AUTOMATION

User manual

UM EN
IL PB BK DI8 DO4/EF-PAC

Order No. –

Diagnostic and firmware functions
of the IL PB BK DI8 DO4/EF-PAC bus coupler

PHOENIX CONTACT
INSPIRING INNOVATIONS

AUTOMATION

User manual

Diagnostic and firmware functions of the IL PB BK DI8 DO4/EF-PAC bus coupler

12/2009

Designation: UM EN IL PB BK DI8 DO4/EF-PAC

Revision: 01

Order No.: –

This user manual is valid for:

Designation	Order No.
IL PB BK DI8 DO4/EF-PAC	2692322

Please observe the following notes

In order to ensure the safe use of the product described, you have to read and understand this manual. The following notes provide information on how to use this manual.

User group of this manual

The use of products described in this manual is oriented exclusively to

Phoenix Contact accepts no liability for erroneous handling or damage to products from Phoenix Contact or third-party products resulting from disregard of information contained in this manual.

Explanation of symbols used and signal words

This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

DANGER
This indicates a hazardous situation which, if not avoided, will result in death or serious injury.

WARNING
This indicates a hazardous situation which, if not avoided, could result in death or serious injury.

CAUTION
This indicates a hazardous situation which, if not avoided, could result in minor or moderate injury.

The following types of messages provide information about possible property damage and general information concerning proper operation and ease-of-use.

NOTE
This symbol and the accompanying text alerts the reader to a situation which may cause damage or malfunction to the device, either hardware or software, or surrounding property.

This symbol and the accompanying text provides additional information to the reader. It is also used as a reference to other sources of information (manuals, data sheets, literature) on the subject matter, product, etc.

General terms and conditions of use for technical documentation

Phoenix Contact reserves the right to alter, correct, and/or improve the technical documentation and the products described in the technical documentation at its own discretion and without giving prior notice, insofar as this is reasonable for the user. The same applies to any technical changes that serve the purpose of technical progress.

The receipt of technical documentation (in particular data sheets, installation instructions, manuals, etc.) does not constitute any further duty on the part of Phoenix Contact to furnish information on alterations to products and/or technical documentation. Any other agreement shall only apply if expressly confirmed in writing by Phoenix Contact. Please note that the supplied documentation is product-specific documentation only and that you are responsible for checking the suitability and intended use of the products in your specific application, in particular with regard to observing the applicable standards and regulations. Although Phoenix Contact makes every effort to ensure that the information content is accurate, up-to-date, and state-of-the-art, technical inaccuracies and/or printing errors in the information cannot be ruled out. Phoenix Contact does not offer any guarantees as to the reliability, accuracy or completeness of the information. All information made available in the technical data is supplied without any accompanying guarantee, whether expressly mentioned, implied or tacitly assumed. This information does not include any guarantees regarding quality, does not describe any fair marketable quality, and does not make any claims as to quality guarantees or guarantees regarding the suitability for a special purpose.

Phoenix Contact accepts no liability or responsibility for errors or omissions in the content of the technical documentation (in particular data sheets, installation instructions, manuals, etc.).

The aforementioned limitations of liability and exemptions from liability do not apply, in so far as liability must be assumed, e.g., according to product liability law, in cases of premeditation, gross negligence, on account of loss of life, physical injury or damage to health or on account of the violation of important contractual obligations. Claims for damages for the violation of important contractual obligations are, however, limited to contract-typical, predictable damages, provided there is no premeditation or gross negligence, or that liability is assumed on account of loss of life, physical injury or damage to health. This ruling does not imply a change in the burden of proof to the detriment of the user.

Statement of legal authority

How to contact us

Internet

Up-to-date information on Phoenix Contact products and our Terms and Conditions can be found on the Internet at:

www.phoenixcontact.com.

Make sure you always use the latest documentation.
It can be downloaded at:

www.phoenixcontact.net/download.

Subsidiaries

If there are any problems that cannot be solved using the documentation, please contact your Phoenix Contact subsidiary.
Subsidiary contact information is available at www.phoenixcontact.com.

Published by

PHOENIX CONTACT GmbH & Co. KG
Flachsmarktstraße 8
32825 Blomberg
Germany
Phone +49 - (0) 52 35 - 3-00
Fax +49 - (0) 52 35 - 3-4 12 00

Should you have any suggestions or recommendations for improvement of the contents and layout of our manuals, please send your comments to

tecdoc@phoenixcontact.com.

Table of contents

1 The PROFIBUS bus coupler

PROFIBUS bus coupler

The PROFIBUS bus coupler configures the station and manages data exchange with a PROFIBUS master. It also provides the power supply for the connected Inline terminals. PROFIsafe I/O modules can also be used.

The PROFIBUS bus coupler can be ordered using Order No. 2692322. Connectors, labeling fields, and an end plate are supplied as standard.

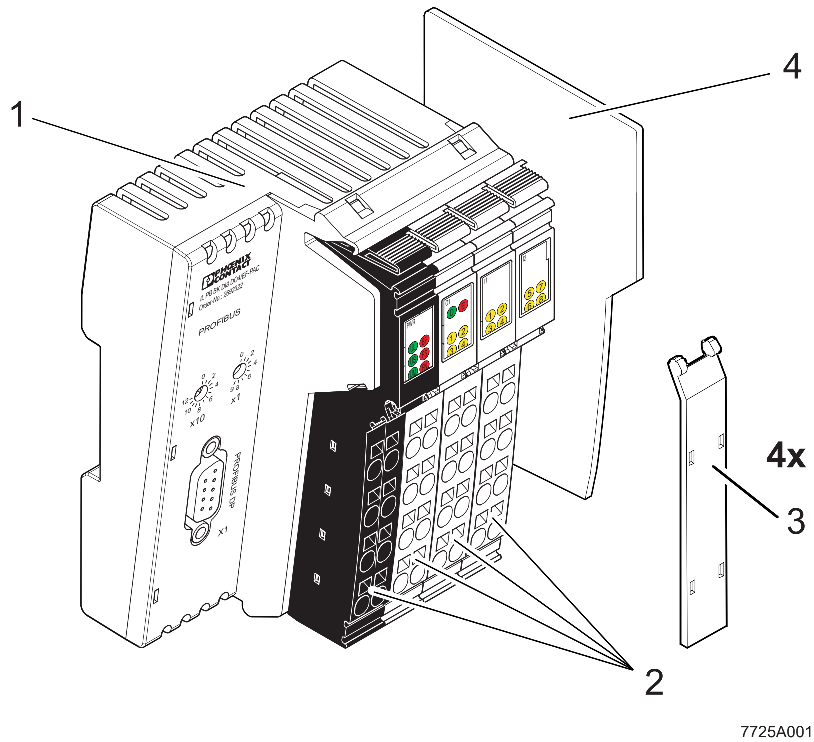


Figure 1-1 The IL PB BK DI8 DO4/EF-PAC PROFIBUS bus coupler

Scope of supply

- PROFIBUS bus coupler (1)
- Connectors (2)
- Labeling field (3)
- End plate (4)

Features

- PROFIBUS connection via 9-pos. D-SUB female connector
- Interface physics RS-485 for PROFIBUS
- Electrical isolation of PROFIBUS interface and logic
- DP/V1 for Class 1 and Class 2 masters
- Data transmission speed of 9.6 kbps up to 12 Mbps (automatic detection)
- Rotary encoding switches for setting the PROFIBUS address
- Supported PROFIBUS addresses 1 to 126
- Up to 16 PCP devices can be connected
- Device description using GSD file
- I & M functions
- Eight digital inputs
- Four digital outputs
- Diagnostic and status LEDs
- Automatic baud rate detection on the local bus (500 kbps or 20 Mbps)

Additional features of EF version:

- Approved for PROFIsafe
- IO-Link call (firmware 2.0 or later)

When using the bus coupler in a PROFIsafe system, please refer to the documentation for the safety terminals used (see "PROFIsafe application notes" on page 5-1).

For additional information about the bus coupler, please refer to the following documents:

Data sheet: DB EN IL PB BK DI8 DO4/EF-PAC

User manual: IL SYS INST UM E, Order No. 2698737

They can be downloaded at:

www.phoenixcontact.net/catalog

Configuration and start-up with the Step7 SIMATIC Manager are described in detail in the UM EN IL PB BK DP/V1 user manual.

2 Diagnostics

The diagnostic concept for the bus coupler consists of two components. Firstly, diagnostic data is supplied to the control system via PROFIBUS in the form of diagnostic bytes. Secondly, local diagnostics are available, whereby the error type is indicated by the specific flashing codes of the LEDs on the bus coupler.

2.1 Activating/deactivating the diagnostic formats

The diagnostic format can be set as a parameter on the terminal. You can select either "IL PB BK DI8 DO4 Format", "Status-PDU Diagnostics" or "ID-Specific Diagnostics". By default, "IL PB BK DI8 DO4 Format" is activated. The"Status-PDU Diagnostics" format must be set for diagnostics of PROFIsafe I/O modules.

