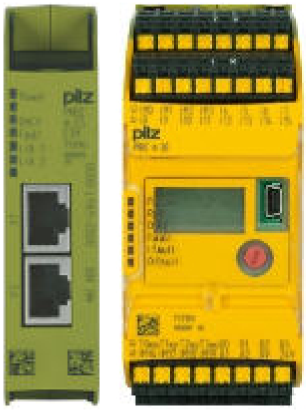
For communication via the integrated Ethernet interface, the interface ***Inputs/outputs that are downloaded via the integrated interface***must be configured in the hardware configuration in the PNOZmulti Configurator.

2.4 Communication via Modbus TCP

For data exchange with Modbus/TCP, the PNOZmulti 2 is the connection server. All the service 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 [📖 59].

The following device combinations are possible:

Communication Module		Base unit
Communication Module PNOZ m ES ETH [📖 41]		PNOZ m B0

INFORMATION

If virtual inputs/outputs are to be downloaded via the communication module (without using a fieldbus module), 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 2 is used to transfer diagnostic data to an application program. This data may only be used for non-safety purposes, e.g. visualisation.

NOTICE

For details of the intended use and application of the configurable control system PNOZmulti 2 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 the product
- Use of the product outside the areas described in this manual
- Use of the product outside the technical details (see chapter entitled "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

- 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 32 Bytes for communication via fieldbuses; this is updated approx. every 15 ms. The Master (PC, PLC) can send 32 Bytes to the PNOZmulti 2 and receive 32 Bytes from the PNOZmulti 2. The Master can process the information in bytes, words or in double words.

4.1.1 Input data (to the PNOZmulti 2)

Double word	Word	Byte	Contents
0	0	0	State of virtual inputs on the PNOZmulti 2 fieldbus module. The inputs are defined in the PNOZmulti Configurator. Each input that is used is given a number there, e.g. i0, i5... The state of input i0 is stored in bit 0 of byte 0; the state of input i5 is stored in bit 5 of byte 0 etc.
		1	
	1	2	
		3	
1	2	4	
		5	
	3	6	
		7	
2	4	8	
		9	
	5	10	
		11	
3	6	12	
		13	
	7	14	
		15	
4	8	16	Reserved
		17	Table number
	9	18	Segment number
		19	Reserved
5	10	20	Reserved
		21	Reserved
	11	22	Reserved
		23	Reserved
6	12	24	Reserved
		25	Reserved
	13	26	Reserved
		27	Reserved

Double word	Word	Byte	Contents
7	14	28	Reserved
		29	Reserved
	15	30	Reserved
		31	Reserved

4.1.2 Output data (from the PNOZmulti 2)

Double Word	Word	Byte	Content
0	0	0	State of virtual outputs on the PNOZmulti 2 fieldbus module. 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.
		1	
	1	2	
		3	
1	2	4	
		5	
	3	6	
		7	
2	4	8	
		9	
	5	10	
		11	
3	6	12	
		13	
	7	14	
		15	
4	8	16	LED status
		17	Table number
	9	18	Segment number
		19	Payload Byte 0
5	10	20	Payload Byte 1
		21	Payload Byte 2
	11	22	Payload Byte 3
		23	Payload Byte 4
6	12	24	Payload Byte 5
		25	Payload Byte 6
	13	26	Payload Byte 7
		27	Payload Byte 8

Double Word	Word	Byte	Content
7	14	28	Payload Byte 9
		29	Payload Byte 10
	15	30	Payload Byte 11
		31	Payload Byte 12

4.1.3 Overview of tables

There are a total of 11 tables, with the following contents:

Table 20:	Process data, base unit
Table 21:	Process data, right-hand expansion modules
Table 22	Process data, left-hand expansion modules
Table 23:	Process data, fieldbus module / communication module
Table 70:	Diagnostic words
Table 71:	Enable elements (only PNOZ m B0)
Table 80:	Project information
Table 90:	Device data, base unit
Table 91:	Device data, right-hand expansion modules
Table 92	Device data, left-hand expansion modules
Table 93:	Device data, fieldbus module / communication module

The content of the tables is described in detail in the chapter entitled Service data [📖 73].

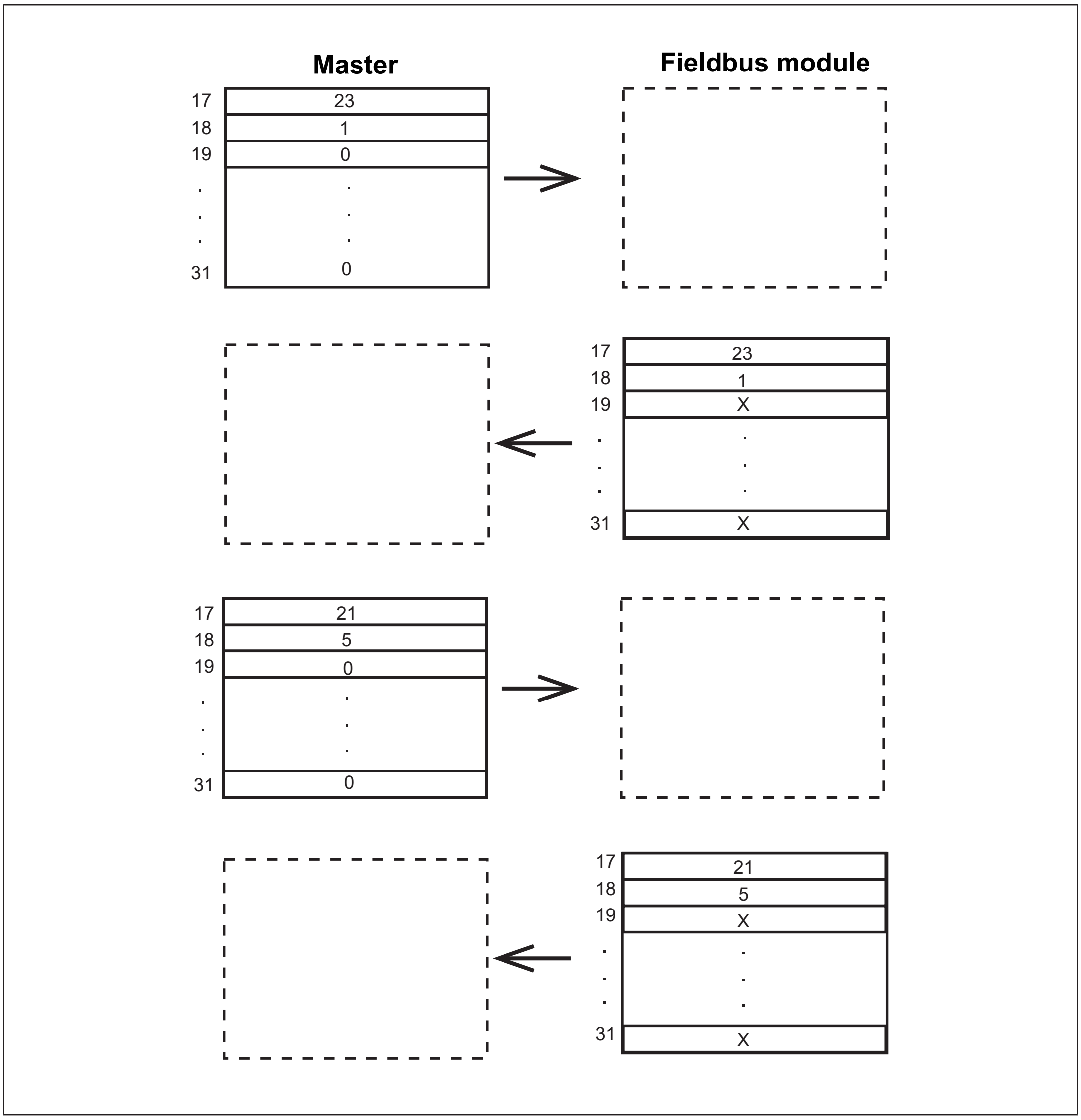
4.1.4 Access to table segments

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 repeats the two numbers and sends the requested data. If data is requested that is not available, the Slave sends the error message FF (hex) or 255 (dec) instead of the segment number. The segments may be requested in any sequence.

4.1.4.1 Example 1 - Access successful

Access to existing segments

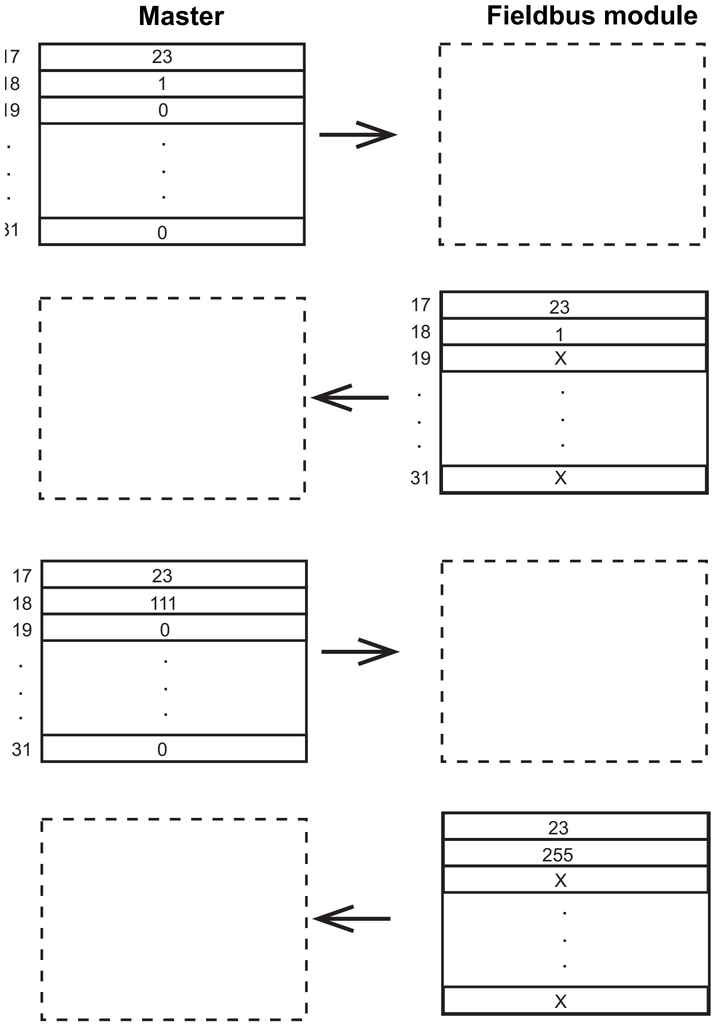
The Master requests segment 1 from table 23. The fieldbus module confirms both these details and sends segment 1. Then the data from segment 5 table 21 is requested.



4.1.4.2 Example 2 - Access failed

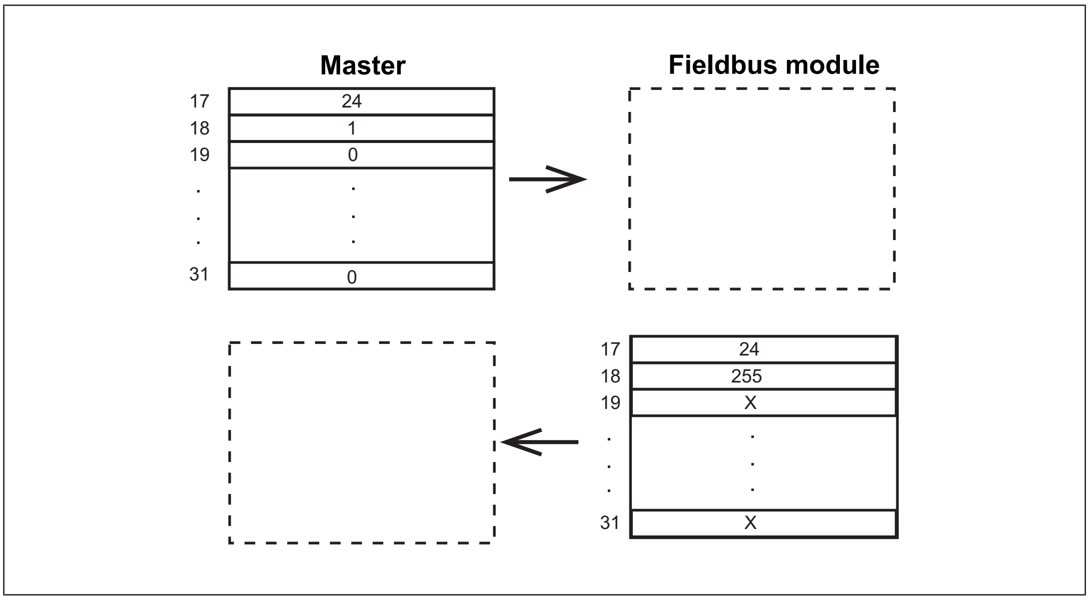
Access to non-existent segment

The Master requests segment 1 from table 23. The fieldbus module confirms both these details and sends segment 1. Then the Master requests segment 111 from table 23. As this table does not contain a segment 111, the Slave registers an error by sending back "255".



Access to non-existent table

The Master requests segment 1 from table 24. As there is no table 24, the Slave registers an error by sending back "255".



4.2 PNOZ m ES Profibus

4.2.1 Cyclical input and output data

Virtual inputs and outputs can be requested or set directly via the following modules. Each element can be selected individually in the master control system, e.g. virtual inputs i0-31. The data width is also established this way.

Input data

The Master writes to the virtual inputs of the PNOZmulti 2.

Description	Input data from PNOZmulti 2
Virtual inputs i0 – i31	4 Input Bytes
Virtual inputs i32 – i63	4 Input Bytes
Virtual inputs i64 – i95	4 Input Bytes
Virtual inputs i96 – i127	4 Input Bytes

Output data

The Master reads the virtual outputs of the PNOZmulti 2.

Description	Output data from PNOZmulti 2
Virtual outputs o0 – o31	4 Output Bytes
Virtual outputs o32 – o63	4 Output Bytes
Virtual outputs o64 – o95	4 Output Bytes
Virtual outputs o96 – o127	4 Output Bytes

4.2.2 Access to table segments

The data in the tables (see Service data [📖 73]) can be requested via the following modules.

Input data

The Master requests a table segment in the form:

Description	Input data from PNOZmulti 2
Table and segment number	2 Input Bytes

Output data

The PNOZmulti 2 replies in the form:

Description	Output data from PNOZmulti 2
Table and segment number + 13 Byte payload	15 Output Bytes

A detailed description is available in the chapter entitled Access to table segments [📖 16].

4.2.3 LED status

Description	Output data from PNOZmulti 2
LED status	1 Output Byte

The LED status of PNOZ m B0 can be requested directly 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

The LED status of PNOZ m B1 can be requested directly 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 FS is lit
- Bit 5: Reserved
- Bit 6 = 1: LED RUN ST is lit
- Bit 7: Reserved

4.2.4 Service data

The service data for PROFIBUS-DP is divided into the following data areas:

- Process data - Base unit and expansion modules
- Process data - Fieldbus module and communication module

- Diagnostic words
- Enable elements (only PNOZ m B0)
- Project data
- Device data

The content of the service data and addressing are described in the chapter entitled Service data [📖 73].

4.2.5 Diagnostic data PNOZ m ES Profibus

These bits contain the diagnostic data of PNOZ m B0

Unit_Diag_ Bit	Contents
00	RUN, base unit is in RUN condition
01	STOP, base unit is in STOP condition
02	Base unit was stopped by the Configurator
03	Reserved
04	External error
05	Internal error
06	External error on the inputs
07	Internal error on the inputs
08	External error on the outputs
09	Internal error on the outputs
10	No connection to base unit
11 … 21	Reserved
22	Error on the base unit
23	Reserved
24	Error on the 1st expansion module, right
25	Error on the 2nd expansion module, right
26	Error on the 3rd expansion module, right
27	Error on the 4th expansion module, right
28	Fault on the 5th expansion module, right
29	Fault on the 6th expansion module, right
30 … 39	Reserved
40	Fault on 1st expansion module, left
41	Fault on 2nd expansion module, left
42	Fault on 3rd expansion module, left

Unit_Diag_ Bit	Contents
43	Fault on 4th expansion module, left
44 ... 45	Reserved
46	Error on the communication module
47 … 55	Reserved
56	Error in the configuration, F_81
57	Error in the user program, F_82
58	Error in the periphery, F_83
59	Error on the speed monitor, F_84
60	Error on the bus module, F_85
61	Error on the link module, F_86
62	Error on the analogue input module, F_87
63	Reserved
64	Error on the input/output module, F_89
65	Error on the Ethernet module, F_8A
66	Internal error on the left-hand expansion module, F_90
67	Internal periphery error, F_93
68	Internal error on the right-hand expansion module, F_94
69	Internal error on the right-hand expansion module, F_95
70	Internal self test error, F_B0
71	Internal data error, F_B1
72	Internal parameter error, F_B2
73	Internal serial/I2C error, F_B3
74	Internal time error, F_B4
75	Internal processor error, F_B5
76	Internal parameter error, F_B6
77	Internal compare error, F_B7
78	Internal sequence error, F_B8
79	Internal periphery error, F_B9
80	Internal error on bus module, F_BA
81	Internal error on link module, F_BB
82	Internal error on speed monitor, F_BC
83	Internal error on analogue input module, F_BD

Unit_Diag_ Bit	Contents
84	Internal error on link module, F_BE
85	Internal error on link module, F_BF
86	Internal error on the left-hand expansion module, F_C0
87	Internal error on the left-hand expansion module, F_C1
88	Internal error on the Ethernet module, F_C2
89	Internal compare error, F_C3
90	Internal error on the right-hand expansion module, F_C4
91	Internal error on the right-hand expansion module, F_C5
92 … 95	Reserved

- Explanation: Bits 56 to 91 display the last content of the error stack (corresponds to the error class). The bits are deleted as soon as PNOZmulti 2 returns to a RUN state.

These bits contain the diagnostic data of PNOZ m B1

Unit_Diag_ Bit	Contents
00	RUN, base unit is in RUN condition
01	STOP, base unit is in STOP condition
02	Reserved
03	Reserved
04	External error
05	Internal error
06	External error on the inputs
07	Reserved
08	External error on the outputs

4.3 PNOZ m ES CANopen

4.3.1 Service Data Objects (SDOs)

All the relevant device parameters and current process data of the CANopen are entered in the object directory of the PNOZmulti 2. These can be read and written via Service Data Objects (SDOs).

To enable the fieldbus module to be incorporated easily within a CANopen network, the object directory is available as an EDS file. The content is described in the chapter entitled Service data [🕮 73].