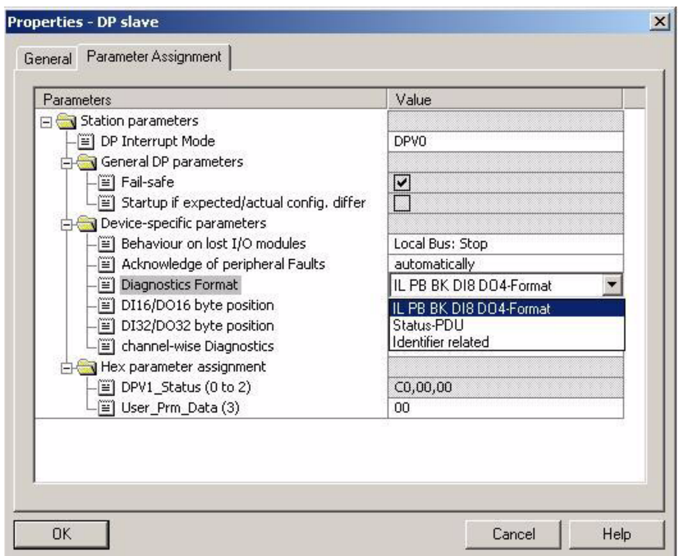


Figure 2-1 Diagnostic formats dialog box

2.1.1 Diagnostics in the IL PB BK DI8 DO4 format

This diagnostic format consists of the following blocks:

1 PROFIBUS standard diagnostics
2 ID-specific diagnostics
3 Status diagnostics (terminal status)
4 Channel-specific diagnostics
5 Revision diagnostics (manufacturer-specific)
6 Status-PDU (manufacturer-specific)

> From block 2 onwards the diagnostic telegram is dynamic. The individual number of bytes in a block depends on the station structure.

The header bytes are used to distinguish between all blocks.

Table 2-1 IL PB BK DI8 DO4-PAC format diagnostics

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Meaning
Block 1									
1	X	X	X	X	X	X	X	X	Station status 1
2	X	X	X	X	X	X	X	X	Station status 2
3	X	X	X	X	X	X	X	X	Station status 3
4	X	X	X	X	X	X	X	X	PROFIBUS master address
5	0	0	0	0	1	0	1	1	Manufacturer ID high byte ($0B_{hex}$)
6	0	1	0	1	0	0	0	0	Manufacturer ID low byte (50_{hex})
Block 2									
7	0	1	Number of local bus devices						ID-specific diagnostics (header)
8	LD 8	LD 7	LD 6	LD 5	LD 4	LD 3	LD 2	LD 1	Local bus devices 1 to 8
9	LD 16	LD 15	LD 14	LD 13	LD 12	LD 11	LD 10	LD 9	Local bus devices 9 to 16
...	...	...	...	...	...	...	...	...	...
15	0	LD 63	LD 62	LD 61	LD 60	LD 59	LD 58	LD 57	Local bus devices 57 to 63

Table 2-1 IL PB BK DI8 DO4-PAC format diagnostics

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Meaning
Block 3									
16	0	0	Number of local bus devices						Status (header)
17	1	0	0	0	0	0	1	0	Status type = terminal status
18	0	0	0	0	0	0	0	0	Slot (= 0 = general status of the bus coupler)
19	0	0	0	0	0	0	SP	SP	Specifier
20	ST LD 4		ST LD 3		ST LD 2		ST LD 1		Status, local bus devices 1 to 4
21	ST LD 8		ST LD 7		ST LD 6		ST LD 5		Status, local bus devices 5 to 8
...	…	…	…	…	…	…	…	…	…
35	0		ST LD 63		ST LD 62		ST LD 61		Status, local bus devices 61 to 63
Block 4									
36	1	0	Slot						Channel-specific diagnostics (header) slot 1
37	IO		Channel						IN/OUT and channel number slot 1
38	CT			ET					Channel and error type slot 1
...	1	0	Slot number						Channel-specific diagnostics (header) slot 2
...	IO		Channel						IN/OUT and channel number slot 2
...	CT			ET					Channel and error type slot 1
...	...	...	...	...	...	...	...	...	...
...	1	0	Slot number						Channel-specific diagnostics (header) slot 10
...	IO		Channel						IN/OUT and channel number slot 10
65	CT			ET					Channel and error type slot 10

Table 2-1 IL PB BK DI8 DO4-PAC format diagnostics

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Meaning
Block 5									
66	1	1	X	X	X	X	X	X	Version (start with C1)
Block 6									
67	0	0	0	0	1	1	0	0	Status (header)
68	1	0	1	0	0	0	0	0	Reserved (header)
69	0	0	0	0	0	0	0	0	Reserved (header)
70	0	0	0	0	0	0	0	0	Reserved (header)
71	0	0	0	0	0	0	B	B	Local bus baud rate 1 = 500 kbaud 2 = 2 Mbaud
72	X	X	X	X	X	X	X	X	Diagnostic location (FF = OK)
73	X	X	X	X	X	X	X	X	User ID code, 1st local bus device with error
74	X	X	X	X	X	X	X	X	User length code, 1st local bus device with error
75	X	X	X	X	X	X	X	X	LB state (high byte)
76	X	X	X	X	X	X	X	X	LB state (low byte)
77	0	0	0	0	0	0	0	X	System limit violated?
78	X	X	X	X	X	X	X	X	Number of accessible local bus devices

Block 1: PROFIBUS standard diagnostics

X: Value 1 is activated. Value 0 is deactivated.

M: Slave transmits 0, master adds if necessary.

Table 2-2 Byte 1: Status 1

7	6	5	4	3	2	1	0	
							M	Station does not exist
						X		Slave is not ready for data exchange
					X			Error in configuration telegram
				X				Extended diagnostics follows in the telegram
			X					The requested function is not supported by the slave
		M						Invalid response from slave
	X							Error in parameter telegram
M								Slave assigned to another master

Table 2-3 Byte 2: Status 2

7	6	5	4	3	2	1	0	
							X	Slave must be reparameterized
						X		Static diagnostics
					1			Fixed to 1 for DP operation
				X				Watchdog activated
			X					Freeze command received
		X						Sync command received
	0							Reserved
M								Slave is deactivated

Table 2-4 Byte 3: Status 3

7	6	5	4	3	2	1	0	
	0	0	0	0	0	0	0	Reserved
X								Slave has more diagnostic information than displayed in the telegram

Table 2-5 Byte: Master address

7	6	5	4	3	2	1	0	
0-125 (00_{hex}-$7E_{hex}$)								Master address following parameterization, default address is 255 (FF_{hex})

Table 2-6 Byte 5 and 6: ID number

7	6	5	4	3	2	1	0	
0-225 (00_{hex}-FF_{hex})								ID number high byte
0-225 (00_{hex}-FF_{hex})								ID number low byte

Block 2: ID-specific diagnostics

The table shows the faulty local bus devices. For every faulty terminal, a "1" is entered.

In the first byte of the block, bits 0 to 5 specify the number of local bus devices and therefore the length L of the block (8 devices for each byte, maximum). The minimum length of this block is 2 bytes (1-byte header + 1 byte (8 devices, maximum)), the maximum length is 9 bytes (1-byte header + 8 bytes (63 devices, maximum)). The size of the ID-specific diagnostic block depends on the number of configured terminals.

Block 3: Status PDU (terminal status)

For every local bus device there are 2 bits for status encoding:

ST LD x: 00	= Terminal data valid
ST LD x: 01	= Terminal data invalid due to an error
ST LD x: 10	= Terminal data invalid due to incorrect terminal being connected
ST LD x: 11	= Terminal data invalid or no terminal connected (despite configuration)

The specifier (SP) equals 1 in the event of a faulty state. The specifier equals 2 if the terminal changes from a faulty state to an error-free state. If the specifier equals 0, the state has not changed.

SP: 0	= No evaluation
SP: 1	= Error occurs (number > 0)
SP: 2	= Error disappears (number = 0)
SP: 3	= Reserved

In the first byte of the block, bits 0 to 5 specify the number of local bus devices and therefore the length of the Status-PDU block. The minimum length of this block is 5 (4-byte header + 1 byte (4 devices, maximum)), the maximum length is 20 (4-byte header + 16 bytes (63 devices, maximum)). The size of the status PDU block therefore depends on the number of configured terminals.

Block 4: Channel-specific diagnostics

Up to 10 channel errors are indicated here. There are 3 bytes per channel error, this block can be a maximum of 30 bytes in size. Each channel error is an individual and independent error. In order to display the channel errors, the I/O terminal must support PCP (Peripherals Communication Protocol) and channel-specific diagnostics must be activated on the bus coupler.

IO: 00_{bin}	= Reserved
IO: 01_{bin}	= Input
IO: 10_{bin}	= Output
IO: 11_{bin}	= Input and output

Channel: Channel number of the relevant channel (0 to 63)

CT: 000_{bin}	= Reserved
CT: 001_{bin}	= 1 bits
CT: 010_{bin}	= 2 bits
CT: 011_{bin}	= 4 bits
CT: 100_{bin}	= 1 byte
CT: 101_{bin}	= 1 word
CT: 110_{bin}	= 2 words
CT: 111_{bin}	= Reserved

ET: 0	= Reserved
ET: 1	= Short circuit
ET: 2	= Undervoltage
ET: 3	= Surge voltage
ET: 4	= Overload
ET: 5	= Overtemperature
ET: 6	= Cable break
ET: 7	= Upper limit value exceeded
ET: 8	= Lower limit value exceeded
ET: 9	= General error

Block 5: Revision diagnostics

Indicates the firmware version, e.g., $C3_{hex}$ = Version 3

Block 6: Status PDU

The sixth block is also encoded as a status PDU block, but in manufacturer-specific format. The information stored here does not usually need to be evaluated.

Byte 5 indicates the baud rate at which the local bus is operated.

Byte 7 and byte 8 indicate the ID and length code of the first faulty terminal. This can be used as additional information for blocks 2 and 3, if required.

The value of byte 11 is 1 if a system limit was violated.

Byte 12 indicates the number of available terminals.

The bus coupler is always counted as a local bus device (LD). In the event of an error on the bus coupler, the corresponding bits for local bus device 1 are set in the diagnostics.

2.1.2 Status PDU diagnostic format

This diagnostic format consists of the following blocks:

1 PROFIBUS standard diagnostics (see also 2.1.1 on page 2-2)

2 Status-PDU (manufacturer-specific)

Table 2-7 Status PDU diagnostics incl. error codes

Byte X	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Meaning
Block 1									
1	X	X	X	X	X	X	X	X	Station status 1
2	X	X	X	X	X	X	X	X	Station status 2
3	X	X	X	X	X	X	X	X	Station status 3
4	X	X	X	X	X	X	X	X	PROFIBUS master address
5	0	0	0	0	1	0	1	1	Manufacturer ID high byte ($0B_{hex}$)
6	0	1	0	1	0	0	0	0	Manufacturer ID low byte (50_{hex})
Block 2									
7	0	0	0	0	1	0	0	1	Status (header) + length
8	1	0	1	0	0	0	0	1	Status type = e.g., $A1_{hex}$ (header)
9	X	X	X	X	X	X	X	X	Slot number (header)
10	0	0	0	0	0	0	X	X	Status specifier (header)
11	X	X	X	X	X	X	X	X	Error code, high byte
12	X	X	X	X	X	X	X	X	Error code, low byte
13	X	X	X	X	X	X	X	X	User ID code, local bus device with error
14	X	X	X	X	X	X	X	X	User length code, local bus device with error
15	X	X	X	X	X	X	X	X	FW version (e.g., 33_{hex})

Block 2: Status PDU

A distinction is made between the following status types (byte 8):

$A1_{hex}$	PROFIsafe F-Parameter error (see also Appendix A3)
$A2_{hex}$	PROFIsafe iParameter error
$A3_{hex}$	PROFIsafe device error
81_{hex}	Standard status message

Byte 11 and byte 12 provide more detailed information on the standard status message:

High byte	Error type
Low byte	Error number

For error descriptions, please refer to Section 2.2, "Representing the error codes", Section A 3, "Error codes for F-Parameters" or the documentation of the individual module.

2.1.3 ID-specific diagnostic format

This diagnostic format consists of the following blocks:

1 PROFIBUS standard diagnostics (see also 2.1.1 on page 2-2)

2 ID-specific diagnostics (see also 2.1.1 on page 2-6)

Table 2-8 ID-specific (terminal) diagnostics

Byte X	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Meaning
Block 1									
1	X	X	X	X	X	X	X	X	Station status 1
2	X	X	X	X	X	X	X	X	Station status 2
3	X	X	X	X	X	X	X	X	Station status 3
4	X	X	X	X	X	X	X	X	PROFIBUS master address
5	0	0	0	0	1	0	1	1	Manufacturer ID high byte ($0B_{hex}$)
6	0	1	0	1	0	0	0	0	Manufacturer ID low byte (50_{hex})
Block 2									
7	0	1	X	X	X	X	X	X	Number of devices (values 0 to 63)
8	X	X	X	X	X	X	X	X	Terminals 1 to 8 (values 0 to 255)
9	X	X	X	X	X	X	X	X	Terminals 9 to 16 (values 0 to 255)
10	X	X	X	X	X	X	X	X	Terminals 17 to 24 (values 0 to 255)
11	X	X	X	X	X	X	X	X	Terminals 25 to 32 (values 0 to 255)
12	X	X	X	X	X	X	X	X	Terminals 33 to 40 (values 0 to 255)
13	X	X	X	X	X	X	X	X	Terminals 41 to 48 (values 0 to 255)
14	X	X	X	X	X	X	X	X	Terminals 49 to 56 (values 0 to 255)
15	X	X	X	X	X	X	X	X	Terminals 57 to 63 (values 0 to 127)

2.2 Representing the error codes

2.2.1 Local diagnostic indicators

Error type and error no. are indicated by flashing codes of the local diagnostic LEDs (LED FS and LED FN).

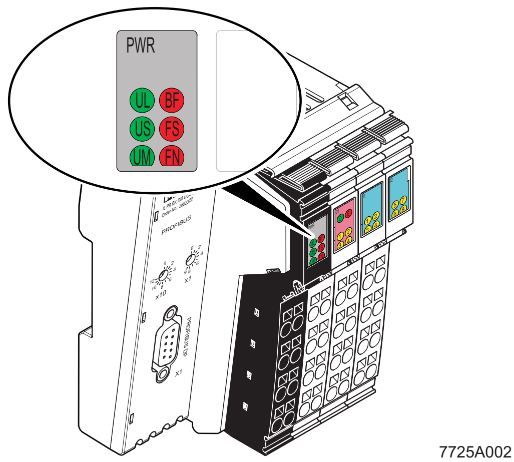


Figure 2-2 Local diagnostic indicators on the bus coupler

2.2.2 Error type and error no.

In the controller, error type and error no. are available via status-PDU diagnostics.

Table 2-9 Error type and error no. (also FS/FN flashing codes)

Error type (FS ON, FN flashing)	Error no. (FS OFF, FN flashing)	Error cause	Error remedy
1		**Parameter errors on PROFIBUS**	
	1	A parameter block is faulty.	The number of terminals does not correspond to the number of parameter blocks or the header byte of the module parameter is incorrect or the parameter block is incomplete.
	2	Too many data blocks for the station.	The number of terminals does not correspond to the number of parameter blocks.
	3	The data length of the parameter block is too short.	Check the number of parameters.
	4	The PD PCP module appears several times in the configuration.	Check the configuration with regard to the number of PD PCP modules. There must only be one PD PCP module. (Except for PROFIsafe terminals - these have two PD PCP modules).
	5	The data block length was exceeded.	Check the number of terminals.
	6	A parameter block is not complete.	Check the structure of the parameters for the terminals.
	7	The number of parameter blocks is larger than the maximum number of terminals in the station.	Check the configuration.
	8	F-Parameter error, PROFIsafe-parameter error	Activate Status-PDU diagnostics when parameterizing the bus coupler in your controller and determine the exact error location and error cause.
	9	iParameter error, PROFIsafe parameter error	Activate Status-PDU diagnostics when parameterizing the bus coupler in your controller and determine the exact error location and error cause.

Table 2-9 Error type and error no. (also FS/FN flashing codes) (continued)

Error type (FS ON, FN flashing)	**Error no. (FS OFF, FN flashing)**	**Error cause**	**Error remedy**
2		**Configuration error on PROFIBUS**	
	1	Fewer terminals have been configured than are available in the station.	Add these terminals to the configuration.
	2	More terminals have been configured than are available in the station.	Delete the extra terminals from your configuration.
	3	Configuration block is faulty.	Determine the exact error location using the terminal-specific diagnostics in your control system.
	4	The ID code in the configuration does not correspond to the terminal.	Determine the exact error location using the terminal-specific diagnostics in your control system. Check the configuration in the hardware configurator.
	5	The length code of the configured terminal does not correspond to the length code of the terminal.	Determine the exact error location using the terminal-specific diagnostics in your control system. Check the configuration in the hardware configurator.
	6	The data length of the parameter block is faulty.	Check the number of parameters. The parameter length must not be longer than the output data length of the relevant terminal.
	7	PROFIBUS address of the IL PB BK DI8 DO4/EF-PAC is larger than 126.	Check the PROFIBUS address of the IL PB BK DI8 DO4/EF-PAC. After the address is set, the IL PB BK DI8 DO4/EF-PAC must be restarted.
	8	More than 244 bytes are required for the configuration.	Reduce the number of terminals in the station or group several Inline terminals in the configuration, so that the configuration data is compressed.

Table 2-9 Error type and error no. (also FS/FN flashing codes) (continued)

Error type (FS ON, FN flashing)	Error no. (FS OFF, FN flashing)	Error cause	Error remedy
3	**Configuration and parameter errors on the local bus**		
	1	General PCP parameterization failed.	Check the station configuration and the PCP devices.
	2	PCP communication was aborted.	Check the PCP PDU size of the PCP devices.
	3	Module error occurred during initialization.	Try to restart the station.
	4	An error occurred in the I/O circuit (e.g., short circuit or overload at the actuator).	Remove the error from your I/O devices.
	5	Terminal error	Activate channel-specific diagnostics when parameterizing the bus coupler in your controller and determine the exact error location and error cause.
	6	Safety terminal error	Activate Status-PDU diagnostics when parameterizing the bus coupler in your controller and determine the exact error location and error cause.
	7	More than 62 Inline devices are connected.	Re-configure the station structure with regard to the maximum permissible number of Inline devices.
	8	More than 16 PCP devices are connected.	Reduce the number of PCP devices in the station.
	9	The sum of the process data in the local bus is greater than 244 bytes.	Check the amount of process data and reduce the number of devices in the station.
4	**Local bus error within the station**		
	1	Local bus was stopped.	Check the configuration of the station.