4.3.2 Service data

The service data for CANopen is divided into the following data areas:

- Process data - Base unit and expansion modules
- Process data - Fieldbus module and communication module
- Diagnostic words
- Enable elements (only PNOZ m B0)
- Project data
- Device data

The content of the service data and addressing are described in the chapter entitled Service data [📖 73].

4.3.3 Process Data Objects (PDOs)

The output data 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		
16	2000	11	TPDO 3	380h + node address
17	2000	12		
18	2000	13		
19	2000	14		
20	2000	15		
21	2000	16		
22	2000	17		
23	2000	18		

Byte	Object Index (hex)	Sub Index (hex)	PDO	COB-ID
24	2000	19	TPDO 4	480h + node address
25	2000	1A		
26	2000	1B		
27	2000	1C		
28	2000	1D		
29	2000	1E		
30	2000	1F		
31	2000	20		

The input data is stored as follows:

Byte	Object Index (hex)	Sub Index (hex)	PDO	COB-ID
0	2100	1	RPDO 1	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	400h + node address
17	2100	12		
18	2100	13		
19	2100	14		
20	2100	15		
21	2100	16		
22	2100	17		
23	2100	18		

Byte	Object Index (hex)	Sub Index (hex)	PDO	COB-ID
24	2100	19	RPDO 4	500h + node address
25	2100	1A		
26	2100	1B		
27	2100	1C		
28	2100	1D		
29	2100	1E		
30	2100	1F		
31	2100	20		

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

4.3.4 Mapping suggestion output data

Index (hex)	Size (byte)	Contents							
		Virtual outputs o0 … o127							
0x2000:01	1	Bit 7	…						Bit 0
		o7	o6	o5	o4	o3	o2	o1	o0
0x2000:02	1	o15	o14	o13	o12	o11	o10	o9	o8
0x2000:03	1	o23	o22	o21	o20	o19	o18	o17	o16
0x2000:04	1	o31	o30	o29	o28	o27	o26	o25	o24
0x2000:05	1	o39	o38	o37	o36	o35	o34	o33	o32
0x2000:06	1	o47	o46	o45	o44	o43	o42	o41	o40
0x2000:07	1	o55	o54	o53	o52	o51	o50	o49	o48
0x2000:08	1	o63	o62	o61	o60	o59	o58	o57	o56
0x2000:09	1	o71	o70	o69	o68	o67	o66	o65	o64
0x2000:0A	1	o79	o78	o77	o76	o75	o74	o73	o72
0x2000:0B	1	o87	o86	o85	o84	o83	o82	o81	o80
0x2000:0C	1	o95	o94	o93	o92	o91	o90	o89	o88
0x2000:0D	1	o103	o102	o101	o100	o99	o98	o97	o96
0x2000:0E	1	o111	o110	o109	o108	o107	o106	o105	o104
0x2000:0F	1	o119	o118	o117	o116	o115	o114	o113	o112
0x2000:10	1	o127	o126	o125	o124	o123	o122	o121	o120
0x2000:11	1	LED status							
0x2000:12	1	Table number							
0x2000:13	1	Segment number							

Index (hex)	Size (byte)	Contents
0x2000:14 … 0x2000:20	13	Byte 0 to 12 of payload

4.3.5 Mapping suggestion input data

Index (hex)	Size (byte)	Contents							
		Virtual inputs i0 … i127							
		Bit 7	…						Bit 0
0x2100:01	1	i7	i6	i5	i4	i3	i2	i1	i0
0x2100:02	1	i15	i14	i13	i12	i11	i10	i9	i8
0x2100:03	1	i23	i22	i21	i20	i19	i18	i17	i16
0x2100:04	1	i31	i30	i29	i28	i27	i26	i25	i24
0x2100:05	1	i39	i38	i37	i36	i35	i34	i33	i32
0x2100:06	1	i47	i46	i45	i44	i43	i42	i41	i40
0x2100:07	1	i55	i54	i53	i52	i51	i50	i49	i48
0x2100:08	1	i63	i62	i61	i60	i59	i58	i57	i56
0x2100:09	1	i71	i70	i69	i68	i67	i66	i65	i64
0x2100:0A	1	i79	i78	i77	i76	i75	i74	i73	i72
0x2100:0B	1	i87	i86	i85	i84	i83	i82	i81	i80
0x2100:0C	1	i95	i94	i93	i92	i91	i90	i89	i88
0x2100:0D	1	i103	i102	i101	i100	i99	i98	i97	i96
0x2100:0E	1	i111	i110	i109	i108	i107	i106	i105	i104
0x2100:0F	1	i119	i118	i117	i116	i115	i114	i113	i112
0x2100:10	1	i127	i126	i125	i124	i123	i122	i121	i120
0x2100:11	1	Reserved							
0x2100:12	1	Table number							
0x2100:13	1	Segment number							
0x2100:14 … 0x2100:20	13	Reserved							

4.4 PNOZ m ES EtherCAT

4.4.1 Service Data Objects (SDOs)

All the relevant device parameters and current process data of the EtherCAT are entered in the object directory of the PNOZmulti 2. These can be read and written via Service Data Objects (SDOs).

To enable the fieldbus module to be incorporated easily within a EtherCAT network, the object directory is available as an XML file. The content is described in the chapter entitled Service data [📖 73].

4.4.2 Service data

The service data for EtherCAT is divided into the following data areas:

- Process data - Base unit and expansion modules
- Process data - Fieldbus module and communication module
- Diagnostic words
- Enable elements (only PNOZ m B0)
- Project data
- Device data

The content of the service data and addressing are described in the chapter entitled Service data [📖 73].

4.4.3 Process Data Objects (PDOs)

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

PDO (hex)	Size	Name	Index (hex)	Sub-Index(hex)	Content
0x1A00	32	TxPDO	2000	01 - 20	Output data (from the PNOZmulti)
0x1A01	128	TxPDO	2000	01 - 11	Default configuration (freely configurable)
			2001	01 - 08	
			2001	49 - 50	
			2002	01 - 08	
			2002	25 – 2C	
			2002	49–50	
			2007	01–08	
			2007	25–2C	
			2007	49–50	
			200A	01–2E	
0x1600	19	RxPDO	2100	01-13	Input data (to the PNOZmulti)

Key to abbreviations:
TxPDO: Transmit Process Data Object
RxPDO: Receive Process Data Object

4.4.4 Mapping suggestion output data

Index (hex)	Size (byte)	Contents							
		Virtual outputs o0 … o127							
0x2000:01	1	Bit 7	…						Bit 0
		o7	o6	o5	o4	o3	o2	o1	o0
0x2000:02	1	o15	o14	o13	o12	o11	o10	o9	o8
0x2000:03	1	o23	o22	o21	o20	o19	o18	o17	o16
0x2000:04	1	o31	o30	o29	o28	o27	o26	o25	o24
0x2000:05	1	o39	o38	o37	o36	o35	o34	o33	o32
0x2000:06	1	o47	o46	o45	o44	o43	o42	o41	o40
0x2000:07	1	o55	o54	o53	o52	o51	o50	o49	o48
0x2000:08	1	o63	o62	o61	o60	o59	o58	o57	o56
0x2000:09	1	o71	o70	o69	o68	o67	o66	o65	o64
0x2000:0A	1	o79	o78	o77	o76	o75	o74	o73	o72
0x2000:0B	1	o87	o86	o85	o84	o83	o82	o81	o80
0x2000:0C	1	o95	o94	o93	o92	o91	o90	o89	o88
0x2000:0D	1	o103	o102	o101	o100	o99	o98	o97	o96
0x2000:0E	1	o111	o110	o109	o108	o107	o106	o105	o104
0x2000:0F	1	o119	o118	o117	o116	o115	o114	o113	o112
0x2000:10	1	o127	o126	o125	o124	o123	o122	o121	o120
0x2000:11	1	LED status							
0x2000:12	1	Table number							
0x2000:13	1	Segment number							
0x2000:14 … 0x2000:20	13	Byte 0 to 12 of payload							

4.4.5 Mapping suggestion input data

Index (hex)	Size (byte)	Contents							
		Virtual inputs i0 … i127							
		Bit 7	…						Bit 0
0x2100:01	1	i7	i6	i5	i4	i3	i2	i1	i0
0x2100:02	1	i15	i14	i13	i12	i11	i10	i9	i8
0x2100:03	1	i23	i22	i21	i20	i19	i18	i17	i16
0x2100:04	1	i31	i30	i29	i28	i27	i26	i25	i24
0x2100:05	1	i39	i38	i37	i36	i35	i34	i33	i32

Index (hex)	Size (byte)	Contents							
0x2100:06	1	i47	i46	i45	i44	i43	i42	i41	i40
0x2100:07	1	i55	i54	i53	i52	i51	i50	i49	i48
0x2100:08	1	i63	i62	i61	i60	i59	i58	i57	i56
0x2100:09	1	i71	i70	i69	i68	i67	i66	i65	i64
0x2100:0A	1	i79	i78	i77	i76	i75	i74	i73	i72
0x2100:0B	1	i87	i86	i85	i84	i83	i82	i81	i80
0x2100:0C	1	i95	i94	i93	i92	i91	i90	i89	i88
0x2100:0D	1	i103	i102	i101	i100	i99	i98	i97	i96
0x2100:0E	1	i111	i110	i109	i108	i107	i106	i105	i104
0x2100:0F	1	i119	i118	i117	i116	i115	i114	i113	i112
0x2100:10	1	i127	i126	i125	i124	i123	i122	i121	i120
0x2100:11	1	Reserved							
0x2100:12	1	Table number							
0x2100:13	1	Segment number							
0x2100:14 … 0x2100:20	13	Reserved							

4.5 PNOZ m ES Powerlink

4.5.1 Service Data Objects (SDOs)

All the relevant device parameters and current process data of the Ethernet POWERLINK are entered in the object directory of the PNOZmulti 2. These can be read and written via Service Data Objects (SDOs).

To enable the fieldbus module to be incorporated easily within a Ethernet POWERLINK network, the object directory is available as an XDD file. The content is described in the chapter entitled Service data [📖 73].

4.5.2 Service data

The service data for Ethernet POWERLINK is divided into the following data areas:

- Process data - Base unit and expansion modules
- Process data - Fieldbus module and communication module
- Diagnostic words
- Enable elements (only PNOZ m B0)
- Project data
- Device data

The content of the service data and addressing are described in the chapter entitled Service data [📖 73].

4.5.3 Process Data Objects (PDOs)

A PDO can be assembled individually from the SDOs.

Maximum size for the PDOs:

- 254 Bytes input data
- 32 Bytes output data

The inputs and outputs are viewed from the Managing Node.

4.5.4 Mapping suggestion output data

Index (hex)	Size (byte)	Contents							
		Virtual outputs o0 … o127							
0x2000:01	1	Bit 7	…						Bit 0
		o7	o6	o5	o4	o3	o2	o1	o0
0x2000:02	1	o15	o14	o13	o12	o11	o10	o9	o8
0x2000:03	1	o23	o22	o21	o20	o19	o18	o17	o16
0x2000:04	1	o31	o30	o29	o28	o27	o26	o25	o24
0x2000:05	1	o39	o38	o37	o36	o35	o34	o33	o32
0x2000:06	1	o47	o46	o45	o44	o43	o42	o41	o40
0x2000:07	1	o55	o54	o53	o52	o51	o50	o49	o48
0x2000:08	1	o63	o62	o61	o60	o59	o58	o57	o56
0x2000:09	1	o71	o70	o69	o68	o67	o66	o65	o64
0x2000:0A	1	o79	o78	o77	o76	o75	o74	o73	o72
0x2000:0B	1	o87	o86	o85	o84	o83	o82	o81	o80
0x2000:0C	1	o95	o94	o93	o92	o91	o90	o89	o88
0x2000:0D	1	o103	o102	o101	o100	o99	o98	o97	o96
0x2000:0E	1	o111	o110	o109	o108	o107	o106	o105	o104
0x2000:0F	1	o119	o118	o117	o116	o115	o114	o113	o112
0x2000:10	1	o127	o126	o125	o124	o123	o122	o121	o120
0x2000:11	1	LED status							
0x2000:12	1	Table number							
0x2000:13	1	Segment number							
0x2000:14 … 0x2000:20	13	Byte 0 to 12 of payload							

4.5.5 Mapping suggestion input data

Index (hex)	Size (byte)	Contents							
		Virtual inputs i0 … i127							
		Bit 7	…						Bit 0
0x2100:01	1	i7	i6	i5	i4	i3	i2	i1	i0
0x2100:02	1	i15	i14	i13	i12	i11	i10	i9	i8
0x2100:03	1	i23	i22	i21	i20	i19	i18	i17	i16
0x2100:04	1	i31	i30	i29	i28	i27	i26	i25	i24
0x2100:05	1	i39	i38	i37	i36	i35	i34	i33	i32
0x2100:06	1	i47	i46	i45	i44	i43	i42	i41	i40
0x2100:07	1	i55	i54	i53	i52	i51	i50	i49	i48
0x2100:08	1	i63	i62	i61	i60	i59	i58	i57	i56
0x2100:09	1	i71	i70	i69	i68	i67	i66	i65	i64
0x2100:0A	1	i79	i78	i77	i76	i75	i74	i73	i72
0x2100:0B	1	i87	i86	i85	i84	i83	i82	i81	i80
0x2100:0C	1	i95	i94	i93	i92	i91	i90	i89	i88
0x2100:0D	1	i103	i102	i101	i100	i99	i98	i97	i96
0x2100:0E	1	i111	i110	i109	i108	i107	i106	i105	i104
0x2100:0F	1	i119	i118	i117	i116	i115	i114	i113	i112
0x2100:10	1	i127	i126	i125	i124	i123	i122	i121	i120
0x2100:11	1	Reserved							
0x2100:12	1	Table number							
0x2100:13	1	Segment number							
0x2100:14 … 0x2100:20	13	Reserved							

4.6 PNOZ m ES Profinet

4.6.1 Cyclical input and output data

Virtual inputs and outputs can be requested or set directly via the following modules. Each element can be selected individually in the master control system, e.g. virtual inputs i0-31. The data width is also established this way.

Input data

The Master writes to the virtual inputs of the PNOZmulti 2.

Description	Input data from PNOZmulti 2
Virtual inputs i0 – i31	4 Input Bytes
Virtual inputs i32 – i63	4 Input Bytes
Virtual inputs i64 – i95	4 Input Bytes
Virtual inputs i96 – i127	4 Input Bytes

Output data

The Master reads the virtual outputs of the PNOZmulti 2.

Description	Output data from PNOZmulti 2
Virtual outputs o0 – o31	4 Output Bytes
Virtual outputs o32 – o63	4 Output Bytes
Virtual outputs o64 – o95	4 Output Bytes
Virtual outputs o96 – o127	4 Output Bytes

4.6.2 Access to table segments

The data in the tables (see Service data [📖 73]) can be requested via the following modules.

Input data

The Master requests a table segment in the form:

Description	Input data from PNOZmulti 2
Table and segment number	2 Input Bytes

Output data

The PNOZmulti 2 replies in the form:

Description	Output data from PNOZmulti 2
Table and segment number + 13 Byte payload	15 Output Bytes

A detailed description is available in the chapter entitled Access to table segments [📖 16].

4.6.3 LED status

The LED status of the PNOZmulti 2 can be requested directly via the following module.

Description	Output data from PNOZmulti 2
LED status	1 Output Byte

- 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

4.6.4 Service data

The service data for PROFINET is divided into the following data areas:

- Process data - Base unit and expansion modules
- Process data - Fieldbus module and communication module
- Diagnostic words
- Enable elements (only PNOZ m B0)
- Project data
- Device data

The content of the service data and addressing are described in the chapter entitled Service data [📖 73].

4.6.5 Diagnostic data PNOZ m ES Profinet

These bits contain the diagnostic data

Unit_Diag_ Bit	Contents
01	STOP, base unit is in STOP condition
02	Base unit was stopped by the Configurator
03	Reserved
04	External error
05	Internal error
06	External error on the inputs
07	Internal error on the inputs
08	External error on the outputs
09	Internal error on the outputs
10	No connection to base unit
11 … 21	Reserved
22	Error on the base unit
23	Reserved
24	Error on the 1st expansion module, right
25	Error on the 2nd expansion module, right
26	Error on the 3rd expansion module, right
27	Error on the 4th expansion module, right

Unit_Diag_ Bit	Contents
28	Fault on the 5th expansion module, right
29	Fault on the 6th expansion module, right
30 … 39	Reserved
40	Fault on 1st expansion module, left
41	Fault on 2nd expansion module, left
42	Fault on 3rd expansion module, left
43	Fault on 4th expansion module, left
44 ... 45	Reserved
46	Error on the communication module
47 … 55	Reserved
56	Error in the configuration, F_81
57	Error in the user program, F_82
58	Error in the periphery, F_83
59	Error on the speed monitor, F_84
60	Error on the bus module, F_85
61	Error on the link module, F_86
62	Error on the analogue input module, F_87
63	Reserved
64	Error on the input/output module, F_89
65	Error on the Ethernet module, F_8A
66	Internal error on the left-hand expansion module, F_90
67	Internal periphery error, F_93
68	Internal error on the right-hand expansion module, F_94
69	Internal error on the right-hand expansion module, F_95
70	Internal self test error, F_B0
71	Internal data error, F_B1
72	Internal parameter error, F_B2
73	Internal serial/I2C error, F_B3
74	Internal time error, F_B4
75	Internal processor error, F_B5
76	Internal parameter error, F_B6

Unit_Diag_ Bit	Contents
77	Internal compare error, F_B7
78	Internal sequence error, F_B8
79	Internal periphery error, F_B9
80	Internal error on bus module, F_BA
81	Internal error on link module, F_BB
82	Internal error on speed monitor, F_BC
83	Internal error on analogue input module, F_BD
84	Internal error on link module, F_BE
85	Internal error on link module, F_BF
86	Internal error on the left-hand expansion module, F_C0
87	Internal error on the left-hand expansion module, F_C1
88	Internal error on the Ethernet module, F_C2
89	Internal compare error, F_C3
90	Internal error on the right-hand expansion module, F_C4
91	Internal error on the right-hand expansion module, F_C5
92 … 95	Reserved

- Explanation: Bits 56 to 91 display the last content of the error stack (corresponds to the error class). The bits are deleted as soon as PNOZmulti 2 returns to a RUN state.

4.7 PNOZ m ES Ethernet/IP

4.7.1 Cyclical input and output data

17 or 32 bytes must always be sent and received for communication with the PNOZmulti.