3 Acyclic communication (DP/V1 and PCP)

DP/V1

DP/V1 extends the cyclic data exchange function according to IEC 61158 to include acyclic services. This makes it easy to operate even complex terminals.

PCP

PCP is used in the local bus to exchange data acyclically. This is usually the parameterization data of complex terminals (e.g., IB IL RS 232) or variable length data.

DP/V1 is a mechanism which corresponds to PCP at PROFIBUS level. The IL PB BK DI8 DO4/EF-PAC coupler prepares the data records, which are sent via DP/V1 from the Class 1 or Class 2 master, for the PCP mechanism in the local bus. PCP data from the local bus is in turn converted into DP/V1 telegrams by the bus coupler.

> Before programming the application, check whether your control system or configuration tool supports DP/V1. If not, the functions can be used via the cyclic process data channel (DP/V0), see Section 3.5 on page 3-13 and onwards.

The different communication types will be described in the following.

3.1 Acyclic communication via the Class 1 master (C1 master)

C1 master

The C1 master carries out parameterization during slave startup and is also the master for cyclic data traffic. It may also be necessary to read a parameter acyclically from the terminal as an option using this C1 master.

Corresponding read and write access rights are therefore defined for the C1 master. As it already has a connection to the slave during cyclic data traffic, the C1 master does not have to establish an explicit connection (using "Initiate"), but can communicate with the slave directly via "Read" and "Write".

3.2 Acyclic communication via the Class 2 master (C2 master)

C2 master

For communication in the C2 master, the data fields are identical to those for C1 communication, and it is only the SAPs (Service Access Points) which differ. Additional fields are "Initiate" and "Abort" to establish and release the connection via SAP49 and 50. If DP/V1 terminals are already in use, the routines for connection management can be adapted easily. The C2 master can be implemented in various forms, e.g., in the form of a display device or operator interface. In a display device, the data is retrieved from the slave on request if, for example, a specific parameter is to be read. Access to the operator interface is usually acyclic.

> Up to 16 terminals capable of PCP can be connected to the IL PB BK DI8 DO4/EF-PAC bus coupler.

3.3 PCP communication basics

PCP (Peripherals Communication Protocol) controls the transmission of parameter data in the local bus. Special PCP services are available for this purpose.

Application example

To explain the basics of PCP communication, the following concrete PCP application is used as an example:

A frequency inverter (FI), together with other field devices, is connected to a PLC via a bus interface. The device versions are standardized according to the Drives profile.

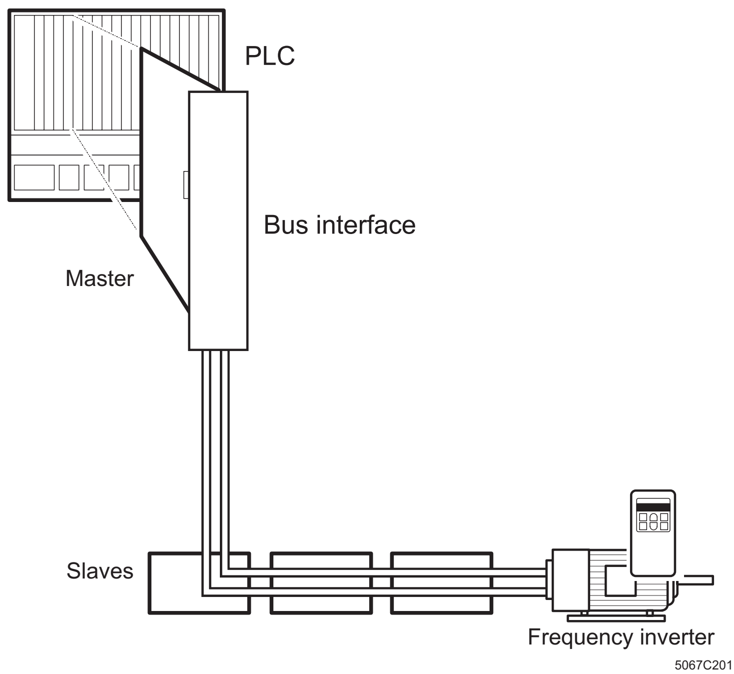


Figure 3-1 Application example

Device parameters

Device parameters are data from intelligent field devices (PCP devices), which is required for the startup phase of machines and systems. Once it has been entered, this data only has to be modified upon a change in the parameterization or in the event of an error. The parameters are preconfigured and can be taken from the device documentation provided by the manufacturer.

Parameters of a frequency inverter

As an electrical drive controller, a key feature of a frequency inverter is that changes can be made to process variables (e.g., speed, position, and torque) using analog or digital signals. Additional information is required to optimize the adaptation of the drive controller and motor to the process. As well as setpoint information, the frequency inverter also requires information about the motor type point, the minimum and maximum permissible speed of the system, the maximum speed variation during acceleration and deceleration, starting ramp, starting current, etc.

These types of additional information are device-specific parameters, which can be modified via the parameter data channel.

The parameter values for all PCP devices are the subject of communication via the parameter data channel. To enable the individual parameters to be distinguished during communication, each parameter has a number, the index.

Index

The index is the address of the communication object. It is required to identify the object.

Table 3-1 Object description (example)

	Object description (OD)		
Index	**Type**	**Object**	**Name**
...	...	...	...
60 $4A_{hex}$	Ramp	Record	Speed quick stop
60 $4B_{hex}$	Integer16	Array	Setpoint factor
...	...	...	...

Object description

The object description includes all the properties of the object, such as data type, object type, name, etc.

Object types

There are various different object types:

Simple variable

– Simple variable type objects.
Examples include measured values, the time or status of a terminal.

Array

– Array type objects, i.e., several "simple variable" objects of the same type, which are grouped to form one object. Each element can be accessed individually.
An example of an array is a range of the same type of measured values.

Record

– Record type objects, i.e., several "simple variable" objects of different types, which are grouped to form one object. As for the array type, each element of a record can be accessed individually. An example of a record is the group of data in a test report, which contains not only the actual measured value, but also additional information, e.g., the time of the measurement.

Program invocation

– Program invocation type objects, i.e., program sequences that can be run.

For additional information about PCP communication, please refer to the IBS SYS PCP G4 UM E user manual, Order No. 2745169.

3.4 Acyclic communication in DP/V1 mode

3.4.1 The communication mechanism

Whenever data is accessed, a distinction must be made between accessing data from local bus devices and data from the IL PB BK DI8 DO4/EF-PAC bus coupler:

Table 3-2 Objects on the IL PB BK DI8 DO4/EF-PAC bus coupler

Data type	Access to I/O terminal	Access to IL PB BK DI8 DO4/EF-PAC	Slot	Index/ dec
PDU length	–	x	0	3
Master control	–	x	0	4
PCP status	–	x	0	5
Terminal activation	–	x	0	6
Terminal activation/restart	–	x	0	7
Terminal diagnostics	–	x	0	12
Cycle count	–	x	0	20
Cycle error count	–	x	0	21
ID cycle count	–	x	0	22
ID cycle error count	–	x	0	23
Data cycle count	–	x	0	24
Data cycle error count	–	x	0	25
I & M functions	–	x	0	255
PD IN	–	x	1 to 63	13
PCP access	x	–	2 to 63	47

When accessing the IL PB BK DI8 DO4/EF-PAC bus coupler, use the DP/V1 format. Read and write accesses can be executed in one step (request -> response).

The PCP data from I/O terminals is addressed via 16-bit object indices. DP/V1 only has fields for 8-bit indices. Additional parameters have therefore been added to the data block for use when accessing the local bus, as for PROFIDrive. A sequence involving 2 steps is used, which follows the PROFIDrive profile:

Read:

1. a) Send the request as a DP/V1 write request (PCP read) to slot x
   b) Receive the DP/V1 write response - often automatically via the master
2. a) Send a DP/V1 read to slot x
   b) Receive the DP/V1 read response - usually automatically via the master

Write:

1. a) Send the request as a DP/V1 write request (PCP write) to slot x
   b) Receive the DP/V1 write response - often automatically via the master
2. a) Send a DP/V1 read to slot x
   b) Receive the DP/V1 read response - usually automatically via the master

Note that when communicating with objects on local bus devices, the response should be fetched using "Read". Otherwise the result will be overwritten during the next communication. Communication is always carried out via DP/V1 index 47, and the object index and assigned subindex of the I/O devices are transmitted as part of the data field.

Request and response

The section below provides additional information about the format of write and read access (request and response).