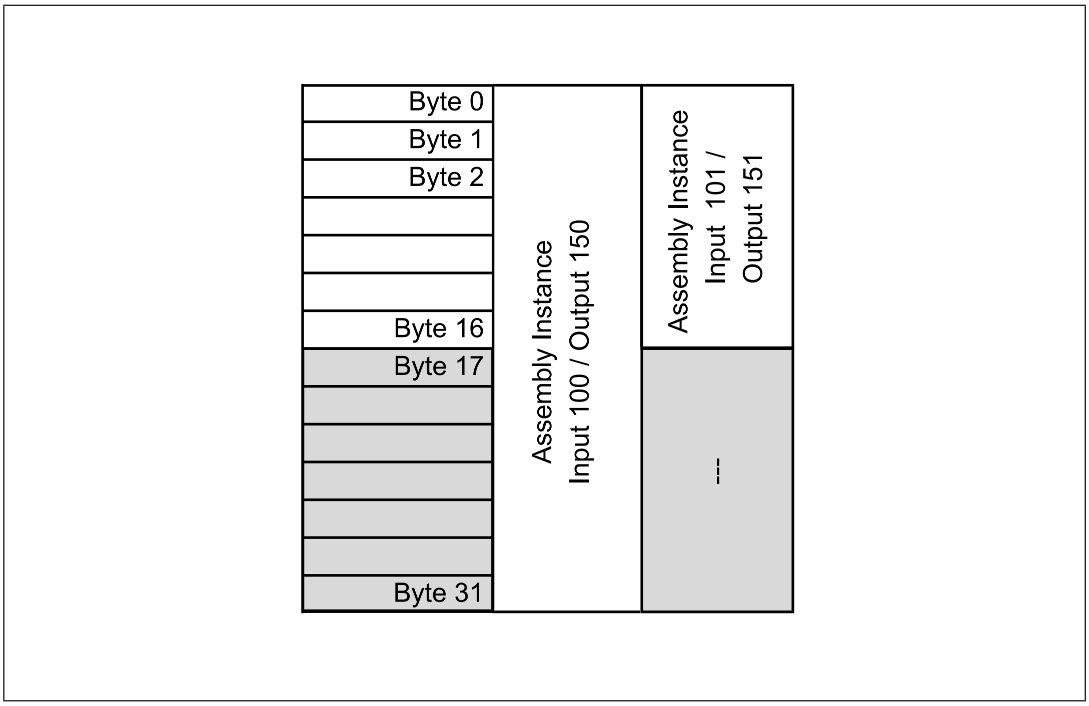
Fixed parameters for the input/output assembly instances are set in the fieldbus module interfaces. The following data lengths can be selected:

Assembly Instance Input	Data length	Description
100	32 Bytes	Inputs, LEDs, tables
101	17 Bytes	Inputs, LEDs

Assembly Instance Output	Data length	Description
150	32 Bytes	Outputs, LEDs, tables
151	17 Bytes	Outputs, LEDs

Assembly Instance Configuration	Data length	Description
4	0 Bytes	-

The content and structure of the Assembly Instances is described in the chapter entitled Input data (to the PNOZmulti 2) [📖 14] and Output data (from the PNOZmulti 2) [📖 15].



4.7.2 Service data

The service data for EtherNet/IP is divided into the following data areas:

- Process data - Base unit and expansion modules
- Process data - Fieldbus module and communication module
- Diagnostic words
- Enable elements (only PNOZ m B0)
- Project data
- Device data

The content of the service data and addressing are described in the chapter entitled Service data [📖 73].

4.8 PNOZ m ES CC-Link

4.8.1 Input and output data

Virtual inputs and outputs can be requested or set directly via the following addresses. The implementation to the names of the inputs and outputs in PNOZmulti 2 is performed as in the table listed below.

The data is structured as follows:

- **Input area**
  - Inputs on PNOZmulti Configurator: i00 .. i127
  - Input data CC-Link: RYmn.. RY(m+50)n, RWw l .. RWw l+2

with l = address can be set as required on the master side (word address)

with m = address can be set as required on the master side (Bit address)

with n = 0 .. F (Bit number)

Example: (with m=100) i23 -> n = 7 -> RY117

Bit addressed input data i00 - i87

n	**F**	**E**	**D**	**C**	**B**	**A**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RY m n	i15	i14	i13	i12	i11	i10	i09	i08	i07	i06	i05	i04	i03	i02	i01	i00
RY(m+10)n	i31	i30	i29	i28	i27	i26	i25	i24	i23	i22	i21	i20	i19	i18	i17	i16
RY(m+20)n	i47	i46	i45	i44	i43	i42	i41	i40	i39	i38	i37	i36	i35	i34	i33	i32
RY(m+30)n	i63	i62	i61	i60	i59	i58	i57	i56	i55	i54	i53	i52	i51	i50	i49	i48
RY(m+40)n	i79	i78	i77	i76	i75	i74	i73	i72	i71	i70	i69	i68	i67	i66	i65	i64
RY(m+50)n									i87	i86	i85	i84	i83	i82	i81	i80

Word addressed input data i88 - i127

	High byte								**Low byte**							
Bit no	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RWw l	i103	i102	i101	i100	i99	i98	i97	i96	i95	i94	i93	i92	i91	i90	i89	i88
RWw l+1	i119	i118	i117	i116	i115	i114	i113	i112	i111	i110	i109	i108	i107	i106	i105	i104
RWw l+2	-	-	-	-	-	-	-	-	i127	i126	i125	i124	i123	i122	i121	i120

▶ **Output range**

– Outputs on PNOZmulti Configurator: o00 .. o127

– Output data CC-Link: RXmn .. RX(m+50)n, RWr l .. RWr l+2

with l = address can be set as required on the master side (word address)

with m = address can be set as required on the master side (Bit address)

with n = 0 .. F (Bit number)

Example: (with m=100) o22 -> n = 6 -> RX116

Bit addressed output data o00 - o87

n	**F**	**E**	**D**	**C**	**B**	**A**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RX m n	o15	o14	o13	o12	o11	o10	o09	o08	o07	o06	o05	o04	o03	o02	o01	o00
RX(m+10)n	o31	o30	o29	o28	o27	o26	o25	o24	o23	o22	o21	o20	o19	o18	o17	o16
RX(m+20)n	o47	o46	o45	o44	o43	o42	o41	o40	o39	o38	o37	o36	o35	o34	o33	o32
RX(m+30)n	o63	o62	o61	o60	o59	o58	o57	o56	o55	o54	o53	o52	o51	o50	o49	o48
RX(m+40)n	o79	o78	o77	o76	o75	o74	o73	o72	o71	o70	o69	o68	o67	o66	o65	o64
RX(m+50)n									o87	o86	o85	o84	o83	o82	o81	o80

Word addressed output data o88 - o127

	High byte								Low byte							
Bit No.	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RWr I	o103	o102	o101	o100	o99	o98	o97	o96	o95	o94	o93	o92	o91	o90	o89	o88
RWr I +1	o119	o118	o117	o116	o115	o114	o113	o112	o111	o110	o109	o108	o107	o106	o105	o104
RWr I +2	-	-	-	LED RUN FS	LED DIAG	LED FAULT	LED IFAULT	LED OFAULT	o127	o126	o125	o124	o123	o122	o121	o120

4.8.2 Service data

The service data for EtherNet/IP is divided into the following data areas:

- Process data - Base unit and expansion modules
- Process data - Fieldbus module and communication module
- Diagnostic words
- Enable elements (only PNOZ m B0)
- Project data
- Device data

The content of the service data and addressing are described in the chapter entitled Service data [📖 73].

4.8.3 LED status

The LED status of PNOZmulti 2 can be read out via the following High Byte.

	High Byte								Low Byte							
Bit no	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RWr (I+2)	LED Byte								Output data o120 - o127							

- 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 FS is lit
- Bit 5-7: Reserved

4.8.4 Access to table segments

The data in the tables can be requested via the following addresses.

Input data

The Master requests a table segment:

	High Byte	Low Byte
RWw (I+3)	Segment number	Table number

Output data

PNOZmulti 2 answers as follows:

	High Byte	Low Byte
RWr (I+3)	Segment number	Table number
RWr(I+4)	Segment Byte 1	Segment Byte 0
RWr(I+5)	Segment Byte 3	Segment Byte 2
RWr(I+6)	Segment Byte 5	Segment Byte 4
RWr(I+7)	Segment Byte 7	Segment Byte 6
RWr(I+8)	Segment Byte 9	Segment Byte 8
RWr(I+9)	Segment Byte 11	Segment Byte 10
RWr(I+A)	Reserved	Segment Byte 12
RWr(I+B)	Reserved	Reserved

5 ETH/RS232 interfaces

5.1 Overview

The communication interface on the configurable control system PNOZmulti 2 is used to

- Download and upload the project,
- Read the diagnostic words,
- Read the error stack.

Base unit PNOZ m B1

The base unit PNOZ m B1 has an Ethernet interface integrated in the device.

Base unit PNOZ m B0

The base unit PNOZ m B0 does not contain an integrated interface. Communication modules with interface can be connected:

- **PNOZ m ES RS232**

  The communication module provides a serial RS232 interface
- **PNOZ m ES ETH**

  The communication module provides 2 Ethernet interfaces

5.2 Communication Module PNOZ m ES RS232

The connection to the RS 232 interface of the communication partner and the communication module PNOZ m ES RS232 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

5.3 Communication Module PNOZ m ES ETH

5.3.1 Introduction

This chapter describes the special features of communication with the expansion module PNOZ m ES ETH via Ethernet and Modbus/TCP. Access to data from the PNOZmulti 2 via tables and segments is described in Chapter Basics [📖 14].

5.3.2 Overview

The expansion module PNOZ m ES ETH provides an interface to Ethernet. Via this interface, the configurable control system PNOZmulti 2 can be connected via Ethernet to control systems that support Modbus/TCP. Modbus/TCP is designed for fast data exchange at field level. The expansion module PNOZ m ES ETH is a passive Modbus/TCP subscriber (Server). The basic functions of communication with Ethernet or Modbus/TCP conform to IEEE 802.3. The central controller (Client) reads output information from the Servers and writes input information to the Servers as part of each cycle. In addition to the cyclical payload transmission, the PNOZ m ES ETH also has diagnostic functions.

The connections are established via two RJ45 sockets.

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

The communication module PNOZ m ES ETH supports Modbus/TCP [📖 59].

The module can manage up to 8 Modbus/TCP connections and up to 4 PG port (Port 9000) connections.

5.3.3 Module features

- Can be configured in the PNOZmulti Configurator
- Network protocols: TCP/IP, Modbus/TCP
- Status indicators for communication and display of errors
- Transmission rate 10 MBit/s (10BaseT) and 100 MBit/s (100BaseTX), full and half duplex

5.3.4 Modbus/TCP

There is no need to configure a connection on the PNOZ m ES ETH. Port 502 is used in accordance with the Modbus/TCP specification.

For details of Modbus/TCP Modbus/TCP [📖 59]

5.4 Integrated Ethernet interface

5.4.1 Overview

The base unit PNOZ m B1 has an integrated Ethernet interface. Via this interface, other systems can be connected that support TCP/IP. In addition to the transfer of usable data, the PNOZ m B1 also has diagnostic functions.

The connections are established via two RJ45 sockets.

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

Communication is via port 9000 as a standard, if nothing else is set at the base unit.

5.4.2 Module features

- Can be configured in the PNOZmulti Configurator
- Network protocols: TCP/IP
- Status indicators for communication and display of errors
- Transmission rate 10 MBit/s (10BaseT) and 100 MBit/s (100BaseTX)

5.5 Communication procedure

The PNOZmulti 2 is always the connection's Server in the communication; the communication partner (PC, SPS) is the Client. With both communication interfaces, the bytes are sent in the same sequence.

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 2. Requests are used to receive data from the PNOZmulti 2 or send data to the PNOZmulti 2:

1. Request:

The user sends a request to the PNOZmulti 2 via the communications partner.

2. Response:

The PNOZmulti 2 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.

5.6 Telegram structure

The telegram used for communication consists of:

- a header,
- the payload,
- a footer.

Telegram section	Byte	Request		Byte	Response
Header	0	0x05		0	0x05
	1	0x15		1	0x15
	2	0x00		2	0x00
	3	Payload amount +5		3	Payload amount +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
Payload	8	Payload Byte 0		8	Payload Byte 0
	9	Payload Byte 1		9	Payload Byte 1
	10	Payload Byte 2		10	Payload Byte 2
	...	...		...	...
	Payload amount +7	Payload Byte n		Payload amount +7	Payload Byte n
Footer	Payload amount +8	BCC		Payload amount +8	BCC
	Payload amount +9	0x10		Payload amount +9	0x10

5.6.1 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 [📖 47]
  - 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 [📖 58].
- Byte 5: High Byte of segment number
- Byte 6: Low Byte of segment number
- Byte 7
  - Request: Always 0x00
  - Response: Reserved

5.6.2 Usable data

The telegram contains payload from Byte 8. The content and number of payload bytes depend on the request. 0 - 40 bytes of payload can be transmitted. If there is no payload available, the BCC (Block Control Check) will follow directly after Byte 7.

The last byte of payload is the byte "Payload amount + 7".

If a telegram contains 13 payload bytes, for example, Byte 20 will be the last payload byte.

5.6.3 Footer

The bytes "Payload amount + 8" and "Payload amount + 9" contain the

- check sum (Block Control Check = BCC) in the byte "Payload amount + 8":

  The check sum is calculated as follows:

  BCC = 0 – (Byte 4 + ... + Byte "Payload amount + 7")
- 0x10 as identifier for the telegram end in the byte "Payload amount + 9".

Example for 13 bytes of payload:

- Check sum in Byte 21
  - Content: 0 - (Sum of the Bytes 4 + … + Byte 20)
- 0x10 as identifier for the telegram end is in Byte 22.

5.7 Usable data

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

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

The virtual inputs are defined by the communication partner and transferred to the PNOZmulti 2 . 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
…	...	...	...	...	...	...	...	...
15	i127	i126	i125	i124	i123	i122	i121	i120

5.7.2 Watchdog (PNOZ m B0)

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.

5.7.3 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
...	...	...	...	...	...	...	...	...
15	o127	o126	o125	o124	o123	o122	o121	o120

5.7.4 LED status

The statuses of the LEDs are stored in a byte.

Status of LEDs with PNOZ m B0:

- 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

Status of LEDs with PNOZ m B1:

- 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 FS is lit
- Bit 5: Reserved
- Bit 6 = 1: LED RUN ST is lit
- Bit 7: Reserved

5.7.5 Tables

Further information can be requested in table form.

A table consists of one or more PNOZmulti 2 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 2repeats the two numbers and sends the requested data.

There are a total of 11 tables, with the following contents:

Table 20:	Process data, base unit
Table 21:	Process data, right-hand expansion modules
Table 22	Process data, left-hand expansion modules
Table 23:	Process data, fieldbus module / communication module
Table 70:	Diagnostic words
Table 71:	Enable elements (only PNOZ m B0)
Table 80:	Project information
Table 90:	Device data, base unit
Table 91:	Device data, right-hand expansion modules
Table 92	Device data, left-hand expansion modules
Table 93:	Device data, fieldbus module / communication module

The content of the tables is described in detail in the chapter entitled Service data [📖 73].

5.8 Requests

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

The following requests are available:

Request No.	Segment No.	Meaning	PNOZ m B0	PNOZ m B1
0x14	0x01	Sending virtual inputs to PNOZmulti 2	x	x
0x14	0x02	Sending virtual inputs and requesting the LED states from PNOZmulti 2	x	
0x2C	0x01	Requesting virtual input and output data from 2 PNOZmulti 2	x	x
0x2C	0x02	Requesting virtual input and output data from 2 and LED states from PNOZmulti 2	x	
0x2F	0x00	Requesting the diagnostic data in table form from PNOZmulti 2	x	x
0x5C	0x00	Sending of virtual data and requesting virtual output data from PNOZmulti 2 (suitable for a cyclical exchange)	x	

5.8.1 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.

5.8.2 Sending virtual inputs to PNOZmulti 2

Request 0x14 Segment 0x01

The communication partner uses this request to send virtual inputs to the PNOZmulti 2.

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	Byte	Response
0	0x05	0	0x05
1	0x15	1	0x15
2	0x00	2	0x00
3	0x25	3	0x05
4	0x14	4	
5	0x00	5	0x00
6	0x01	6	0x01
7	0x00	7	0x00
8	Virtual inputs Input Byte 0: i7 to i0	8	0x6B
...	...	9	0x10
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		

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 2 with error message 0x63 (request cannot be executed).

5.8.3 Sending virtual inputs and requesting the LED states from PNOZmulti 2

Request 0x14 Segment 0x02

The communication partner uses this request to send virtual inputs to the PNOZmulti 2 in exactly the same way as request 0x14 segment 0x01. It also requests the virtual outputs and the LED status from the PNOZmulti 2 .

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
0	0x05
1	0x15
2	0x00
3	0x26
4	0x14
5	0x00
6	0x02
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	Control Byte
41	BCC =0 - (Byte 4 + ... + Byte 40)
42	0x10

Byte	Response
0	0x05
1	0x15
2	0x00
3	0x16
4	0x94
5	0x00
6	0x02
7	0x00
8	Virtual outputs Output Byte 0: o7 to o0
...	...
23	Virtual outputs Output Byte 15: o127 to o120
24	LED status
25	BCC
26	0x10

For details of the LED status see chapter entitled LED status.

INFORMATION

If a fieldbus module is configured, virtual inputs cannot be activated via the communication module.

5.8.3.1 Control Byte (Byte 40)

Bit 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 2 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

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.

INFORMATION

The watchdog functions of the requests 0x14 segment 0x02 and 0x5C use the same watchdog timer. In other words, the watchdog timer is reset if either request is called.

5.8.4 Requesting virtual input and output data from PNOZmulti 2

Request 0x2C Segment 0x01

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	0x01		6	0x02
7	0x00		7	0x00
8	0xD3		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	BCC
			41	0x10

5.8.5 Requesting virtual input and output data and the LED statuses from PNOZmulti 2

Request 0x2C Segment 0x02

The communication partner uses this request to request the status of the LEDs and 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

5.8.6 Requesting the diagnostic data in table form from PNOZmulti 2

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		Byte	Response
0	0x05		0	0x05
1	0x15		1	0x15
2	0x00		2	0x00
3	0x07		3	0x14
4	0x2F		4	0xAF
5	0x00		5	0x00
6	0x00		6	0x00
7	0x00		7	0x00
8	Table No.		8	Table No.
9	Segment No.		9	Segment No.
10	BCC		10	Byte 0 of Table x, Segment y
11	0x10		...	...
			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

5.8.7 Sending virtual inputs and requesting the virtual output data from PNOZmulti 2 (see fieldbus communication)

Request 0x5C

The communication partner uses this request to send input data to the PNOZmulti and request output data from the PNOZmulti (see section entitled Basics [📖 14]).

As with fieldbus communication, 32 Bytes each are reserved for input and output data (Byte 8 – 39); they are updated approx. every 15 ms.

Byte	Request		Byte	Response
0	0x05		0	0x05
1	0x15		1	0x15
2	0x00		2	0x00
3	0x26		3	0x26
4	0x5C		4	0xDC
5	Control Byte		5	Control Byte
6	0x00		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
...	...		...	...
39	Input Byte 31		39	Output Byte 31
40	0x00		40	Reserved
41	BCC		41	BCC
42	0x10		42	0x10

5.8.7.1 Input data (to the PNOZmulti)

Input Byte	Content
0	i7 to i0
1	i15 to i8
2	i23 to i16
3	i31 to i24
4	i39 to i32
5	i47 to i40
6	i55 to i48
7	i63 to i56
8	i71 to i64
9	i79 to i72

Input Byte	Content
10	i87 to i80
11	i95 to i88
12	i103 to i96
13	i111 to i104
14	i119 to i112
15	i127 to i120
16	0x00
17	Table number
18	Segment number
19	0x00
20	0x00
21	0x00
22	0x00
23	0x00
24	0x00
25	0x00
26	0x00
27	0x00
28	0x00
29	0x00
30	0x00
31	0x00

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

INFORMATION

If a fieldbus module is configured, virtual inputs cannot be activated via the communication module.

5.8.7.2 Output data (from the PNOZmulti)

Output Byte	Content
0	o7 to o0
1	o15 to o8
2	o23 to o16
3	o31 to o24
4	o39 to o32
5	o47 to o40

Output Byte	Content
6	o55 to o48
7	o63 to o56
8	o71 to o64
9	o79 to o72
10	o87 to o80
11	o95 to o88
12	o103 to o96
13	o111 to o104
14	o119 to o112
15	o127 to o120
16	LED status
17	Table number
18	Segment number
19	Payload Byte 0
20	Payload Byte 1
21	Payload Byte 2
22	Payload Byte 3
23	Payload Byte 4
24	Payload Byte 5
25	Payload Byte 6
26	Payload Byte 7
27	Payload Byte 8
28	Payload Byte 9
29	Payload Byte 10
30	Payload Byte 11
31	Payload Byte 12

The content of the tables and segments is described in detail in the section entitled Service data [📖 73].

5.8.7.3 Control Byte (Byte 5)

Bit 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 Re-sponse	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 0x5C 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.

5.9 Troubleshooting

5.9.1 Request format does not meet specifications

If the request format does not meet the specifications, the PNOZmulti 2 sends the following response:

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

5.9.2 Error while executing a request

If an error occurs while executing a request, the PNOZmulti 2 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 2 not ready

6 Modbus/TCP

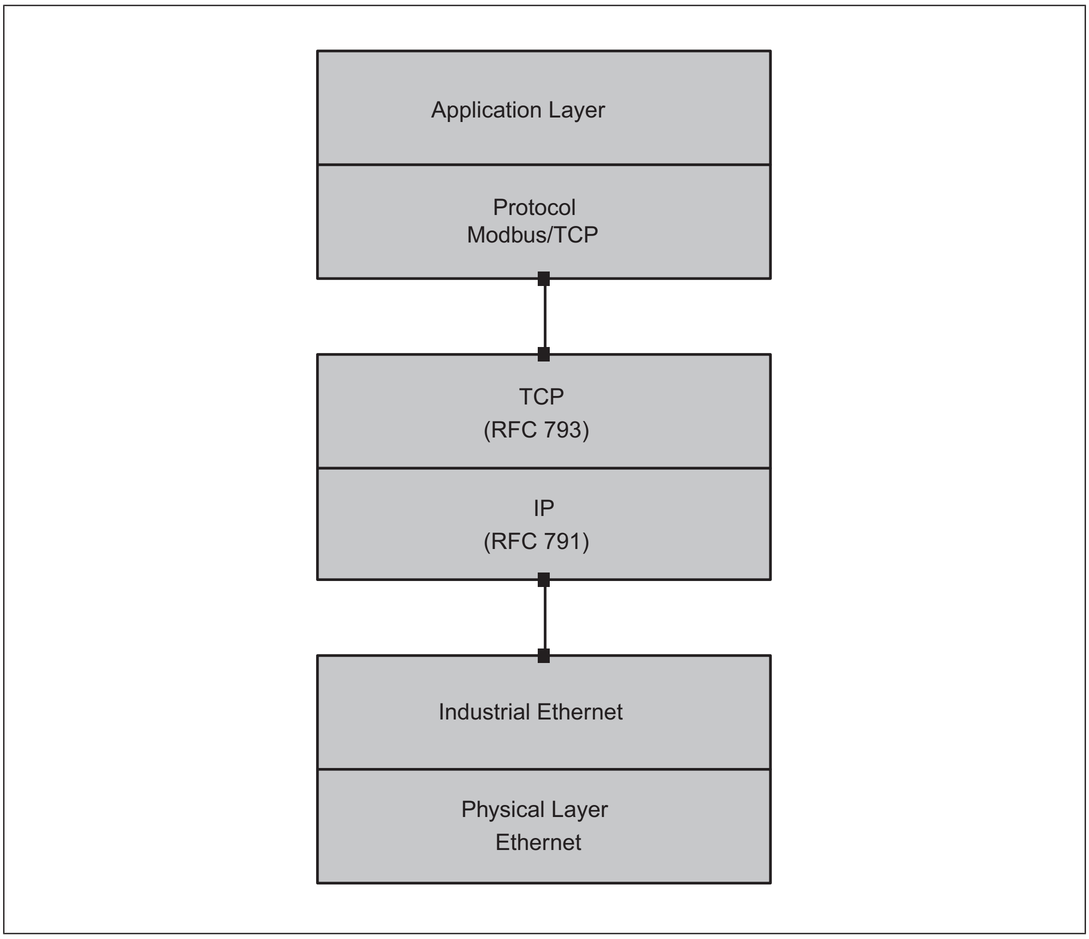
6.1 System requirements

- PNOZmulti Configurator: from Version 9.0.0
- PNOZ m B0 + PNOZ m ES ETH

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).

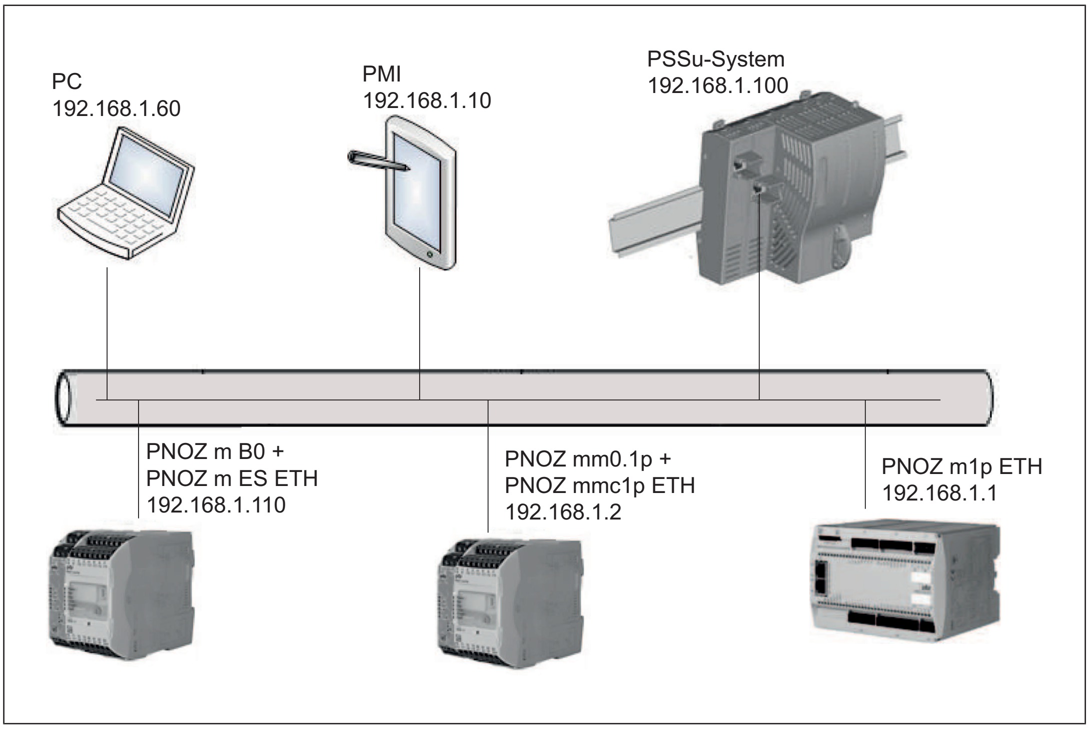
Modbus/TCP is connection-oriented, i.e. before usable data is transferred via Modbus/TCP, a connection must first be established between two Modbus/TCP interfaces. The initiator of the connection is called the Client. The communications partner with which the Client establishes the connection is called 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 2

An interface for communication via Modbus/TCP is provided on the PNOZmulti 2 through the communication module PNOZ m ES ETH.

A PNOZ m ES ETH can manage a max. of 8 Modbus/TCP connections. The PNOZmulti 2 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 2 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 2 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 2 , 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 2 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 (Word/16 Bits) 4x00000 … 4x65535 [read/write]	4x[xxxxx]	4x04108 (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.

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.

6.4.2 Function codes

The following function codes (FC) are available for communication with the PNOZmulti 2 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 connection Client reads word data from the connection Server, data length ≥ 1 Word, content: Diagnostic word (data received from 4x)
FC 04	Read Input Register	The connection Client reads word data from the connection Server, 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)

Error codes on Modbus/TCP

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

6.4.3 Data transfer limits

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

Data transfer		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 Input and output data, watchdog

The table below describes the Modbus/TCP data areas that contain the current state of the virtual inputs of the PNOZmulti 2. 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/ Discrete Input (0x, 1x)	Content	High Byte	Low Byte
0	15... 0	State of virtual inputs 0...15	i15…i8	i7…i0
1	31... 16	State of virtual inputs 16...31	i31…i24	i23…i16
2	47...32	State of virtual inputs 32...47	i47…i40	i39…i32
3	63...48	State of virtual inputs 48...63	i63…i56	i55…i48
4	79... 64	State of virtual inputs 64...79	i79…i72	i71…i64
5	95...80	State of virtual inputs 80...95	i95…i88	i87…i80
6	111...96	State of virtual inputs 96...111	i111…i104	i103…i96
7	127...112	State of virtual inputs 112...127	i127…i120	i119…i112

The table below describes the Modbus/TCP data areas that contain the states of the virtual outputs of the PNOZmulti 2.

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

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 2 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/ Discrete 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.

Bit 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 2 is "0", then the watchdog has been triggered.

6.4.5 Allocation table of the virtual inputs and outputs

Addressing the virtual inputs (coils) in Registers 0 to 7.

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

Allocation of virtual inputs i0, i1 ... i127 to the Bits in Registers 0 to 7

Addressing the virtual outputs (discrete inputs) in Registers 512 to 519.

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

Allocation of virtual outputs o0, o1 ... o127 to the Bits in Registers 512 to 519.

6.4.6 Service data

The service data for Modbus/TCP is divided into the following data areas:

- Process data - Base unit and expansion modules
- Process data - Fieldbus module and communication module
- Diagnostic words
- Enable elements
- Project data
- Device data

The content of the service data and addressing are described in the chapter entitled Service data [📖 73]

6.4.7 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
511	8176	Status of LEDs	Reserved	PNOZmulti 2 LEDs

- 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

6.4.8 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
Project and device data	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
Current state of the virtual inputs	1000 ms

INFORMATION

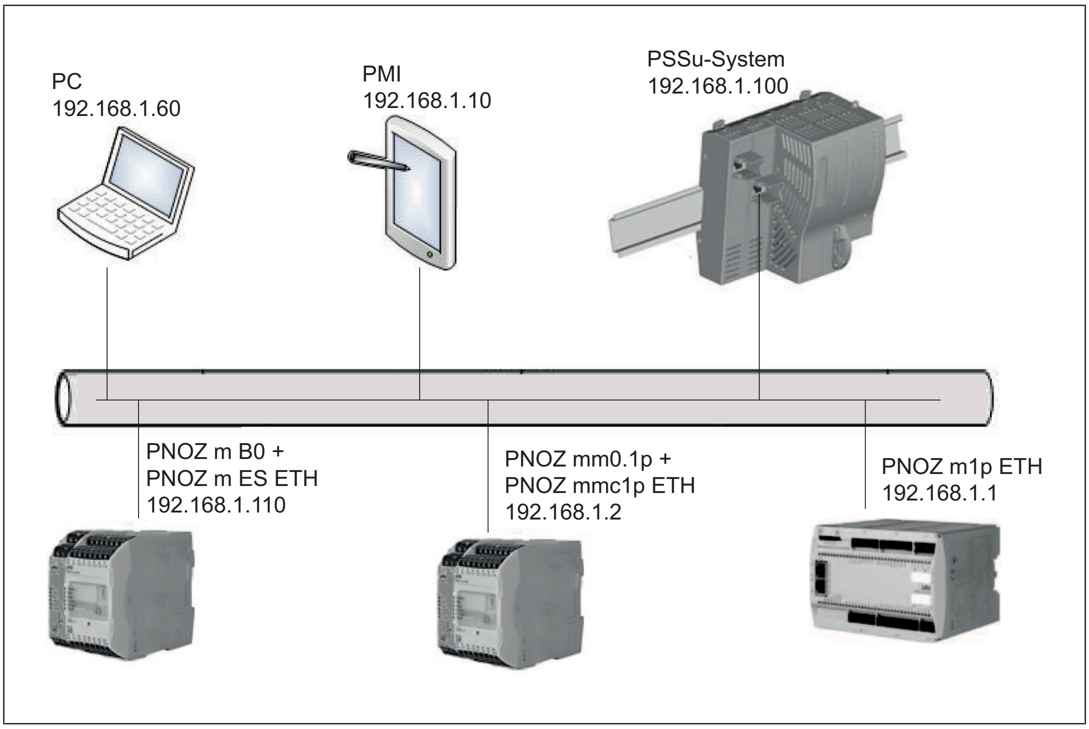
The update time may increase if there are additional TCP/IP connections (e.g. PNOZmulti Configurator, PMI, control system) on the PG port (Port 9000).

6.5 Examples of Clients and Servers

The following subscribers communicate via Modbus/TCP or Ethernet:

- Devices with Server role:
  - PNOZ m ES ETH
  - PNOZ m1p ETH
  - PNOZ mmc1p ETH
- Devices with Client role:
  - PSSu system in the automation system PSS 4000
  - Operator terminal PMI
- PC as programming device for PNOZmulti 2, PSSu system and PMI

The PSSu system and the operator terminal PMI access the configurable control system PNOZmulti 2 (Server role) simultaneously.



7 Diagnostic word

7.1 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)

7.2 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
  - Two-hand button
  - Operating mode selector switch
- Logic elements
  - RS flip-flop
  - Start element
- Press elements
  - Light curtain
- Muting elements
  - Sequential muting
  - Parallel muting
  - Cross muting
- Output elements
  - Output elements with feedback loop
  - Safety valve

7.3 Structure of the diagnostic word

The diagnostic word has 16 Bits:

Bit 15	Bit 14	...	Bit 2	Bit 1	Bit 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). Otherwise, at least one of the bits from 0 ... 15 of the diagnostic word is set and can be

evaluated,
e.g.: Bit 1 on a safety gate element: 1:00000000 00000010
Key: Safety gate was opened

7.4 Evaluate diagnostic word

- **Evaluation in the user program**

  One bit from the diagnostic word can be linked further within the PNOZmulti 2 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

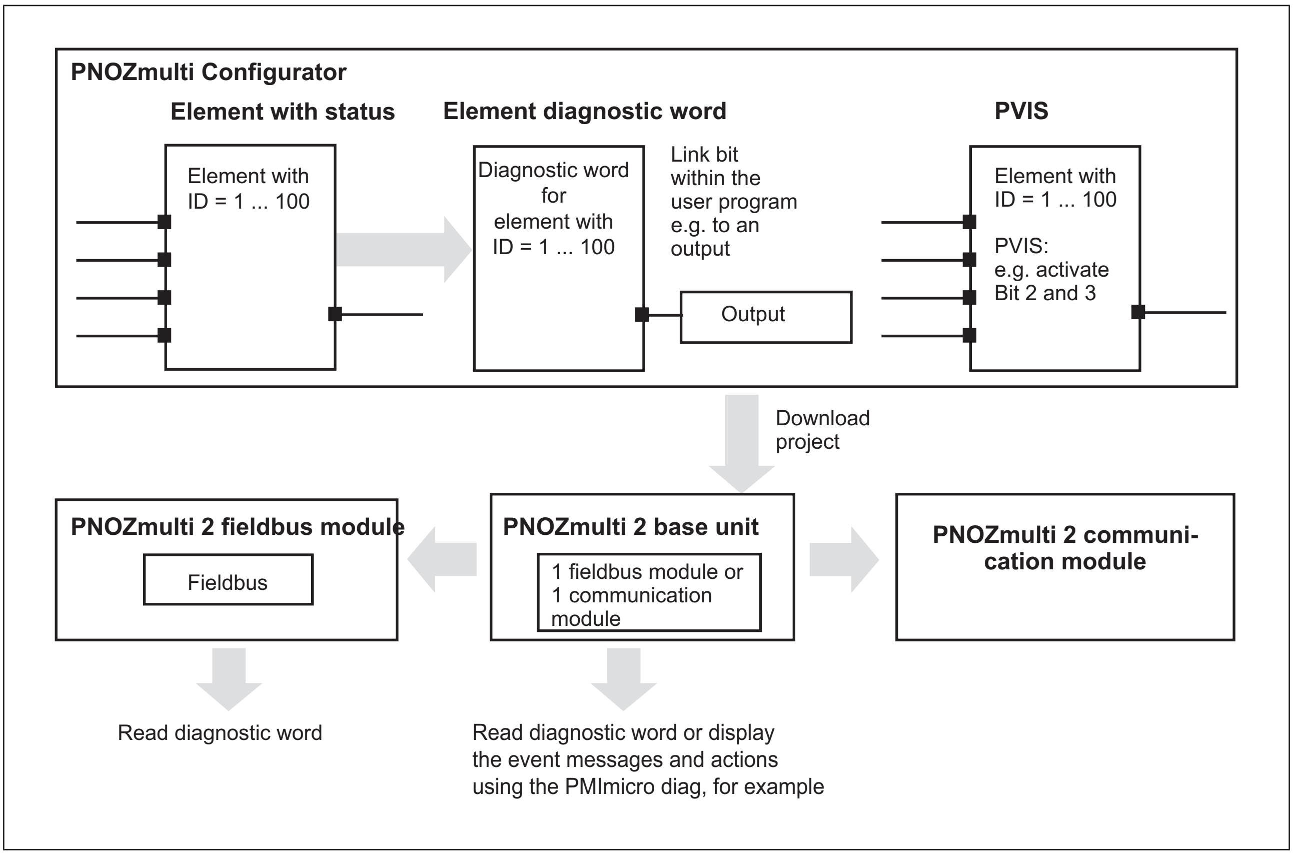
Detailed information on PVIS expanded diagnostics can be found in the PNOZmulti Configurator's online help.