The format for all types of access (request and response (positive), read and write) in DP/V1 is:

<DP/V1 header> <Data (PCP/DP/V1)>

The DP/V1 header for a positive DP/V1 response always has the following format:

<DP/V1 service (positive)> <Slot> <DP/V1 index> <DP/V1 length>

In the event of a faulty response, the format is as follows:

– For a DP/V1 error:

  <DP/V1 service (negative)> <Error decode> <Error code 1> <Error code 2>

– For an I/O device error:

  <DP/V1 service (positive)> <Slot> <DP/V1 index> <DP/V1 length>
  <Error data (PCP/DP/V1)>

The <Data (PCP/DP/V1)> is optional depending on the service and has the following structure:

Table 3-3 Structure of the data depending on the service

Access	Service	Data
Write objects (IL PB BK DI8 DO4/EF-PAC)	DP/V1 write request	Object data
	DP/V1 write response	None
Read objects (IL PB BK DI8 DO4/EF-PAC)	DP/V1 read request	None
	DP/V1 read response	Object data
Write objects (I/O device)	DP/V1 write request (PCP write)	Write PCP request/Invoke ID/Index high/Index low/Subindex/Length of PCP data/x bytes of PCP object data
	DP/V1 write response	None
	DP/V1 read request	None
	DP/V1 read response	Write PCP ack/Invoke ID/Status
Read objects (I/O device)	DP/V1 write request (PCP read)	Read PCP req/Invoke ID/Index high/Index low/ Subindex
	DP/V1 write response	None
	DP/V1 read request	None
	DP/V1 read response	Read PCP ack/Invoke ID/Status/ Length of PCP data/x bytes of PCP object data

The meaning of the individual parameters is as follows:

– <DP/V1 service>:
  In the request there is a distinction between DP/V1 read ($5E_{hex}$) and DP/V1 write ($5F_{hex}$); in the error response there is a distinction between DE_{hex} (read error) and DF_{hex} (write error).

- <Slot>:
  The slot of the device to be addressed in the station. The bus coupler is addressed with slot = 0, the integrated DI8/DO4 terminals are addressed with slot = 1. Starting with the first connected device, the devices are addressed with slots 2 to 63.
- <DP/V1 index>:
  For access to the PCP objects of the local bus devices, PROFIBUS index 47_{dec} (= $2F_{hex}$) is to be used. The PCP index is transmitted as part of the data field. For access to the bus coupler, the object index can be used directly.
- <DP/V1 length>:
  For write access, the length of the subsequent data is specified here, and for read access, the length of the expected data is specified. On a response, this parameter contains the actual length of the DP/V1 data.
- <Error decode>:
  80_{hex} indicates an error in DP/V1.
- <Error code 1> and <Error code 2>:
  Error codes from DP/V1 access (see "Error codes for DP/V1 and VC1 communication" on page A-3).
- <Write PCP/read PCP>:
  This specifies whether the following object indices should be written or read. Read PCP = 06_{hex}; Write PCP = 07_{hex}.
- <Object data>:
  This is only the contents of an object. The length and scope of the data has already been described by <DP/V1 length>.
- <Invoke ID>:
  The Invoke ID is used for some I/O devices. Check this in the relevant data sheet.
- <Index high and Index low>:
  This specifies the object index of the addressed PCP object in two bytes. For example, for index 2300_{hex} the value 23_{hex} should be entered for Index high and the value 00_{hex} should be entered for Index low.
- <Subindex>:
  When working with a PCP object, the subindex can be used to select a specific element from an array or record.
- <Length of PCP data>:
  This value specifies how many bytes of PCP object data (object contents) follow.
- <PCP object data>:
  This is the actual contents of a PCP object.
- <Status>:
  For a positive PCP response, the status is = 00_{hex}, in the event of an error it is 44_{hex}.
- -<Error data (PCP/DP/V1)>:
  The structure of error data is as follows, <PCP confirmation code> <Invoke ID> <Status = 44_{hex}> <PCP error code (4 bytes)>.
  (PCP error code, see "Error codes for DP/V1 and VC1 communication" on page A-3).

> When accessing PCP, please note the first byte in the DP/V1 data block. With 06_{hex}, PCP Read is executed and with 07_{hex} PCP Write is executed.

3.4.2 Examples

The section below provides a few examples to aid understanding (all values in hex). These examples indicate how objects on the bus coupler and the I/O devices can be read and written.

The station structure is as follows:
- IL PB BK DI8 DO4/EF-PAC
- IB IL 24 DI8-PAC
- IB IL AI4/EF-PAC
- IB IL AO 4/8/U/BP-PAC
- PD/PCP x words

The IB IL AO 4/8/U/BP and IB IL AI4/EF are capable of PCP.

In order to understand the examples, the object dictionary for the IB IL AI4/EF-PAC terminal should be referred to:

The object dictionary for the IB IL AI4/EF-PAC contains the following object, for example:

Table 3-4 Object dictionary for the IB IL AI4/EF

Index	Data type	N	L	Meaning	Object name	Rights
0080_{hex}	Array of Unsigned 16	6	2		Config Table	rd/wr

N: Number of elements

L: Length of an element in bytes

rd: Read access permitted

wr: Write access permitted

Table 3-5 Object description:

Object	Config Table	
Access	Read, write	
Data type	Array of Unsigned 16	6 x 2 bytes
Index	0080_{hex}	
Subindex	00_{hex}	Write all elements
	01_{hex}	Configuration of channel 1
	02_{hex}	Configuration of channel 2
	03_{hex}	Configuration of channel 3
	04_{hex}	Configuration of channel 4
	05_{hex}	Reserved
	06_{hex}	Reserved
Length (bytes)	$0C_{hex}$	Subindex 00_{hex}
	02_{hex}	Subindex 01_{hex} to 06_{hex}
Data	Terminal configuration	

Objects 4 and 4 on the bus coupler are also used in these examples. For an explanation regarding these objects, please refer to section "Object dictionaries" on page B-1.

Example 1:
Reading the connected local PCP devices and their status (slot 0, index 5 on the bus coupler)

Read request (master -> slave)

Data	Data structure
5E 00 05 20	DP/V1 Read / Slot / Index / Maximum Length

Read request (slave -> master)

Data	Data structure
5E 00 05 06 03 01 00 04 01 00	DP/V1 read/Slot/Index/Actual length/ 6 bytes of object data

The data shows that there is one PCP device each on slots 3 and 4, and its connection status is OK, see Section "Object dictionaries" on page B-1.

Example 2:
Writing the bus restart bit (slot 0, index 4, bit 0 on the bus coupler)

Write request (master -> slave)

Data	Data structure
5F 00 04 01 01	DP/V1 write/Slot/Index/Length/1 byte of data

Write response (slave -> master)

Data	Data structure
5F 00 04 01	DP/V1 write/Slot/Index/Length

The data block is only important in the request. The response indicates that the command has been received. As can be seen in Section "Object dictionaries" on page B-1, the local bus is restarted with bit 0 (01_{hex}) at index 4, slot 0.

Example 3:
Reading the Config Table on the connected IB IL AI4/EF (slot 3, index 0080)

Write request (master -> slave)

Data	Data structure
5F 03 2F 05 06 00 00 80 00	DP/V1 write/Slot/Index/Length/Read PCP request/Invoke ID/PCP index high/PCP index low/PCP subindex

Write response (slave -> master)

Data	Data structure
5F 03 2F 05	DP/V1 write/Slot/Index/Length

Read request (master -> slave)

Data	Data structure
5E 03 2F 20	DP/V1 Read / Slot / Index / Maximum Length

Read response (slave -> master)

Data	Data structure
5E 03 2F 10 86 00 00 0C 03 01 03 00 03 01 03 00 00 00 00 00	DP/V1 read/Slot/Index/Actual length/PCP read response/Invoke ID/Status/Length of PCP data/12 bytes of PCP object data

In the read response, the user receives the 12 bytes of object 80_{hex} on the IB IL AI4/EF as described above. The settings for channels 1 and 3 are the same as well as for channels 2 and 4. Bytes 9 to 12 are reserved and are indicated as 0. The Invoke ID was mirrored and the status indicates that communication was error-free.

Example 4:
Writing the Config Table for channel 3 on the connected IB IL AI4/EF (slot 3, index 0080, subindex 3)

Write request (master -> slave)

Data	Data structure
5F 03 2F 08 07 00 00 80 03 02 03 01	DP/V1 write/Slot/Index/Length/Write PCP request/Invoke ID/PCP index high/ PCP index low/PCP subindex/ Length of PCP object data/PCP object data

Write response (slave -> master)

Data	Data structure
5F 03 2F 08	DP/V1 write/Slot/Index/Length

Read request (master -> slave)

Data	Data structure
5E 03 2F 20	DP/V1 Read / Slot / Index / Maximum Length

Read response (slave -> master)

Data	Data structure
5E 03 2F 03 87 00 00	DP/V1 read/Slot/Index/Actual length/ Write PCP response/Invoke ID/Status

In the read response, the mirrored Invoke ID is the result and the status is 00.