- **Evaluation via the RS232/Ethernet interfaces**

  The diagnostic word is requested via the communication module using the element's ID (see also ETH/RS232 interfaces [📖 41]).

- **Evaluation via a fieldbus**

  The diagnostic word is requested via a connected fieldbus module, via the element's ID (see also Fieldbus Modules [📖 14]).


PNOZmulti Configurator
Element with status
Element diagnostic word
PVIS
Element with ID = 1 ... 100
Diagnostic word for element with ID = 1 ... 100
Link bit within the user program e.g. to an output
Output
Element with ID = 1 ... 100
PVIS: e.g. activate Bit 2 and 3
Download project
PNOZmulti 2 fieldbus module
Fieldbus
PNOZmulti 2 base unit
1 fieldbus module or 1 communication module
PNOZmulti 2 communication module
Read diagnostic word
Read diagnostic word or display the event messages and actions using the PMImicro diag, for example

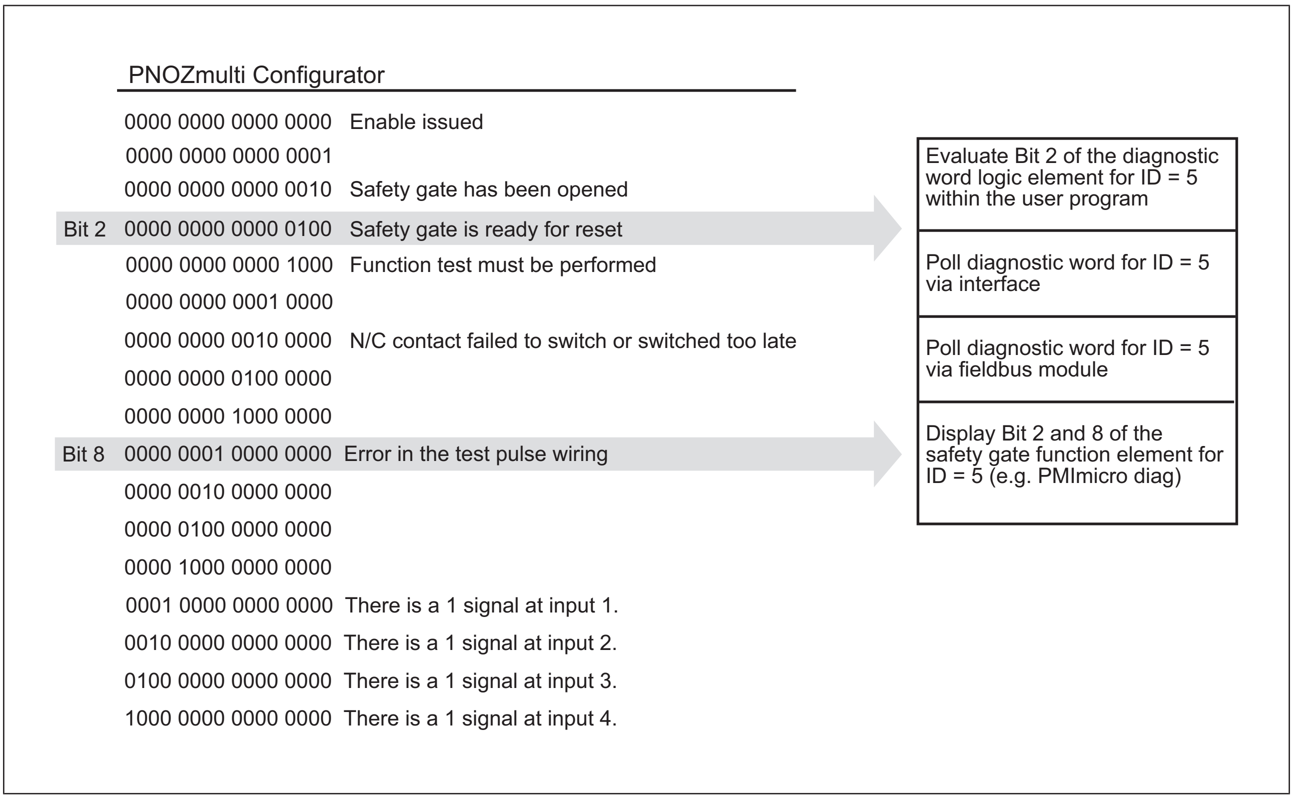
7.4.1 Example - Evaluate diagnostic word of a safety gate

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



PNOZmulti Configurator

	0000 0000 0000 0000	Enable issued
	0000 0000 0000 0001	
	0000 0000 0000 0010	Safety gate has been opened
Bit 2	0000 0000 0000 0100	Safety gate is ready for reset
	0000 0000 0000 1000	Function test must be performed
	0000 0000 0001 0000	
	0000 0000 0010 0000	N/C contact failed to switch or switched too late
	0000 0000 0100 0000	
	0000 0000 1000 0000	
Bit 8	0000 0001 0000 0000	Error in the test pulse wiring
	0000 0010 0000 0000	
	0000 0100 0000 0000	
	0000 1000 0000 0000	
	0001 0000 0000 0000	There is a 1 signal at input 1.
	0010 0000 0000 0000	There is a 1 signal at input 2.
	0100 0000 0000 0000	There is a 1 signal at input 3.
	1000 0000 0000 0000	There is a 1 signal at input 4.

Evaluate Bit 2 of the diagnostic word logic element for ID = 5 within the user program

Poll diagnostic word for ID = 5 via interface

Poll diagnostic word for ID = 5 via fieldbus module

Display Bit 2 and 8 of the safety gate function element for ID = 5 (e.g. PMImicro diag)

8 Service data

The service data of the configurable control system PNOZmulti 2 is divided into various data ranges. The service data is addressed differently, based on the fieldbus. They are fully available after the control system has been in "Run" state for three seconds.

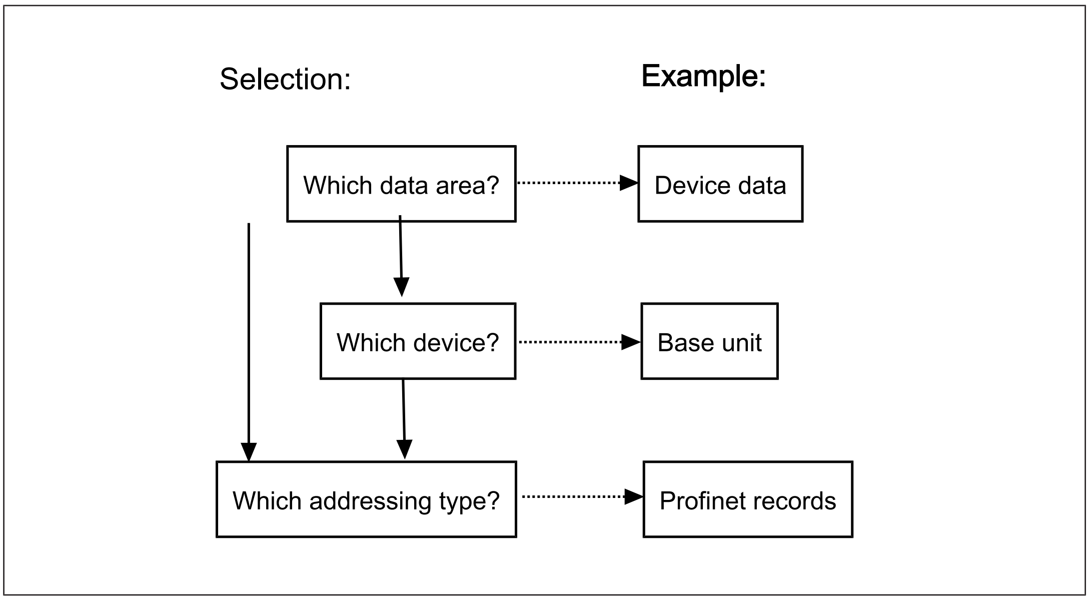
The content of the service data is described generally in the sections below. This is followed by the fieldbus-specific addressing.

Data ranges:

- Process data: Base unit and expansion modules [📖 74]
  - State of inputs i0 … i31 [📖 75]
  - State of outputs o0 … o31 [📖 75]
  - Status of system LEDs [📖 75]
  - Status of IO LEDs [📖 76]
- Process data: Fieldbus and communication module [📖 92]
  - State of virtual inputs i0 … i127 [📖 92]
  - State of virtual outputs o0 … o127 [📖 93]
  - Status of system LEDs [📖 94]
- Diagnostic words [📖 97]
  - Diagnostics [📖 97]
- Enable elements (only PNOZ m B0) [📖 98]
  - Element IDs [📖 98]
- Project data [📖 100]
  - Check sums [📖 100]
  - Date [📖 100]
  - Project name [📖 101]
- Device data [📖 103]
  - Product [📖 103]
  - Firmware [📖 103]
  - Operating hours [📖 104]

The following example illustrates the procedure for selecting certain service data.

In the example, the device data for the base unit is selected with the fieldbus module PROFINET.



Data area:	Device data
Device:	Device data addressing on base unit
Addressing type	

INFORMATION

The service data is only updated piecemeal in each cycle by the PNOZmulti.
Updating all of the data can take up to 500 ms.
An exception is data requested in table form. This is updated fully in each cycle.

8.1 Process data: Base unit and expansion modules

The process data of the base unit and expansion modules consists of 36 Bytes and contains the following information:

- State of the inputs and outputs
- The status of the system LED for
  - supply voltage,
  - Diagnostics
- State of the inputs and outputs, which is displayed on the device.

8.1.1 State of inputs i0 … i31

Byte	Bit 7	...						Bit 0
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	i31	i30	i29	i28	i27	i26	i25	i24

Fieldbus-specific addressing is described in the chapter entitled Process data addressing [📖 78].

8.1.2 State of outputs o0 … o31

Byte	Bit 7	...						Bit 0
4	o7	o6	o5	o4	o3	o2	o1	o0
5	o15	o14	o13	o12	o11	o10	o9	o8
6	o23	o22	o21	o20	o19	o18	o17	o16
7	o31	o30	o29	o28	o27	o26	o25	o24

Fieldbus-specific addressing is described in the chapter entitled Process data addressing [📖 78].

8.1.3 Status of system LEDs

Status of the system LEDs for supply voltage and base unit diagnostics (see also Assignment of system and I/O-LEDs [📖 77])

Byte	Bit 7	...		Bit 4	Bit 3	...		Bit 0
8	System LED 1				System LED 0			
9	System LED 3				System LED 2			
10	System LED 5				System LED 4			
11	Reserved							

LED status	4 Bit coding
Off	0x0
On	0xF
Flashes	0x3

Fieldbus-specific addressing is described in the chapter entitled Process data addressing [📖 78].

8.1.4 Status of IO LEDs

Status of the system LEDs for supply voltage and base unit diagnostics (see also Assignment of system and I/O-LEDs [📖 77])

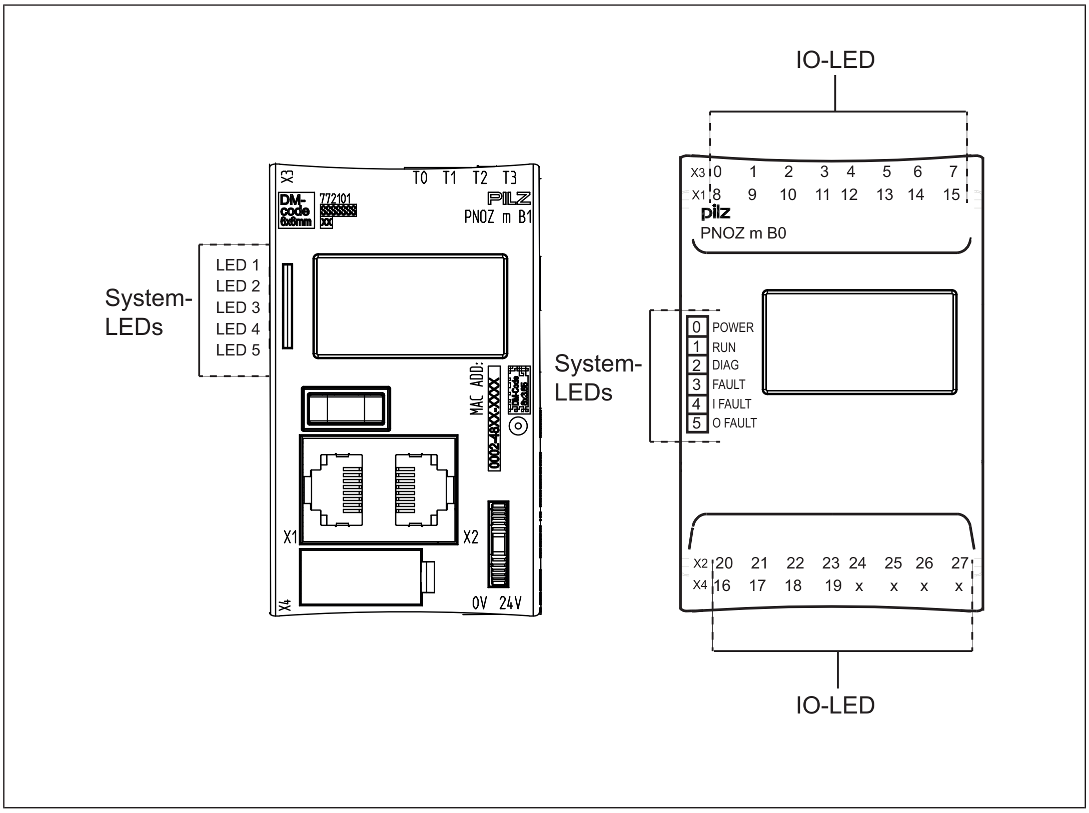
Byte	Bit 7	...	Bit 4	Bit 3	...	Bit 0
12	I/O-LED 1			I/O-LED 0		
13	I/O-LED 3			I/O-LED 2		
14	I/O-LED 5			I/O-LED 4		
15	I/O-LED 7			I/O-LED 6		
16	I/O-LED 9			I/O-LED 8		
17	I/O-LED 11			I/O-LED 10		
18	I/O-LED 13			I/O-LED 12		
19	I/O-LED 15			I/O-LED 14		
20	I/O-LED 17			I/O-LED 16		
21	I/O-LED 19			I/O-LED 18		
22	I/O-LED 21			I/O-LED 20		
23	I/O-LED 23			I/O-LED 22		
24	I/O-LED 25			I/O-LED 24		
25	I/O-LED 27			I/O-LED 26		
26	I/O-LED 29			I/O-LED 28		
27	I/O-LED 31			I/O-LED 30		
28	Reserved					
...	...					
35	Reserved					

LED status	4 Bit coding
Off	0x0
On	0xF
Flashes	0x3

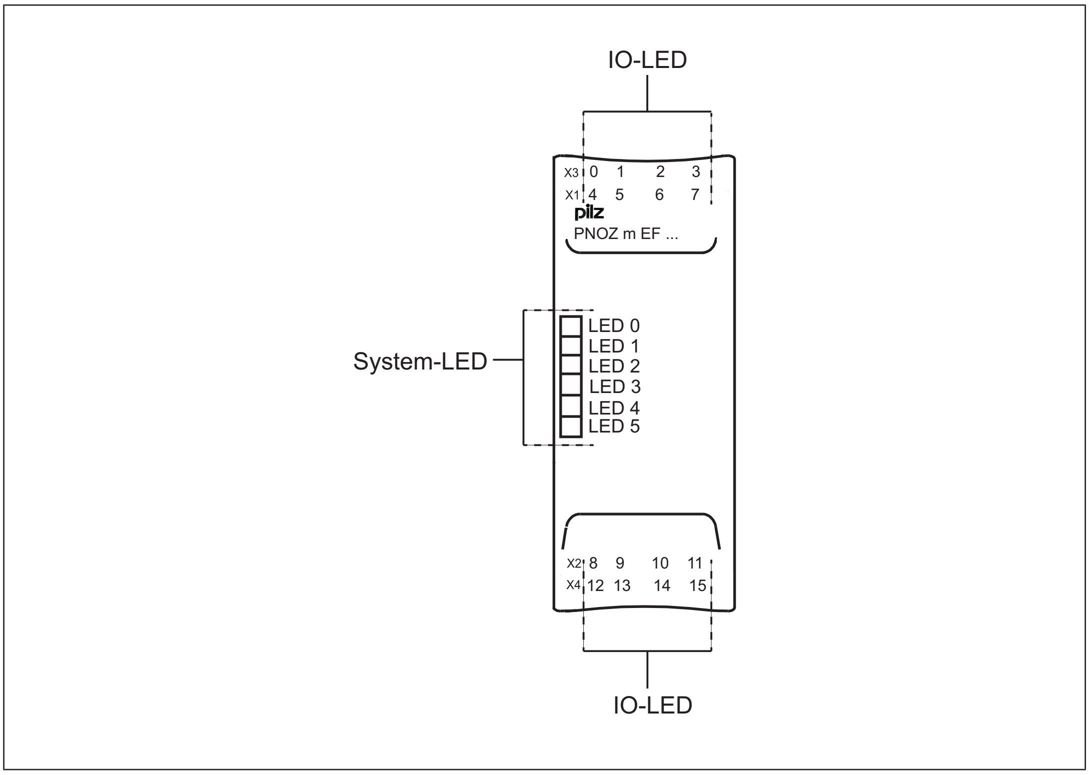
Fieldbus-specific addressing is described in the chapter entitled Process data addressing [📖 78].

8.1.5 Assignment of system and I/O-LEDs

Assignment of the I/O LEDs and system LEDs to the LEDs on the base unit:



Assignment of the I/O LEDs and system LEDs to the LEDs on the right-hand expansion modules:



8.1.6 Process data addressing

8.1.6.1 Process data addressing on base unit

Tables

Process data		Tables		
Information	**Byte**	**Table**	**Segment**	**Byte**
Inputs	0 ... 3	20	0	0 ... 3
Outputs	4 ... 7	20	0	4 ... 7
System LEDs	8 ... 10	20	0	8 ... 10
I/O LEDs	12 ... 23	20	1	0 ... 11
I/O LEDs	24 ... 27	20	2	0 ... 3

SDOs

Process data		SDOs	
Information	**Byte**	**Index**	**Subindex**
Inputs	0 ... 3	0x2001	1 ... 4
Outputs	4 ... 7	0x2001	5 ... 8
System LEDs	8 ... 10	0x2001	9 ... 11
I/O LEDs	12 ... 27	0x2001	13 ... 28

Modbus Register

Process data		Modbus Register	
Information	**Byte**	**Register**	**Coil/Discrete Input**
Inputs	0 ... 3	1024 ... 1025	16384 ... 16415
Outputs	4 ... 7	1026 ... 1027	16416 ... 16447
System LEDs	8 ... 10	1028 ... 1029	16448 ... 16471
I/O LEDs	12 ... 27	1030 ... 1037	16480 ... 16607

Profinet records

Process data		Profinet	
Information	**Byte**	**Record**	**Byte**
Inputs	0 ... 3	1	0 ... 3
Outputs	4 ... 7	1	4 ... 7
System LEDs	8 ... 10	1	8 ... 10
I/O LEDs	12 ... 27	1	12 ... 27

Ethernet/IP instances

Process data		Ethernet/IP		
Information	Byte	Instance	Attribute	Byte
Inputs	0 ... 3	1	1	0 ... 3
Outputs	4 ... 7	1	1	4 ... 7
System LEDs	8 ... 10	1	1	8 ... 10
I/O LEDs	12 ... 27	1	1	12 ... 27

8.1.6.2 Process data addressing, 1st right-hand expansion module

Tables

Process data		Tables		
Information	Byte	Table	Segment	Byte
Inputs	0 ... 3	21	0	0 ... 3
Outputs	4 ... 7	21	0	4 ... 7
System LEDs	8 ... 10	21	0	8 ... 10
I/O LEDs	12 ... 23	21	1	0 ... 11
I/O LEDs	24 ... 27	21	2	0 ... 3

SDOs

Process data		SDOs	
Information	Byte	Index	Subindex
Inputs	0 ... 3	0x2001	73 ... 76
Outputs	4 ... 7	0x2001	77 ... 80
System LEDs	8 ... 10	0x2001	81 ... 83
I/O LEDs	12 ... 27	0x2001	85 ... 100

Modbus Register

Process data		Modbus Register	
Information	Byte	Register	Coil/Discrete Input
Inputs	0 ... 3	1060 ... 1061	16960 ... 16991
Outputs	4 ... 7	1062 ... 1063	16992 ... 17023
System LEDs	8 ... 10	1064 ... 1065	17024 ... 17047
I/O LEDs	12 ... 27	1066 ... 1073	17056 ... 17183

Profinet records

Process data		Profinet	
Information	Byte	Record	Byte
Inputs	0 ... 3	3	0 ... 3
Outputs	4 ... 7	3	4 ... 7
System LEDs	8 ... 10	3	8 ... 10
I/O LEDs	12 ... 27	3	12 ... 27

Ethernet/IP instances

Process data		Ethernet/IP		
Information	Byte	Instance	Attribute	Byte
Inputs	0 ... 3	2	1	0 ... 3
Outputs	4 ... 7	2	1	4 ... 7
System LEDs	8 ... 10	2	1	8 ... 10
I/O LEDs	12 ... 27	2	1	12 ... 27

8.1.6.3 Process data addressing, 2nd right-hand expansion module

Tables

Process data		Tables		
Information	Byte	Table	Segment	Byte
Inputs	0 ... 3	21	3	0 ... 3
Outputs	4 ... 7	21	3	4 ... 7
System LEDs	8 ... 10	21	3	8 ... 10
I/O LEDs	12 ... 23	21	4	0 ... 11
I/O LEDs	24 ... 27	21	5	0 ... 3

SDOs

Process data		SDOs	
Information	Byte	Index	Subindex
Inputs	0 ... 3	0x2002	1 ... 4
Outputs	4 ... 7	0x2002	5 ... 8
System LEDs	8 ... 10	0x2002	9 ... 11
I/O LEDs	12 ... 27	0x2002	13 ... 28

Modbus Register

Process data		Modbus Register	
Information	Byte	Register	Coil/Discrete Input
Inputs	0 ... 3	1078 ... 1079	17248 ... 17279
Outputs	4 ... 7	1080 ... 1081	17280 ... 17311
System LEDs	8 ... 10	1082 ... 1083	17312 ... 17335
I/O LEDs	12 ... 27	1084 ... 1091	17344 ... 17471

Profinet records

Process data		Profinet	
Information	Byte	Record	Byte
Inputs	0 ... 3	4	0 ... 3
Outputs	4 ... 7	4	4 ... 7
System LEDs	8 ... 10	4	8 ... 10
I/O LEDs	12 ... 27	4	12 ... 27

Ethernet/IP instances

Process data		Ethernet/IP		
Information	Byte	Instance	Attribute	Byte
Inputs	0 ... 3	2	2	0 ... 3
Outputs	4 ... 7	2	2	4 ... 7
System LEDs	8 ... 10	2	2	8 ... 10
I/O LEDs	12 ... 27	2	2	12 ... 27

8.1.6.4 Process data addressing, 3rd right-hand expansion module

Tables

Process data		Tables		
Information	Byte	Table	Segment	Byte
Inputs	0 ... 3	21	6	0 ... 3
Outputs	4 ... 7	21	6	4 ... 7
System LEDs	8 ... 10	21	6	8 ... 10
I/O LEDs	12 ... 23	21	7	0 ... 11
I/O LEDs	24 ... 27	21	8	0 ... 3

SDOs

Process data		SDOs	
Information	Byte	Index	Subindex
Inputs	0 ... 3	0x2002	37 ... 40
Outputs	4 ... 7	0x2002	41 ... 44
System LEDs	8 ... 10	0x2002	45 ... 47
I/O LEDs	12 ... 27	0x2002	49 ... 64

Modbus Register

Process data		Modbus Register	
Information	Byte	Register	Coil/Discrete Input
Inputs	0 ... 3	1096 ... 1097	17536 ... 17567
Outputs	4 ... 7	1098 ... 1099	17568 ... 17599
System LEDs	8 ... 10	1100 ... 1101	17600 ... 17623
I/O LEDs	12 ... 27	1102 ... 1109	17632 ... 17759

Profinet records

Process data		Profinet	
Information	Byte	Record	Byte
Inputs	0 ... 3	5	0 ... 3
Outputs	4 ... 7	5	4 ... 7
System LEDs	8 ... 10	5	8 ... 10
I/O LEDs	12 ... 27	5	12 ... 27

Ethernet/IP instances

Process data		Ethernet/IP		
Information	Byte	Instance	Attribute	Byte
Inputs	0 ... 3	2	3	0 ... 3
Outputs	4 ... 7	2	3	4 ... 7
System LEDs	8 ... 10	2	3	8 ... 10
I/O LEDs	12 ... 27	2	3	12 ... 27

8.1.6.5 Process data addressing, 4th right-hand expansion module

Tables

Process data		Tables		
Information	Byte	Table	Segment	Byte
Inputs	0 ... 3	21	9	0 ... 3
Outputs	4 ... 7	21	9	4 ... 7
System LEDs	8 ... 10	21	9	8 ... 10
I/O LEDs	12 ... 23	21	10	0 ... 11
I/O LEDs	24 ... 27	21	11	0 ... 3

SDOs

Process data		SDOs	
Information	Byte	Index	Subindex
Inputs	0 ... 3	0x2002	73 ... 76
Outputs	4 ... 7	0x2002	77 ... 80
System LEDs	8 ... 10	0x2002	81 ... 83
I/O LEDs	12 ... 27	0x2002	85 ... 100

Modbus Register

Process data		Modbus Register	
Information	Byte	Register	Coil/Discrete Input
Inputs	0 ... 3	1114 ... 1115	17824 ... 17855
Outputs	4 ... 7	1116 ... 1117	17856 ... 17887
System LEDs	8 ... 10	1118 ... 1119	17888 ... 17911
I/O LEDs	12 ... 27	1120 ... 1127	17920 ... 18047

Profinet records

Process data		Profinet	
Information	Byte	Record	Byte
Inputs	0 ... 3	6	0 ... 3
Outputs	4 ... 7	6	4 ... 7
System LEDs	8 ... 10	6	8 ... 10
I/O LEDs	12 ... 27	6	12 ... 27

Ethernet/IP instances

Process data		Ethernet/IP		
Information	Byte	Instance	Attribute	Byte
Inputs	0 ... 3	2	4	0 ... 3
Outputs	4 ... 7	2	4	4 ... 7
System LEDs	8 ... 10	2	4	8 ... 10
I/O LEDs	12 ... 27	2	4	12 ... 27

8.1.6.6 Process data addressing, 5th right-hand expansion module

Tables

Process data		Tables		
Information	Byte	Table	Segment	Byte
Inputs	0 ... 3	21	12	0 ... 3
Outputs	4 ... 7	21	12	4 ... 7
System LEDs	8 ... 10	21	12	8 ... 10
I/O LEDs	12 ... 23	21	13	0 ... 11
I/O LEDs	24 ... 27	21	14	0 ... 3

SDOs

Process data		SDOs	
Information	Byte	Index	Subindex
Inputs	0 ... 3	0x2003	1 ... 4
Outputs	4 ... 7	0x2003	5 ... 8
System LEDs	8 ... 10	0x2003	9 ... 11
I/O LEDs	12 ... 27	0x2003	13 ... 28

Modbus Register

Process data		Modbus Register	
Information	Byte	Register	Coil/Discrete Input
Inputs	0 ... 3	1132 ... 1133	18112 ... 18143
Outputs	4 ... 7	1134 ... 1135	18144 ... 18175
System LEDs	8 ... 10	1136 ... 1137	18176 ... 18199
I/O LEDs	12 ... 27	1138 ... 1145	18208 ... 18335

Profinet records

Process data		Profinet	
Information	Byte	Record	Byte
Inputs	0 ... 3	7	0 ... 3
Outputs	4 ... 7	7	4 ... 7
System LEDs	8 ... 10	7	8 ... 10
I/O LEDs	12 ... 27	7	12 ... 27

Ethernet/IP instances

Process data		Ethernet/IP		
Information	Byte	Instance	Attribute	Byte
Inputs	0 ... 3	2	5	0 ... 3
Outputs	4 ... 7	2	5	4 ... 7
System LEDs	8 ... 10	2	5	8 ... 10
I/O LEDs	12 ... 27	2	5	12 ... 27

8.1.6.7 Process data addressing, 6th right-hand expansion module

Tables

Process data		Tables		
Information	Byte	Table	Segment	Byte
Inputs	0 ... 3	21	15	0 ... 3
Outputs	4 ... 7	21	15	4 ... 7
System LEDs	8 ... 10	21	15	8 ... 10
I/O LEDs	12 ... 23	21	16	0 ... 11
I/O LEDs	24 ... 27	21	17	0 ... 3

SDOs

Process data		SDOs	
Information	Byte	Index	Subindex
Inputs	0 ... 3	0x2003	37 ... 40
Outputs	4 ... 7	0x2003	41 ... 44
System LEDs	8 ... 10	0x2003	45 ... 47
I/O LEDs	12 ... 27	0x2003	49 ... 64

Modbus Register

Process data		Modbus Register	
Information	Byte	Register	Coil/Discrete Input
Inputs	0 ... 3	1150 ... 1151	18400 ... 18431
Outputs	4 ... 7	1152 ... 1153	18432 ... 18463
System LEDs	8 ... 10	1154 ... 1155	18464 ... 18487
I/O LEDs	12 ... 27	1156 ... 1163	18496 ... 18623

Profinet records

Process data		Profinet	
Information	Byte	Record	Byte
Inputs	0 ... 3	8	0 ... 3
Outputs	4 ... 7	8	4 ... 7
System LEDs	8 ... 10	8	8 ... 10
I/O LEDs	12 ... 27	8	12 ... 27

Ethernet/IP instances

Process data		Ethernet/IP		
Information	Byte	Instance	Attribute	Byte
Inputs	0 ... 3	2	6	0 ... 3
Outputs	4 ... 7	2	6	4 ... 7
System LEDs	8 ... 10	2	6	8 ... 10
I/O LEDs	12 ... 27	2	6	12 ... 27

8.1.6.8 Process data addressing, 7th right-hand expansion module

Only valid for PNOZ m B1!

Tables

Process data		Tables		
Information	Byte	Table	Segment	Byte
Inputs	0 ... 3	21	18	0 ... 3
Outputs	4 ... 7	21	18	4 ... 7
System LEDs	8 ... 10	21	18	8 ... 10
I/O LEDs	12 ... 23	21	19	0 ... 11
I/O LEDs	24 ... 27	21	20	0 ... 3

SDOs

Process data		SDOs	
Information	Byte	Index	Subindex
Inputs	0 ... 3	0x2003	73 ... 76
Outputs	4 ... 7	0x2003	77 ... 80
System LEDs	8 ... 10	0x2003	81 ... 83
I/O LEDs	12 ... 27	0x2003	85 ... 100

8.1.6.9 Process data addressing, 8th right-hand expansion module

Only valid for PNOZ m B1!

Tables

Process data		Tables		
Information	Byte	Table	Segment	Byte
Inputs	0 ... 3	21	21	0 ... 3
Outputs	4 ... 7	21	21	4 ... 7
System LEDs	8 ... 10	21	21	8 ... 10
I/O LEDs	12 ... 23	21	22	0 ... 11
I/O LEDs	24 ... 27	21	23	0 ... 3

SDOs

Process data		SDOs	
Information	Byte	Index	Subindex
Inputs	0 ... 3	0x2004	1 ... 4
Outputs	4 ... 7	0x2004	5 ... 8
System LEDs	8 ... 10	0x2004	9 ... 11
I/O LEDs	12 ... 27	0x2004	13 ... 28

8.1.6.10 Process data addressing, 1st left-hand expansion module

Tables

Process data		Tables		
Information	Byte	Table	Segment	Byte
Inputs	0 ... 3	22	0	0 ... 3
Outputs	4 ... 7	22	0	4 ... 7

Process data		Tables		
System LEDs	8 ... 10	22	0	8 ... 10
I/O LEDs	12 ... 23	22	1	0 ... 11
I/O LEDs	24 ... 27	22	2	0 ... 3

SDOs

Process data		SDOs	
Information	**Byte**	**Index**	**Subindex**
Inputs	0 ... 3	0x2007	1 ... 4
Outputs	4 ... 7	0x2007	5 ... 8
System LEDs	8 ... 10	0x2007	9 ... 11
I/O LEDs	12 ... 27	0x2007	13 ... 28

Modbus Register

Process data		Modbus Register	
Information	**Byte**	**Register**	**Coil/Discrete Input**
Inputs	0 ... 3	1348 ... 1349	21568 ... 21599
Outputs	4 ... 7	1350 ... 1351	21600 ... 21631
System LEDs	8 ... 10	1352 ... 1353	21632 ... 21655
I/O LEDs	12 ... 27	1354 ... 1361	21664 ... 21791

Profinet records

Process data		Profinet	
Information	**Byte**	**Record**	**Byte**
Inputs	0 ... 3	19	0 ... 3
Outputs	4 ... 7	19	4 ... 7
System LEDs	8 ... 10	19	8 ... 10
I/O LEDs	12 ... 27	19	12 ... 27

Ethernet/IP instances

Process data		Ethernet/IP		
Information	**Byte**	**Instance**	**Attribute**	**Byte**
Inputs	0 ... 3	3	1	0 ... 3
Outputs	4 ... 7	3	1	4 ... 7
System LEDs	8 ... 10	3	1	8 ... 10
I/O LEDs	12 ... 27	3	1	12 ... 27

8.1.6.11 Process data addressing, 2nd left-hand expansion module

Tables

Process data		Tables		
Information	Byte	Table	Segment	Byte
Inputs	0 ... 3	22	3	0 ... 3
Outputs	4 ... 7	22	3	4 ... 7
System LEDs	8 ... 10	22	3	8 ... 10
I/O LEDs	12 ... 23	22	4	0 ... 11
I/O LEDs	24 ... 27	22	5	0 ... 3

SDOs

Process data		SDOs	
Information	Byte	Index	Subindex
Inputs	0 ... 3	0x2007	37 ... 40
Outputs	4 ... 7	0x2007	41 ... 44
System LEDs	8 ... 10	0x2007	45 ... 47
I/O LEDs	12 ... 27	0x2007	49 ... 64

Modbus Register

Process data		Modbus Register	
Information	Byte	Register	Coil/Discrete Input
Inputs	0 ... 3	1366 ... 1367	21856 ... 21887
Outputs	4 ... 7	1368 ... 1369	21888 ... 21919
System LEDs	8 ... 10	1370 ... 1371	21920 ... 21943
I/O LEDs	12 ... 27	1372 ... 1379	21952 ... 22079

Profinet records

Process data		Profinet	
Information	Byte	Record	Byte
Inputs	0 ... 3	20	0 ... 3
Outputs	4 ... 7	20	4 ... 7
System LEDs	8 ... 10	20	8 ... 10
I/O LEDs	12 ... 27	20	12 ... 27

Ethernet/IP instances

Process data		Ethernet/IP		
Information	Byte	Instance	Attribute	Byte
Inputs	0 ... 3	3	2	0 ... 3
Outputs	4 ... 7	3	2	4 ... 7
System LEDs	8 ... 10	3	2	8 ... 10
I/O LEDs	12 ... 27	3	2	12 ... 27

8.1.6.12 Process data addressing, 3rd left-hand expansion module

Tables

Process data		Tables		
Information	Byte	Table	Segment	Byte
Inputs	0 ... 3	22	6	0 ... 3
Outputs	4 ... 7	22	6	4 ... 7
System LEDs	8 ... 10	22	6	8 ... 10
I/O LEDs	12 ... 23	22	7	0 ... 11
I/O LEDs	24 ... 27	22	8	0 ... 3

SDOs

Process data		SDOs	
Information	Byte	Index	Subindex
Inputs	0 ... 3	0x2007	73 ... 76
Outputs	4 ... 7	0x2007	77 ... 80
System LEDs	8 ... 10	0x2007	81 ... 83
I/O LEDs	12 ... 27	0x2007	85 ... 100

Modbus Register

Process data		Modbus Register	
Information	Byte	Register	Coil/Discrete Input
Inputs	0 ... 3	1384 ... 1385	22144 ... 22175
Outputs	4 ... 7	1386 ... 1387	22176 ... 22207
System LEDs	8 ... 10	1388 ... 1389	22208 ... 22231
I/O LEDs	12 ... 27	1390 ... 1397	22240 ... 22367