Example 5:
In the event of an error: Reading a faulty object index on the connected IB IL AI4/EF (slot 3, index 0180)

Write request (master -> slave)

Data	Data structure
5F 03 2F 05 06 00 01 80 00	DP/V1 write/Slot/Index/Length/Read PCP request/Invoke ID/PCP index high/ PCP index low/PCP subindex

Write response (slave -> master)

Data	Data structure
5F 03 2F 05	DP/V1 write/Slot/Index/Length

Read request (master -> slave)

Data	Data structure
5E 03 2F 20	DP/V1 Read / Slot / Index / Maximum Length

Read response (slave -> master)

Data	Data structure
5E 03 2F 07 86 00 44 06 07 00 00	DP/V1 read/Slot/Index/Actual length/ PCP read response/Invoke ID/Status/ Error data PCP

Status byte 44_{hex} indicates an error during execution. In this case, the PCP read request is sent to the I/O terminal first. However, the I/O terminal does not recognize this index and acknowledges it using error code 06 07. According to Section "Error codes for PCP communication" on page A-1, this means that this object does not exist. The last 2 bytes are also part of the PCP error data, however, they are not used in this example. If they do not equal zero, refer to the relevant I/O device data sheet for more detailed information.

Example 6:
In the event of an error: Reading an object on a device without PCP (slot 2, index 0080)

Write request (master -> slave)

Data	Data structure
5F 02 2F 05 06 00 00 80 00	DP/V1 write/Slot/Index/Length/Read PCP request/Invoke ID/PCP index high/ PCP index low/PCP subindex

Write response (slave -> master)

Data	Data structure
DF 80 D2 00	DP/V1 write error/Error decode/Error code 1/Error code 2

Since the addressed device does not support PCP, the request is rejected immediately. A message is sent by DP/V1, whereby error code D2 stands for "Terminal does not have PCP". See also Section "Error codes for DP/V1 and VC1 communication" on page A-3.

3.5 Acyclic communication in DP/V0 mode via process data

DP/V1 communication is relatively new compared to cyclic DP/V0 communication. However, the service life of control systems and plants is so long that extensions and modifications are made. In many cases, the control system is not DP/V1-compatible, but is expected to operate complex devices.

Acyclic services

Consequently, it is possible to operate acyclic services within the process data. That means even a control system that does not have DP/V1 can operate more complex terminals.

For additional information about PCP communication, please refer to Sections "PCP communication basics" on page 3-2 and "Acyclic communication in DP/V1 mode" on page 3-4.

3.5.1 Mechanism for transmission in the process data

VC1 module

Transmission is via a virtual C1 module (VC1 module). A C1 module should be selected in the hardware configurator in the same way as "normal" I/O devices and therefore specified in the configuration and parameter telegram.

The VC1 module is only a virtual device because the process data can be used to transmit communication data (PCP) and is not linked to a specific module. During active process data exchange, it is possible to assign the VC1 module sequentially to different modules with communication objects and to exchange parameter data parallel to the process data.

Process data width

The process data width occupied by the VC1 module in the process data channel can be selected from 4 to 16 words in increments of 2 words. This means that communication objects can be used even if resources are limited. If there are sufficient free resources, a data width of up to 16 words can be used, providing the same ease of operation as for DP/V1 communication.

As the data width of the VC1 module is between 4 and 16 words, but the user data can be up to 72 bytes (36 words) per communication, it may be necessary to split the data and transmit it in several steps.

This leads to:

– Start fragment
– Continue fragment
– End fragment
– Error or abort fragment

Each fragment contains a service byte, which is used for the precise assignment of the fragment. The individual fragments and the service byte are explained in detail in the following.

> The VC1 module (listed in the GSD as "PD PCP x words") may be configured at any position after the IL PB BK DI8 DO4/EF-PAC bus coupler. We recommend configuring the VC1 device in the last position. In this way the configured slot and the actual slot occupied by the I/O device will always be the same. It is not linked to any hardware, so a module is not actually inserted.

Start fragment:

Byte 1:	**Service**
Byte 2:	Terminal number
Byte 3:	Invoke ID
Byte 4:	Index high
Byte 5:	Index low
Byte 6:	Subindex
Byte 7:	Length, if required
Byte 8:	Data block, if required
...	
Byte n:	Data block, if required

Table 3-6 Byte 1 - Service in start fragment:

Byte 1							
7	6	5	4	3	2	1	0
Request/ Response	0	0	Fragmenta- tion	Action			

Bit 7:	**Request/Response**
	0 = Request
	1 = Response
Bits 6 to 5:	**Fragment type**
	00 = Start fragment
Bit 4:	**Fragmentation**
	0 = Not fragmented
	1 = Fragmented
Bits 3 to 0:	**Action**
00_{hex}	No action (clear)
01_{hex} to 02_{hex}	Reserved
03_{hex}	Read (IL PB BK DI8 DO4/EF)
04_{hex}	Write (IL PB BK DI8 DO4/EF)
05_{hex}	Reserved
06_{hex}	Read PCP (I/O device)
07_{hex}	Write PCP (I/O device)
08_{hex}	Read and write simultaneously (special objects only)
09_{hex} to $0F_{hex}$	Reserved

Continue fragment:

Byte 1: **Service**
Byte 2: Data block, if required
...
Byte n: Data block, if required

Table 3-7 Byte 1 - Service in continue fragment:

Byte 1							
7	6	5	4	3	2	1	0
Request/ Response	0	1	Fragment number (01_{hex} - $1F_{hex}$)				

Bit 7: **Request/Response**
0 = Request
1 = Response

Bits 6 to 5: **Fragment type**
01 = Continue fragment

Bits 4 to 0: **Counter**
01_{hex} to $1F_{hex}$ fragment number

End fragment:

Byte 1: **Service**
Byte 2: Data block, if required
...
Byte n: Data block, if required

Table 3-8 Byte 1 - Service in end fragment:

Byte 1							
7	6	5	4	3	2	1	0
Request/ Response	1	0	Reserved				

Bit 7: **Request/Response**
0 = Request
1 = Response

Bits 6 to 5: **Fragment type**
10 = Last fragment (end fragment)

Bits 4 to 0: **Reserved**

Abort/error fragment:

Byte 1: **Service**
Byte 2: Error code, if required
...
Byte n: Error code, if required

Table 3-9 Byte 1 - Service in abort/error fragment:

Byte 1							
7	6	5	4	3	2	1	0
Request/ Response	1	1	Reserved				

Bit 7: **Request/Response**
0 = Request
1 = Response

Bits 6 to 5: **Fragment type**
11 = Abort/error fragment

Bits 4 to 0: **Reserved**

Communication can be reset at any time using 00_{hex} and 60_{hex}.

A response is sent after every request. This response indicates that the request has been received and shows its current status:

Response structure:

Byte 1: **Service (mirrored request with set response bit)**
Byte 2: Status, if required
Byte 3: Length, only on first read response
...
Byte n: Data block, if required

The status is indicated when local PCP transmission is complete and in the event of an error. In the event of an error, the data block can provide details. An error has occurred if the value of the status byte does not equal 00_{hex}.

00_{hex} **No error**
44_{hex} Error during communication
Other errors See "Error codes for DP/V1 and VC1 communication" on page A-3.

DOWNLOADED FROM WWW.SCATTS.CO.UK

For VC1, the parameters have the following meaning:

– <Terminal number>:

  The IL PB BK DI8 DO4/EF-PAC bus coupler is counted as device 0, the integrated DI8/DO4 terminals as 1, and the first connected device and onwards as terminal = 2 ... 63.

– <Invoke ID>:

  The invoke ID has a length of one byte and is only used for a few I/O devices. Check this in the relevant data sheet.

– <Index high and Index low>:

  This specifies the object index of the addressed object in two bytes. This also applies for objects on the IL PB BK DI8 DO4/EF-PAC bus coupler. For example, for index $5FE0_{hex}$ the value $5F_{hex}$ should be entered for Index high and the value $E0_{hex}$ should be entered for Index low. For index 4_{hex} on the bus coupler, 00_{hex} is Index high and 04_{hex} is Index low.

– <Sub index:>

  When working with a PCP object, the subindex can be used to select a specific element from an array or record. Therefore the subindex should be specified when accessing I/O devices. The bus coupler has no arrays or records, so subindex 0 should be specified.

– <Length>:

  This value specifies how many bytes of object data (object contents) follow. Depending on the addressed slot, this may be bus coupler object data or I/O device object data.

– <Data block>:

  This is only the contents of an object. The length and scope of the data has already been described by the <Length> parameter.

To aid understanding, the same examples as for DP/V1 services are used in the following section. This means that the description of the examples for DP/V1 communication is valid again here, see Section "Examples" on page 3-7. See also "Error codes for DP/V1 and VC1 communication" on page A-3.

3.5.2 Examples for VC1 services

Example 1:
Reading the connected local PCP devices and their status (slot 0, index 5 on the bus coupler)

Read request (master -> slave)

Data (8 words VC1)	Data structure
03 00 00 00 05 00 I 00 00 00 00 00 00 00 00 00 00	Read request (bus coupler)/Slot/Invoke ID/ Index high/Index low/Subindex/ 10 bytes unused

Read response (slave -> master)

Data (8 words VC1)	Data structure
83 00 06 03 01 00 04 01 00 I 00 00 00 00 00 00 00	Read response (bus coupler)/Status/Actual length/6 bytes of object data/7 bytes unused

Clear request (master -> slave)

Data (8 words VC1)	Data structure
00 I 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	Clear request/15 bytes unused

Clear response (slave -> master)

Data (8 words VC1)	Data structure
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	Clear response

The status byte equals zero. This means that communication was error-free. The data shows that there is one PCP device each on slot 3 and 4, and its connection status is OK, see Section "Object dictionaries" on page B-1.