Profinet records

Process data		Profinet	
Information	**Byte**	**Record**	**Byte**
Inputs	0 ... 3	21	0 ... 3
Outputs	4 ... 7	21	4 ... 7
System LEDs	8 ... 10	21	8 ... 10
I/O LEDs	12 ... 27	21	12 ... 27

Ethernet/IP instances

Process data		Ethernet/IP		
Information	**Byte**	**Instance**	**Attribute**	**Byte**
Inputs	0 ... 3	3	3	0 ... 3
Outputs	4 ... 7	3	3	4 ... 7
System LEDs	8 ... 10	3	3	8 ... 10
I/O LEDs	12 ... 27	3	3	12 ... 27

8.1.6.13 Process data addressing, 4th left-hand expansion module

Tables

Process data		Tables		
Information	**Byte**	**Table**	**Segment**	**Byte**
Inputs	0 ... 3	22	9	0 ... 3
Outputs	4 ... 7	22	9	4 ... 7
System LEDs	8 ... 10	22	9	8 ... 10
I/O LEDs	12 ... 23	22	10	0 ... 11
I/O LEDs	24 ... 27	22	11	0 ... 3

SDOs

Process data		SDOs	
Information	**Byte**	**Index**	**Subindex**
Inputs	0 ... 3	0x2008	1 ... 4
Outputs	4 ... 7	0x2008	5 ... 8
System LEDs	8 ... 10	0x2008	9 ... 11
I/O LEDs	12 ... 27	0x2008	13 ... 28

Modbus Register

Process data		Modbus Register	
Information	Byte	Register	Coil/Discrete Input
Inputs	0 ... 3	1402 ... 1403	22432 ... 22463
Outputs	4 ... 7	1404 ... 1405	22464 ... 22495
System LEDs	8 ... 10	1406 ... 1407	22496 ... 22519
I/O LEDs	12 ... 27	1408 ... 1415	22528 ... 22655

Profinet records

Process data		Profinet	
Information	Byte	Record	Byte
Inputs	0 ... 3	22	0 ... 3
Outputs	4 ... 7	22	4 ... 7
System LEDs	8 ... 10	22	8 ... 10
I/O LEDs	12 ... 27	22	12 ... 27

Ethernet/IP instances

Process data		Ethernet/IP		
Information	Byte	Instance	Attribute	Byte
Inputs	0 ... 3	3	4	0 ... 3
Outputs	4 ... 7	3	4	4 ... 7
System LEDs	8 ... 10	3	4	8 ... 10
I/O LEDs	12 ... 27	3	4	12 ... 27

8.2 Process data: Fieldbus and communication module

The process data of the fieldbus and communication modules consists of 60 Bytes and contains the following information:

- State of the virtual inputs and outputs
- The status of the system LED for
  - supply voltage,
  - Diagnostics

8.2.1 State of virtual inputs i0 … i127

Byte	Bit 7	...						Bit 0
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	i31	i30	i29	i28	i27	i26	i25	i24

Byte	Bit 7	...						Bit 0
4	i39	i38	i37	i36	i35	i34	i33	i32
5	i47	i46	i45	i44	i43	i42	i41	i40
6	i55	i54	i53	i52	i51	i50	i49	i48
7	i63	i62	i61	i60	i59	i58	i57	i56
8	i71	i70	i69	i68	i67	i66	i65	i64
9	i79	i78	i77	i76	i75	i74	i73	i72
10	i87	i86	i85	i84	i83	i82	i81	i80
11	i95	i94	i93	i92	i91	i90	i89	i88
12	i103	i102	i101	i100	i99	i98	i97	i96
13	i111	i110	i109	i108	i107	i106	i105	i104
14	i119	i118	i117	i116	i115	i114	i113	i112
15	i127	i126	i125	i124	i123	i122	i121	i120
16	Reserved							
…	...							
23	Reserved							

Fieldbus-specific addressing is described in the chapter entitled Process data addressing [📖 95].

8.2.2 State of virtual outputs o0 … o127

Byte	Bit 7	...						Bit 0
24	o7	o6	o5	o4	o3	o2	o1	o0
25	o15	o14	o13	o12	o11	o10	o9	o8
26	o23	o22	o21	o20	o19	o18	o17	o16
27	o31	o30	o29	o28	o27	o26	o25	o24
28	o39	o38	o37	o36	o35	o34	o33	o32
29	o47	o46	o45	o44	o43	o42	o41	o40
30	o55	o54	o53	o52	o51	o50	o49	o48
31	o63	o62	o61	o60	o59	o58	o57	o56
32	o71	o70	o69	o68	o67	o66	o65	o64
33	o79	o78	o77	o76	o75	o74	o73	o72
34	o87	o86	o85	o84	o83	o82	o81	o80
35	o95	o94	o93	o92	o91	o90	o89	o88
36	o103	o102	o101	o100	o99	o98	o97	o96
37	o111	o110	o109	o108	o107	o106	o105	o104
38	o119	o118	o117	o116	o115	o114	o113	o112
39	o127	o126	o125	o124	o123	o122	o121	o120
40	Reserved							

Byte	Bit 7	...	Bit 0
...	...		
47	Reserved		

Fieldbus-specific addressing is described in the chapter entitled Process data addressing [📖 95].

8.2.3 Status of system LEDs

Status of the system LEDs for the fieldbus module (see also Assignment of system LEDs [📖 94])

Byte	Bit 7	...	Bit 4	Bit 3	...	Bit 0
48	System LED 1			System LED 0		
49	System LED 3			System LED 2		
50	Reserved					
...	...					
59	Reserved					

LED status	4 Bit coding
Off	0x0
On	0xF
Flashes	0x3

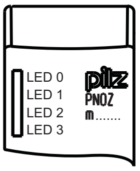
INFORMATION

The LED status "On" and "Flashes" cannot be determined exactly on some fieldbus modules.

Fieldbus-specific addressing is described in the chapter entitled Process data addressing [📖 95].

8.2.3.1 Assignment of system LEDs

Assignment of system LEDs to LEDs on the fieldbus modules:



8.2.4 Process data addressing

8.2.4.1 Process data addressing, fieldbus module

Tables

Process data		Tables		
Information	**Byte**	**Table**	**Segment**	**Byte**
Inputs	0 ... 11	23	0	0 ... 11
Inputs	12 ... 15	23	1	0 ... 3
Outputs	24 ... 35	23	2	0 ... 11
Outputs	36 ... 39	23	3	0 ... 3
System LEDs	48 ... 49	23	4	0 ... 1

SDOs

Process data		Tables	
Information	**Byte**	**Index**	**Subindex**
Inputs	0 ... 15	---	---
Outputs	24 ... 39	---	---
System LEDs	48 ... 49	0x2009	1 ... 2

Modbus Register

Process data		Modbus Register	
Information	**Byte**	**Register**	**Coil/Discrete Input (1x)**
Inputs	0 ... 15	1456 ... 1463	23296 ... 23423
Outputs	24 ... 39	1468 … 1475	23488 ... 23615
System LEDs	48 ... 49	1480	23680 ... 23695

Profinet records

Process data		Profinet	
Information	**Byte**	**Record**	**Byte**
Inputs	0 ... 15	---	---
Outputs	24 ... 39	---	---
System LEDs	48 ... 49	25	0 ... 1

Ethernet/IP instances

Process data		Ethernet/IP		
Information	**Byte**	**Instance**	**Attribute**	**Byte**
Inputs	0 ... 15	---	---	---
Outputs	24 ... 39	---	---	---
System LEDs	48 ... 49	4	1	0 ... 1

8.2.4.2 Process data addressing, communication module

Tables

Process data		Tables		
Information	**Byte**	**Table**	**Segment**	**Byte**
Inputs	0 ... 11	23	5	0 ... 11
Inputs	12 ... 15	23	6	0 ... 3
Outputs	24 ... 35	23	7	0 ... 11
Outputs	36 ... 39	23	8	0 ... 3
System LEDs	48 ... 49	23	9	0 ... 3

SDOs

Process data		SDOs	
Information	**Byte**	**Index**	**Subindex**
Inputs	0 ... 15	---	---
Outputs	24 ... 39	---	---
System LEDs	48 ... 49	---	---

Modbus Register

Process data		Modbus Register	
Information	**Byte**	**Register**	**Coil/Discrete Input (1x)**
Inputs	0 ... 15	1486 ... 1493	23776 ... 23903
Outputs	24 ... 39	1498 … 1505	23968 ... 24095
System LEDs	48 ... 49	1498 … 1505	23968 ... 24095

Profinet records

Process data		Profinet	
Information	**Byte**	**Record**	**Byte**
Inputs	0 ... 15	---	---
Outputs	24 ... 39	---	---
System LEDs	48 ... 49	---	---

Ethernet/IP instances

Process data		Ethernet/IP		
Information	Byte	Instance	Attribute	Byte
Inputs	0 ... 15	---	---	---
Outputs	24 ... 39	---	---	---
System LEDs	48 ... 49	---	---	---

8.3 Diagnostic words

The diagnostic words contain information about the elements in the PNOZmulti Configurator. The diagnostic words consist of 200 Bytes.

8.3.1 Diagnostics

Byte	About PNOZmulti Configurator
0	Diagnostic word, element ID1 e.g. A0B1 (hex): Byte 0: A0 Byte 1: B1
1	
2	Diagnostic word, element ID2 e.g. A2B3 (hex): Byte 2: A2 Byte 3: B3
3	
...	...
...	
198	Diagnostic word, element ID100 e.g. AEBF (hex): Byte 198: AE Byte 199: BF
199	

8.3.2 Diagnostic word addressing

Tables

Process data		Tables		
Information	Byte	Table	Segment	Byte
Diagnostics	0 ... 11	70	0	0 ... 11
Diagnostics	12 ... 23	70	1	0 ... 11
Diagnostics	24 ... 179	70	2 ... 14	0 ... 11
Diagnostics	180 ... 191	70	15	0 ... 11
Diagnostics	192 ... 199	70	16	0 ... 7

SDOs

Process data		SDOs	
Information	Byte	Index	Subindex
Diagnostics	0 ... 119	0x200A	1 ... 120
Diagnostics	120 ... 199	0x200B	1 ... 80

Modbus Register

Process data		Modbus Register	
Information	Byte	Register	Coil/Discrete Input (1x)
Diagnostics	0 ... 199	2048 ... 2147	32768 ... 34367

Profinet records

Process data		Profinet	
Information	Byte	Record	Byte
Diagnostics	0 ... 23	27	0 ... 23
Diagnostics	24 ... 47	28	0 ... 23
Diagnostics	48 ... 191	29 ... 35	0 ... 23
Diagnostics	192 ... 199	36	0 ... 7

Ethernet/IP instances

Process data		Ethernet/IP		
Information	Byte	Instance	Attribute	Byte
Diagnostics	0 ... 23	5	1	0 ... 23
Diagnostics	24 ... 47	5	2	0 ... 23
Diagnostics	48 ... 191	5	3 ... 8	0 ... 23
Diagnostics	192 ... 199	5	9	0 ... 7

8.4 Enable elements (only PNOZ m B0)

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

The enable elements consist of 13 Bytes.

8.4.1 Element IDs

Byte	Bit 7	...					Bit 0
0	ID 8	ID 7	ID 6	ID 5	ID 4	ID 3	ID 2	ID 1
1	ID 16	ID 15	ID 14	ID 13	ID 12	ID 11	ID 10	ID 9
2	ID 24	ID 23	ID 22	ID 21	ID 20	ID 19	ID 18	ID 17
3	ID 32	ID 31	ID 30	ID 29	ID 28	ID 27	ID 26	ID 25

Byte	Bit 7	...						Bit 0
4	ID 40	ID 39	ID 38	ID 37	ID 36	ID 35	ID 34	ID 33
5	ID 48	ID 47	ID 46	ID 45	ID 44	ID 43	ID 42	ID 41
6	ID 56	ID 55	ID 54	ID 53	ID 52	ID 51	ID 50	ID 49
7	ID 64	ID 63	ID 62	ID 61	ID 60	ID 59	ID 58	ID 57
8	ID 72	ID 71	ID 70	ID 69	ID 68	ID 67	ID 66	ID 65
9	ID 80	ID 79	ID 78	ID 77	ID 76	ID 75	ID 74	ID 73
10	ID 88	ID 87	ID 86	ID 85	ID 84	ID 83	ID 82	ID 81
11	ID 96	ID 95	ID 94	ID 93	ID 92	ID 91	ID 90	ID 89
12	Reserved				ID 100	ID 99	ID 98	ID 97

8.4.2 Enable element addressing

Tables

Process data		Tables		
Information	Byte	Table	Segment	Byte
ID	0 ... 12	71	0	0 ... 12

SDOs

Process data		SDOs	
Information	Byte	Index	Subindex
ID	0 ... 12	0x200B	81 ... 93

Modbus Register

Process data		Modbus Register	
Information	Byte	Register	Coil/Discrete Input (1x)
ID	0 ... 12	2150 ... 2156	34400 ... 34503

Profinet records

Process data		Profinet	
Information	Byte	Record	Byte
ID	0 ... 12	36	0 ... 12

Ethernet/IP instances

Process data		Ethernet/IP		
Information	Byte	Instance	Attribute	Byte
ID	0 ... 12	5	10	0 ... 12

8.5 Project data

The project data, which was also defined in the PNOZmulti Configurator, consists of 60 Bytes.

8.5.1 Check sums

Byte	Information
0	Project check sum e.g. check sum A1B2 (hex:) ▸ Byte 0: A1 ▸ Byte 1: B2
1	
2	▸ Overall check sum
3	
4	Reserved
...	...
11	Reserved

Fieldbus-specific addressing is described in the chapter entitled Project data addressing [📖 101].

8.5.2 Date

Byte	Information
12	Project creation date: Day and month (DDMM), year (YYYY) e.g. creation date: 28.11.2003 ▸ Byte 12: 1C ▸ Byte 13: 0B ▸ Byte 14: 14 ▸ Byte 15: 03
13	
14	
15	
16	Day and month (DDMM), e.g. 28.11. ▸ Byte 16: 1C ▸ Byte 17: 0B
17	
18	Year (JJJJ), e.g. 2003 ▸ Byte 18: 14 ▸ Byte 19: 03
19	
20	Time (HH:MM), e.g. 14 hours, 25 minutes ▸ Byte 20: 0E ▸ Byte 21: 19
21	
22	Time zone 1 ▸ Byte 22: 0 ▸ Byte 23: 1
23	

Fieldbus-specific addressing is described in the chapter entitled Project data addressing [📖 101].

8.5.3 Project name

Project name defined in the PNOZmulti Configurator under "Enter project data". The name is stored in UNICODE format (2 Bytes contain the Hex code of one UNICODE character).

Byte	Information
24	1st character of project name
25	▶
26	2nd character of project name
27	▶
...	...
...	
54	16th character of project name
55	▶
56	End of character string (identified by "FFFF")
57	▶
58	Reserved
59	Reserved

Fieldbus-specific addressing is described in the chapter entitled Project data addressing [📖 101].

8.5.4 Project data addressing

Tables

Process data		Tables		
Information	Byte	Table	Segment	Byte
Check sums	0 ... 3	80	0	0 ... 3
Date	12 ... 23	80	1	0 ... 11
Project name	24 ... 35	80	2	0 ... 11
Project name	36 ... 47	80	3	0 ... 11
Project name	48 ... 57	80	4	0 ... 9

SDOs

Process data		SDOs	
Information	Byte	Index	Subindex
Check sums	0 ... 3	0x200D	13 ... 16
Date	12 ... 23	0x200D	25 ... 36

Process data		SDOs	
Project name	24 ... 57	0x200D	37 ... 70

Modbus Register

Process data		Modbus Register	
Information	Byte	Register	Coil/Discrete Input (1x)
Check sums	0 ... 3	4096 ... 4097	---
Date	12 ... 23	4102 ... 4107	---
Project name	24 ... 57	4108 ... 4124	---

Profinet records

Process data		Profinet	
Information	Byte	Record	Byte
Check sums	0 ... 3	37	0 ... 3
Date	12 ... 23	37	12 ... 23
Project name	24 ... 57	38	0 ... 33

Ethernet/IP instances

Process data		Ethernet/IP		
Information	Byte	Instance	Attribute	Byte
Check sums	0 ... 3	6	1	0 ... 3
Date	12 ... 23	6	1	12 ... 23
Project name	24 ... 57	6	2	0 ... 33

8.6 Device data

The device data consists of 36 Bytes.

8.6.1 Product

Byte	Information
0	Product number, e.g. 772 100: 000BC804 (hex) ▶ Byte 0: 00 ▶ Byte 1: 0B, ▶ Byte 2: C8, ▶ Byte 3: 04
1	
2	
3	
4	Serial number, e.g. 123 456: 0001E240 (hex) ▶ Byte 4: 00 ▶ Byte 5: 01 ▶ Byte 6: E2, ▶ Byte 7: 40
5	
6	
7	
8	Device type, e.g. PNOZ m B0: 0060 (hex) ▶ Byte 8: 00 ▶ Byte 9: 60 ▶ Device type, e.g. PNOZ m B1: 0061 (hex) ▶ Byte 8: 00 ▶ Byte 9: 61
9	
10	Unit version 20: 14 (hex) ▶ Byte 10: 00 ▶ Byte 11: 14
11	

Fieldbus-specific addressing is described in the chapter entitled Device data addressing [📖 104].

8.6.2 Firmware

Byte	Information
12	Firmware Version A
13	
14	Firmware Version B
15	
16	Check sum, Firmware Version A
17	
18	Check sum, Firmware Version B
19	
20	Reserved
...	...

Byte	Information
23	Reserved

Fieldbus-specific addressing is described in the chapter entitled Device data addressing [📖 104].

8.6.3 Operating hours

The operating hours counter can only be requested for the base unit.

Byte	Information
24	Operating hours counter, e.g. 106786 - Byte 24: Reserved - Byte 25: 01 x 10000 (hex) - Byte 26: 22 x 100 (hex) - Byte 27: 22 x 1 (hex)
25	
26	
27	
28	Reserved
...	...
35	Reserved

Fieldbus-specific addressing is described in the chapter entitled Device data addressing [📖 104].

8.6.4 Device data addressing

8.6.4.1 Device data addressing on base unit

Tables

Device data		Tables		
Information	Byte	Table	Segment	Byte
Product	0 ... 11	90	0	0 ... 11
Firmware	12 ... 19	90	1	0 ... 7
Operating hours	24 ... 27	90	2	0 ... 3

SDOs

Device data		SDOs	
Information	Byte	Index	Subindex
Product	0 ... 11	0x200F	1 ... 12
Firmware	12 ... 19	0x200F	13 ... 20
Operating hours	24 ... 27	0x200F	25 ... 28

Modbus Register

Device data		Modbus Register	
Information	**Byte**	**Register (3x9**	**Coil/Discrete Input**
Product	0 ... 11	8192 ... 8197	---
Firmware	12 ... 19	8198 ... 8201	---
Operating hours	24 ... 27	8204 ... 8209	---

Profinet records

Device data		Profinet	
Information	**Byte**	**Record**	**Byte**
Product	0 ... 11	40	0 ... 11
Firmware	12 ... 19	40	12 ... 19
Operating hours	24 ... 27	40	24 ... 27

Ethernet/IP instances

Device data		Ethernet/IP		
Information	**Byte**	**Instance**	**Attribute**	**Byte**
Product	0 ... 11	7	1	0 ... 11
Firmware	12 ... 19	7	1	12 ... 19
Operating hours	24 ... 27	7	1	24 ... 27

8.6.4.2 Device data addressing, 1st right-hand expansion module

Tables

Device data		Tables		
Information	**Byte**	**Table**	**Segment**	**Byte**
Product	0 ... 11	91	0	0 ... 11
Firmware	12 ... 19	91	1	0 ... 7

SDOs

Device data		SDOs	
Information	**Byte**	**Index**	**Subindex**
Product	0 ... 11	0x200F	73 ... 84
Firmware	12 ... 19	0x200F	85 ... 92

Modbus Register

Device data		Modbus Register	
Information	Byte	Register (3x9	Coil/Discrete Input
Product	0 ... 11	8228 ... 8233	---
Firmware	12 ... 19	8234 ... 8237	---

Profinet records

Device data		Profinet	
Information	Byte	Record	Byte
Product	0 ... 11	42	0 ... 11
Firmware	12 ... 19	42	12 ... 19

Ethernet/IP instances

Device data		Ethernet/IP		
Information	Byte	Instance	Attribute	Byte
Product	0 ... 11	8	1	0 ... 11
Firmware	12 ... 19	8	1	12 ... 19

8.6.4.3 Device data addressing, 2nd right-hand expansion module

Tables

Device data		Tables		
Information	Byte	Table	Segment	Byte
Product	0 ... 11	91	3	0 ... 11
Firmware	12 ... 19	91	4	0 ... 7

SDOs

Device data		SDOs	
Information	Byte	Index	Subindex
Product	0 ... 11	0x2010	1 ... 12
Firmware	12 ... 19	0x2010	13 ... 20

Modbus Register

Device data		Modbus Register	
Information	Byte	Register	Coil/Discrete Input
Product	0 ... 11	8246 ... 8251	---
Firmware	12 ... 19	8252 ... 8255	---

Profinet records

Device data		Profinet	
Information	Byte	Record	Byte
Product	0 ... 11	43	0 ... 11
Firmware	12 ... 19	43	12 ... 19

Ethernet/IP instances

Device data		Ethernet/IP		
Information	Byte	Instance	Attribute	Byte
Product	0 ... 11	8	2	0 ... 11
Firmware	12 ... 19	8	2	12 ... 19

8.6.4.4 Device data addressing, 3rd right-hand expansion module

Tables

Device data		Tables		
Information	Byte	Table	Segment	Byte
Product	0 ... 11	91	6	0 ... 11
Firmware	12 ... 19	91	7	0 ... 7

SDOs

Device data		SDOs	
Information	Byte	Index	Subindex
Product	0 ... 11	0x2010	37 ... 48
Firmware	12 ... 19	0x2010	49 ... 56

Modbus Register

Device data		Modbus Register	
Information	Byte	Register	Coil/Discrete Input
Product	0 ... 11	8264 ... 8269	---
Firmware	12 ... 19	8270 ... 8273	---

Profinet records

Device data		Profinet	
Information	Byte	Record	Byte
Product	0 ... 11	44	0 ... 11
Firmware	12 ... 19	44	12 ... 19

Ethernet/IP instances

Device data		Ethernet/IP		
Information	Byte	Instance	Attribute	Byte
Product	0 ... 11	8	3	0 ... 11
Firmware	12 ... 19	8	3	12 ... 19

8.6.4.5 Device data addressing, 4th right-hand expansion module

Tables

Device data		Tables		
Information	Byte	Table	Segment	Byte
Product	0 ... 11	91	9	0 ... 11
Firmware	12 ... 19	91	10	0 ... 7

SDOs

Device data		SDOs	
Information	Byte	Index	Subindex
Product	0 ... 11	0x2010	73 ... 84
Firmware	12 ... 19	0x2010	85 ... 92

Modbus Register

Device data		Modbus Register	
Information	Byte	Register	Coil/Discrete Input
Product	0 ... 11	8282 ... 8287	---
Firmware	12 ... 19	8288 ... 8291	---

Profinet records

Device data		Profinet	
Information	Byte	Record	Byte
Product	0 ... 11	45	0 ... 11
Firmware	12 ... 19	45	12 ... 19

Ethernet/IP instances

Device data		Ethernet/IP		
Information	Byte	Instance	Attribute	Byte
Product	0 ... 11	8	4	0 ... 11
Firmware	12 ... 19	8	4	12 ... 19

8.6.4.6 Device data addressing, 5th right-hand expansion module

Tables

Device data		Tables		
Information	Byte	Table	Segment	Byte
Product	0 ... 11	91	12	0 ... 11
Firmware	12 ... 19	91	13	0 ... 7

SDOs

Device data		SDOs	
Information	Byte	Index	Subindex
Product	0 ... 11	0x2011	1 ... 12
Firmware	12 ... 19	0x2011	13 ... 20

Modbus Register

Device data		Modbus Register	
Information	Byte	Register	Coil/Discrete Input
Product	0 ... 11	8300 ... 8305	---
Firmware	12 ... 19	8306 ... 8309	---

Profinet records

Device data		Profinet	
Information	Byte	Record	Byte
Product	0 ... 11	46	0 ... 11
Firmware	12 ... 19	46	12 ... 19

Ethernet/IP instances

Device data		Ethernet/IP		
Information	Byte	Instance	Attribute	Byte
Product	0 ... 11	8	5	0 ... 11
Firmware	12 ... 19	8	5	12 ... 19

8.6.4.7 Device data addressing, 6th right-hand expansion module

Tables

Device data		Tables		
Information	**Byte**	**Table**	**Segment**	**Byte**
Product	0 ... 11	91	15	0 ... 11
Firmware	12 ... 19	91	16	0 ... 7

SDOs

Device data		SDOs	
Information	**Byte**	**Index**	**Subindex**
Product	0 ... 11	0x2011	37 ... 48
Firmware	12 ... 19	0x2011	49 ... 56

Modbus Register

Device data		Modbus Register	
Information	**Byte**	**Register**	**Coil/Discrete Input**
Product	0 ... 11	8318 ... 8323	---
Firmware	12 ... 19	8324 ... 8327	---

Profinet records

Device data		Profinet	
Information	**Byte**	**Record**	**Byte**
Product	0 ... 11	47	0 ... 11
Firmware	12 ... 19	47	12 ... 19

Ethernet/IP instances

Device data		Ethernet/IP		
Information	**Byte**	**Instance**	**Attribute**	**Byte**
Product	0 ... 11	8	6	0 ... 11
Firmware	12 ... 19	8	6	12 ... 19

8.6.4.8 Device data addressing, 7th right-hand expansion module

Tables

Device data		Tables		
Information	**Byte**	**Table**	**Segment**	**Byte**
Product	0 ... 11	91	18	0 ... 11
Firmware	12 ... 19	91	19	0 ... 7

SDOs

Device data		SDOs	
Information	**Byte**	**Index**	**Subindex**
Product	0 ... 11	0x2011	73 ... 84
Firmware	12 ... 19	0x2011	85 ... 92

Modbus Register

Device data		Modbus Register	
Information	**Byte**	**Register**	**Coil/Discrete Input**
Product	0 ... 11	8336 ... 8341	---
Firmware	12 ... 19	8342 ... 8345	---

Profinet records

Device data		Profinet	
Information	**Byte**	**Record**	**Byte**
Product	0 ... 11	48	0 ... 11
Firmware	12 ... 19	48	12 ... 19

Ethernet/IP instances

Device data		Ethernet/IP		
Information	**Byte**	**Instance**	**Attribute**	**Byte**
Product	0 ... 11	8	7	0 ... 11
Firmware	12 ... 19	8	7	12 ... 19

8.6.4.9 Device data addressing, 8th right-hand expansion module

Tables

Device data		Tables		
Information	Byte	Table	Segment	Byte
Product	0 ... 11	91	21	0 ... 11
Firmware	12 ... 19	91	22	0 ... 7

SDOs

Device data		SDOs	
Information	Byte	Index	Subindex
Product	0 ... 11	0x2012	1 ... 12
Firmware	12 ... 19	0x2012	13 ... 20

Modbus Register

Device data		Modbus Register	
Information	Byte	Register	Coil/Discrete Input
Product	0 ... 11	8354 ... 8359	---
Firmware	12 ... 19	8360 ... 8363	---

Profinet records

Device data		Profinet	
Information	Byte	Record	Byte
Product	0 ... 11	49	0 ... 11
Firmware	12 ... 19	49	12 ... 19

Ethernet/IP instances

Device data		Ethernet/IP		
Information	Byte	Instance	Attribute	Byte
Product	0 ... 11	8	8	0 ... 11
Firmware	12 ... 19	8	8	12 ... 19

8.6.4.10 Device data addressing, 1st left-hand expansion module

Tables

Device data		Tables		
Information	Byte	Table	Segment	Byte
Product	0 ... 11	92	0	0 ... 11
Firmware	12 ... 19	92	1	0 ... 7

SDOs

Device data		SDOs	
Information	Byte	Index	Subindex
Product	0 ... 11	0x2015	1 ... 12
Firmware	12 ... 19	0x2015	13 ... 20

Modbus Register

Device data		Modbus Register	
Information	Byte	Register	Coil/Discrete Input
Product	0 ... 11	8516 ... 8521	---
Firmware	12 ... 19	8522 ... 8525	---

Profinet records

Device data		Profinet	
Information	Byte	Record	Byte
Product	0 ... 11	58	0 ... 11
Firmware	12 ... 19	58	12 ... 19

Ethernet/IP instances

Device data		Ethernet/IP		
Information	Byte	Instance	Attribute	Byte
Product	0 ... 11	9	1	0 ... 11
Firmware	12 ... 19	9	1	12 ... 19

8.6.4.11 Device data addressing, 2nd left-hand expansion module

Tables

Device data		Tables		
Information	**Byte**	**Table**	**Segment**	**Byte**
Product	0 ... 11	92	3	0 ... 11
Firmware	12 ... 19	92	4	0 ... 7

SDOs

Device data		SDOs	
Information	**Byte**	**Index**	**Subindex**
Product	0 ... 11	0x2015	37 ... 48
Firmware	12 ... 19	0x2015	49 ... 56

Modbus Register

Device data		Modbus Register	
Information	**Byte**	**Register**	**Coil/Discrete Input**
Product	0 ... 11	8534 ... 8539	---
Firmware	12 ... 19	8540 ... 8543	---

Profinet records

Device data		Profinet	
Information	**Byte**	**Record**	**Byte**
Product	0 ... 11	59	0 ... 11
Firmware	12 ... 19	59	12 ... 19

Ethernet/IP instances

Device data		Ethernet/IP		
Information	**Byte**	**Instance**	**Attribute**	**Byte**
Product	0 ... 11	9	2	0 ... 11
Firmware	12 ... 19	9	2	12 ... 19

8.6.4.12 Device data addressing, 3rd left-hand expansion module

Tables

Device data		Tables		
Information	Byte	Table	Segment	Byte
Product	0 ... 11	92	6	0 ... 11
Firmware	12 ... 19	92	7	0 ... 7

SDOs

Device data		SDOs	
Information	Byte	Index	Subindex
Product	0 ... 11	0x2015	73 ... 84
Firmware	12 ... 19	0x2015	85 ... 92

Modbus Register

Device data		Modbus Register	
Information	Byte	Register	Coil/Discrete Input
Product	0 ... 11	8552 ... 8557	---
Firmware	12 ... 19	8558 ... 8561	---

Profinet records

Device data		Profinet	
Information	Byte	Record	Byte
Product	0 ... 11	60	0 ... 11
Firmware	12 ... 19	60	12 ... 19

Ethernet/IP instances

Device data		Ethernet/IP		
Information	Byte	Instance	Attribute	Byte
Product	0 ... 11	9	3	0 ... 11
Firmware	12 ... 19	9	3	12 ... 19

8.6.4.13 Device data addressing, 4th left-hand expansion module

Tables

Device data		Tables		
Information	Byte	Table	Segment	Byte
Product	0 ... 11	92	9	0 ... 11
Firmware	12 ... 19	92	10	0 ... 7

SDOs

Device data		SDOs	
Information	Byte	Index	Subindex
Product	0 ... 11	0x2016	1 ... 12
Firmware	12 ... 19	0x2016	13 ... 20

Modbus Register

Device data		Modbus Register	
Information	Byte	Register	Coil/Discrete Input
Product	0 ... 11	8570 ... 8575	---
Firmware	12 ... 19	8576 ... 8579	---

Profinet records

Device data		Profinet	
Information	Byte	Record	Byte
Product	0 ... 11	61	0 ... 11
Firmware	12 ... 19	61	12 ... 19

Ethernet/IP instances

Device data		Ethernet/IP		
Information	Byte	Instance	Attribute	Byte
Product	0 ... 11	9	4	0 ... 11
Firmware	12 ... 19	9	4	12 ... 19

8.6.4.14 Device data addressing, fieldbus module

Device types:

- PROFIBUS: 0x0001
- CANopen: 0x0020
- Ethernet/IP: 0x0083
- PROFINET: 0x0085
- EtherCAT: 0x0087

- Powerlink: 0x0098
- CC-Link: 0x0090

INFORMATION

No serial number and no check sum are currently entered for the fieldbus modules.

Tables

Device data		Tables		
Information	**Byte**	**Table**	**Segment**	**Byte**
Product	0 ... 11	93	0	0 ... 11
Firmware	12 ... 19	93	1	0 ... 7

SDOs

Device data		SDOs	
Information	**Byte**	**Index**	**Subindex**
Product	0 ... 11	0x2017	1 ... 12
Firmware	12 ... 19	0x2017	13 ... 20

Modbus Register

Device data		Modbus Register	
Information	**Byte**	**Register**	**Coil/Discrete Input**
Product	0 ... 11	8624 ... 8629	---
Firmware	12 ... 19	8630 ... 8633	---

Profinet records

Device data		Profinet	
Information	**Byte**	**Record**	**Byte**
Product	0 ... 11	64	0 ... 11
Firmware	12 ... 19	64	12 ... 19

Ethernet/IP instances

Device data		Ethernet/IP		
Information	**Byte**	**Instance**	**Attribute**	**Byte**
Product	0 ... 11	10	1	0 ... 11
Firmware	12 ... 19	10	1	12 ... 19

8.6.4.15 Device data addressing, communication module

Tables

Device data		Tables		
Information	**Byte**	**Table**	**Segment**	**Byte**
Product	0 ... 11	93	3	0 ... 11
Firmware	12 ... 19	93	4	0 ... 7

SDOs

Device data		SDOs	
Information	**Byte**	**Index**	**Subindex**
Product	0 ... 11	0x2017	37 ... 48
Firmware	12 ... 19	0x2017	49 ... 56

Modbus Register

Device data		Modbus Register	
Information	**Byte**	**Register**	**Coil/Discrete Input**
Product	0 ... 11	8642 ... 8647	---
Firmware	12 ... 19	8648 ... 8651	---

Profinet records

Device data		Profinet	
Information	**Byte**	**Record**	**Byte**
Product	0 ... 11	65	0 ... 11
Firmware	12 ... 19	65	12 ... 19

Ethernet/IP instances

Device data		Ethernet/IP		
Information	**Byte**	**Instance**	**Attribute**	**Byte**
Product	0 ... 11	10	2	0 ... 11
Firmware	12 ... 19	10	2	12 ... 19

Support

Technical support is available from Pilz round the clock.

Americas
Brazil
+55 11 97569-2804
Canada
+1 888-315-PILZ (315-7459)
Mexico
+52 55 5572 1300
USA (toll-free)
+1 877-PILZUSA (745-9872)

Asia
China
+86 21 60880878-216
Japan
+81 45 471-2281
South Korea
+82 31 450 0680

Australia
+61 3 95446300

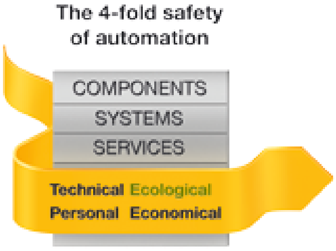
Europe
Austria
+43 1 7986263-0
Belgium, Luxembourg
+32 9 3217575
France
+33 3 88104000
Germany
+49 711 3409-444
Ireland
+353 21 4804983
Italy
+39 0362 1826711
Scandinavia
+45 74436332
Spain
+34 938497433
Switzerland
+41 62 88979-30
The Netherlands
+31 347 320477
Turkey
+90 216 5775552
United Kingdom
+44 1536 462203

You can reach our international hotline on:
+49 711 3409-444
support@pilz.com

Pilz develops environmentally-friendly products using ecological materials and energy-saving technologies. Offices and production facilities are ecologically designed, environmentally-aware and energy-saving. So Pilz offers sustainability, plus the security of using energy-efficient products and environmentally-friendly solutions.

Energy saving by Pilz

BLUECOMPETENCE Alliance Member
Partner of the Engineering Industry Sustainability Initiative



pilz the spirit of safety
Partner of:
The Best of German Engineering

DIN EN ISO 9001 DQS CERTIFIED QUALITY SYSTEM REG. NO. 2462

Pilz GmbH & Co. KG
Felix-Wankel-Straße 2
73760 Ostfildern, Germany
Tel.: +49 711 3409-0
Fax: +49 711 3409-133
info@pilz.com
www.pilz.com

PILZ
THE SPIRIT OF SAFETY

1002971-EN-04, 2016-01 Printed in Germany
© Pilz GmbH & Co. KG, 2015

CMSE®, InduraNET p®, PAS4000®, PAScal®, PASconfig®, Pilz®, PIT®, PLID®, PMCprimo®, PMCprotego®, PMCtendo®, PMD®, PMI®, PNOZ®, Primo®, PSEN®, PSS®, PVIS®, SafetyBUS
SafetyEYE®, SafetyNET p®, THE SPIRIT OF SAFETY® are registered and protected trademarks of Pilz GmbH & Co. KG in some countries. We would point out that product features ma
from the details stated in this document, depending on the status at the time of publication and the scope of the equipment. We accept no responsibility for the validity, accuracy
and entirety of the text and graphics presented in this information. Please contact our Technical Support if you have any questions.