The communication data can be reset to the initial state via "Clear".

Example 2:
Writing the bus restart bit (slot 0, index 4, bit 0 on the bus coupler)

Write request (master -> slave) - Start fragment

Data (6 words VC1)	Data structure
04 00 00 00 04 00 01 01 I 00 00 00 00	Write request (bus coupler)/Slot/Invoke ID/Index high/Index low/Subindex/Length/ 1 byte of data/4 bytes unused

Write response (slave -> master)

Data (6 words VC1)	Data structure
84 00 I 00 00 00 00 00 00 00 00 00 00	Write response (bus coupler)/Status/ 10 bytes unused

Clear request (master -> slave)

Data (6 words VC1)	Data structure
00 I 00 00 00 00 00 00 00 00 00 00 00	Clear request/11 bytes unused

Clear response (slave -> master)

Data (6 words VC1)	Data structure
00 00 00 00 00 00 00 00 00 00 00 00	Clear response

The response indicates that the command has been received. Here, the status is positive (=0). The communication data can be reset to the initial state via "Clear".

Example 3:
Reading the Config Table on the connected IB IL AI4/EF (slot 3, index 0080)

Read request (master -> slave) - Start fragment

Data (4 words VC1)	Data structure
06 03 00 00 80 00 I 00 00	Read request (I/O)/Slot/Invoke ID/ Index high/Index low/Subindex/ 2 bytes unused

Read response (slave -> master)

Data (4 words VC1)	Data structure
96 00 0C 03 01 03 00 03	Read response/Status/Length/ 5 bytes of object data

Read request (master -> slave) - Start fragment acknowledgment

Data (4 words VC1)	Data structure
96 I 00 00 00 00 00 00 00	Start fragment acknowledgment/7 bytes unused

Read response (slave -> master)

Data (4 words VC1)	Data structure
C0 01 03 00 00 00 00 00	End fragment/7 bytes of object data

In the read response, the user receives the 12 bytes of object 80_{hex} on the IB IL AI4/EF as described above. The first 5 object data bytes are transmitted in the start fragment. The missing 7 bytes follow in the second fragment which is already the last. The status byte indicates that communication was error-free.

For the bus coupler the service is finished as soon as the last fragment has been transmitted. If the next service byte differs from the previous one, the next read/write access can be started immediately. Alternatively, the last fragment can be acknowledged or "Clear" can be sent.

Acknowledgment of the last fragment:

Read request (master -> slave) - End fragment acknowledgment

Data (4 words VC1)	Data structure
C0 I 00 00 00 00 00 00 00	End fragment acknowledgment/7 bytes unused

Read response (slave -> master)

Data (4 words VC1)	Data structure
00 00 00 00 00 00 00 00	Clear

or

Clear:

Clear request (master -> slave)

Data (4 words VC1)	Data structure
00 I 00 00 00 00 00 00 00	Clear request/7 bytes unused

Clear response (slave -> master)

Data (4 words VC1)	Data structure
00 00 00 00 00 00 00 00	Clear response

Example 4:
Writing the Config Table on the connected IB IL AI4/EF
(slot 3, index 0080, subindex 0)

Write request (master -> slave) - Start fragment

Data (4 words VC1)	Data structure
17 03 00 00 80 00 0C 03	Write request (I/O)/Slot/Invoke ID/PCP index high/PCP index low/PCP subindex/Length of PCP object data/1 byte of PCP object data

Write response (slave -> master)

Data (4 words VC1)	Data structure
17 I 00 00 00 00 00 00 00	Write response/7 bytes unused

Write request (master -> slave) - 1st continue fragment

Data (4 words VC1)	Data structure
21 00 03 01 03 00 03 01	1. Continue fragment / 7 bytes PCP object data

Write response (slave -> master)

Data (4 words VC1)	Data structure
21 I 00 00 00 00 00 00 00	Response/7 bytes unused

Write request (master -> slave) - End fragment

Data (4 words VC1)	Data structure
40 00 00 00 00 I 00 00 00	End fragment/4 bytes of PCP object data/ 3 bytes unused

Write response (slave -> master)

Data (4 words VC1)	Data structure
87 00 I 00 00 00 00 00 00	Response/Status/6 bytes unused

Clear request (master -> slave)

Data (4 words VC1)	Data structure
00 I 00 00 00 00 00 00 00	Clear request/7 bytes unused

Clear response (slave -> master)

Data (4 words VC1)	Data structure
00 00 00 00 00 00 00 00	Clear response

Here, the write specification is done in 3 fragments. The bus coupler only accepts the request once the last fragment has been received. The response "87" indicates that command "07" was executed. The status "0" indicates that transmission was successful.

The communication data can be reset to the initial state via "Clear".

Example 5:
In the event of an error: Reading a faulty object index on the connected IB IL AI4/EF (slot 3, index 0180)

Read request (master -> slave) - Start fragment

Data (4 words VC1)	Data structure
06 03 00 01 80 00 I 00 00	Read request (I/O)/Slot/Invoke ID/ Index high/Index low/Subindex/ 2 bytes unused

Write response (slave -> master)

Data (4 words VC1)	Data structure
86 44 06 07 00 00 I 00 00	Read response/Status/4 bytes of error code/ 2 bytes unused

Clear request (master -> slave)

Data (4 words VC1)	Data structure
00 I 00 00 00 00 00 00	Clear request/7 bytes unused

Clear response (slave -> master)

Data (4 words VC1)	Data structure
00 00 00 00 00 00 00 00	Clear response

Status byte 44_{hex} indicates an error during execution. In this case, the PCP read request is sent to the I/O terminal first. However, the I/O terminal does not recognize index 0180_{hex} and acknowledges it using error code 06 07. According to "Appendix" on page A-1, this means that this object does not exist. The last 2 bytes are also part of the PCP error data, however, they are not used in this example. If they do not equal zero, refer to the relevant device data sheet for more detailed information.

The communication data can be reset to the initial state via "Clear".

Example 6:
In the event of an error: Reading an object on a device without PCP function (slot 2, index 0080)

Read request (master -> slave)

Data (4 words VC1)	Data structure
06 02 00 00 80 00 I 00 00	Read request/Slot/Invoke ID/Index high/Index low/Subindex/2 bytes unused

Write response (slave -> master)

Data (4 words VC1)	Data structure
86 D2 00 00 00 00 00 00	Read response/Status/6 bytes unused

Clear request (master -> slave)

Data (4 words VC1)	Data structure
00 I 00 00 00 00 00 00 00	Clear request/7 bytes unused

Clear response (slave -> master)

Data (4 words VC1)	Data structure
00 00 00 00 00 00 00 00	Clear response

Since the addressed device does not support PCP, the request is rejected immediately. A message is sent by the bus coupler, which manages the PCP devices, whereby error code D2 stands for "Terminal does not have PCP". See also "Error codes for DP/V1 and VC1 communication" on page A-3. The communication data can be reset to the initial state via "Clear".

4 Dynamic configuration

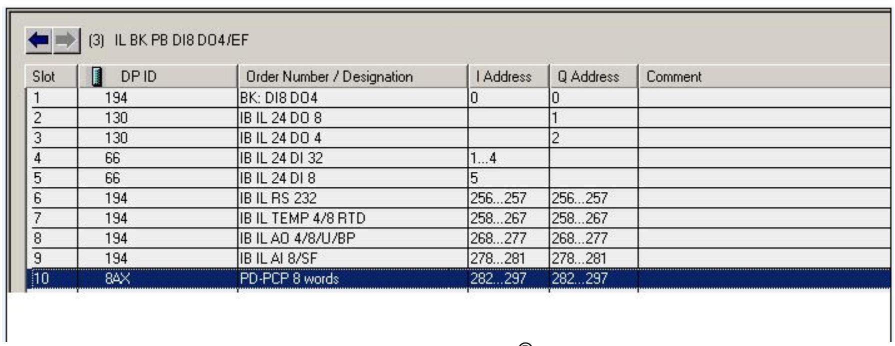
Dynamic configuration is the specification and configuration of a maximum configuration. Any subgroup of this maximum configuration can be operated.

In addition to dynamic configuration, empty spaces can be reserved for future extensions.

4.1 Empty spaces

It can be helpful to reserve empty spaces for a station, which may be used at different configuration levels. You can configure the maximum configuration level and thus also reserve memory in the PLC. However, optional terminals do not have to be connected. They can be deactivated in the configuration.

If the station is subsequently extended to include previously deactivated terminals, the new terminals can be connected and activated in the hardware configurator.



(3) IL BK PB DI8 DO4/EF

Slot	DP ID	Order Number / Designation	I Address	Q Address	Comment
1	194	BK: DI8 DO4	0	0	
2	130	IB IL 24 DO 8		1	
3	130	IB IL 24 DO 4		2	
4	66	IB IL 24 DI 32	1...4		
5	66	IB IL 24 DI 8	5		
6	194	IB IL RS 232	256...257	256...257	
7	194	IB IL TEMP 4/8 RTD	258...267	258...267	
8	194	IB IL AO 4/8/U/BP	268...277	268...277	
9	194	IB IL AI 8/SF	278...281	278...281	
10	8AX	PD-PCP 8 words	282...297	282...297	

Figure 4-1 Configuration table in the STEP 7® hardware configurator

The configuration, e.g., in STEP 7®, is carried out in the same way as for other modular slaves. The configuration can be created from the hardware catalog using drag & drop, see Figure 4-1.

Open the "Properties" dialog box by double-clicking on a terminal.

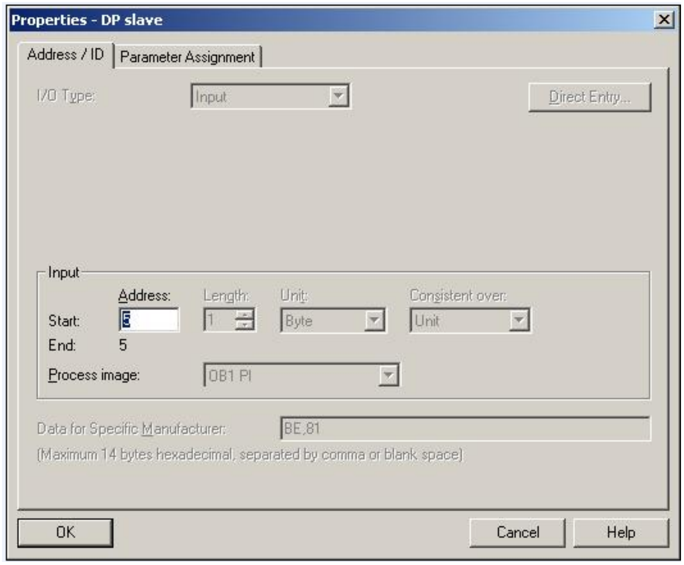


Figure 4-2 DI8 properties dialog box

The "Parameter Assignment" tab can be used to specify whether a device should be active or inactive, see Figure 4-3.

Please note that adjustments to the configuration and actual structure are also carried out for inactive terminals. A message is displayed if deactivated terminals are connected.

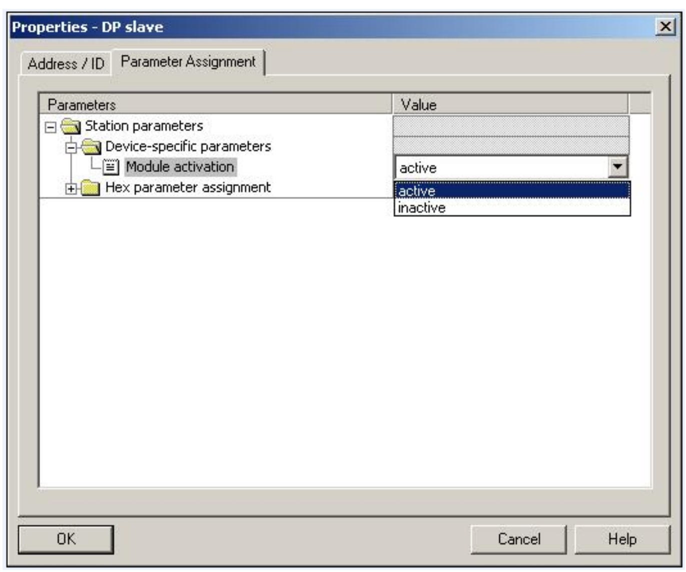


Figure 4-3 Activating/deactivating a terminal

Following activation/deactivation, the configuration can be saved, translated, and downloaded as usual.

Depending on the terminal type, substitute values (DO and AO) to be output in the event of an error can also be set at this point, for example. Furthermore, inputs (AI) can be parameterized. This is also carried out via the dialog box shown in Figure 4-3.

4.2 Principle of dynamic configuration

In dynamic configuration, a maximum configuration is specified during configuration. The addresses are thus reserved in the PLC. Any subgroup of this maximum configuration can be operated. This type of subgroup can be selected and activated during configuration and runtime. The advantage is that the application can divide an identical hardware configuration into active and inactive terminals.

Optional terminals that are only required for additional functions do not have to be connected. They can simply be deactivated by the application.

If a subsequent extension is planned, the application can activate the new devices. The only requirement is that they are part of the maximum configuration.

All settings are stored in the volatile memory. Thus easy replacement is ensured. The terminal does not have to be parameterized in advance.

Three indices on the bus coupler are used for handling:

Index 4: General control bits

Access: Write

Function: Details under "Slot 0" on page B-1. Please observe bit 0 and bit 3 during dynamic configuration.

Structure: Length of 1 bytes

Bit 0: Restart local bus

Bit 3: Unlock dynamic configuration

Index 6: Activation/deactivation of devices and slots

Access: Read and write

Function: Indicates which devices are active/inactive. Deactivation via the parameter telegram (reservation of empty spaces) is also indicated here.

Structure: Length of 8 bytes

Byte 1								Byte 2								Bytes 3 ... 7	Byte 8							
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	...	57	58	59	60	61	62	63	x

Bit = 0: Terminal and slot inactive

Bit = 1: Terminal and slot active

Index 7: Activation/deactivation of devices and slots and bus reset

Access:	Only via DP/V0, command 8
Function:	Indicates which devices are active/inactive. Deactivation via the parameter telegram (reservation of empty spaces) is also indicated here. In addition, the bus is reset.
Structure:	Length of 5 to 12 bytes, maximum

Byte 1:	08_{hex}							
Byte 2:	00_{hex}							
Byte 3:	07_{hex}							
Byte 4:	Length of data n							
Byte 5:	1	2	3	4	5	6	7	8
...								
Byte 4+n:	x	x	x	x	x	x	x	x

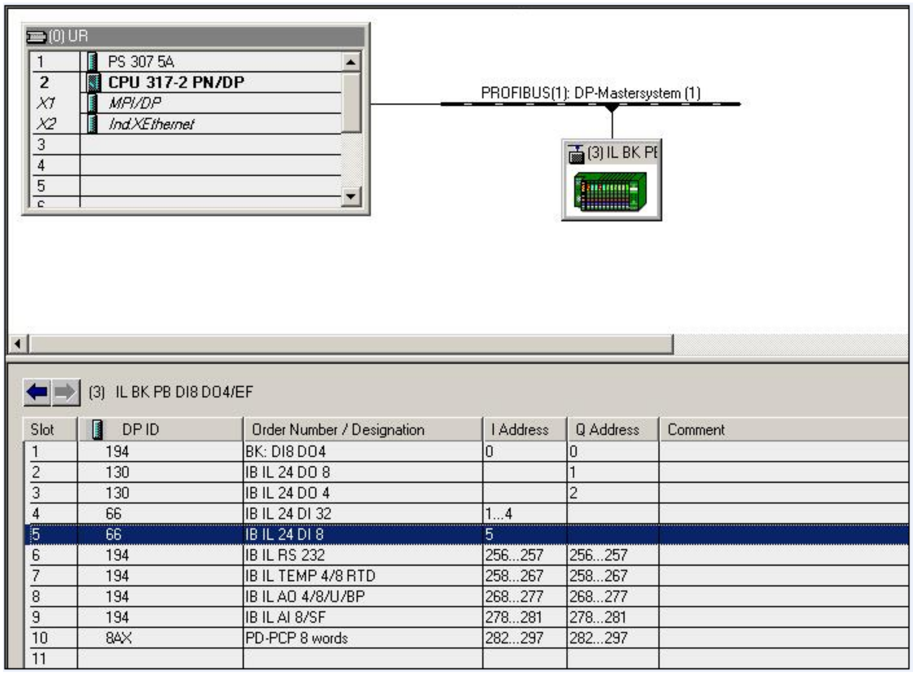
Bit = 0: Terminal and slot inactive

Bit = 1: Terminal and slot active

4.3 Startup

4.3.1 Planning configuration

Figure 4-4 shows an example of the maximum configuration, as provided. All terminals are activated by default.



Slot	DP ID	Order Number / Designation	I Address	Q Address	Comment
1	194	BK: DI8 DO4	0	0	
2	130	IB IL 24 DO 8		1	
3	130	IB IL 24 DO 4		2	
4	66	IB IL 24 DI 32	1...4		
5	66	IB IL 24 DI 8	5		
6	194	IB IL RS 232	256...257	256...257	
7	194	IB IL TEMP 4/8 RTD	258...267	258...267	
8	194	IB IL AO 4/8/U/BP	268...277	268...277	
9	194	IB IL AI 8/SF	278...281	278...281	
10	8AX	PD-PCP 8 words	282...297	282...297	
11					

Figure 4-4 Configuration in the STEP 7® hardware configurator

Access to indices 4 and 6 described in the introduction is either via PROFIBUS DP/V1 or via the process data interface. Index 7 is intended to be used if no DP/V1 master exists or if both the specification and the bus restart are to be implemented using a single service in the process data interface. If the process data interface is used, the effort required is minimized.

For access via process data, configure the VC1 process data interface. This is the "PD PCP x words" (x = 4, 6, .., 16) module. It can be specified as the first or last module in the station and the data width and address can be selected according to the options in the CPU.

The remaining terminals can be configured as usual.

DOWNLOADED FROM WWW.SCATTS.CO.UK

4.3.2 Options for specifying the active configuration

In the following example, the 8-channel digital terminals are not part of the station, i.e., these devices are part of the maximum configuration, but are to be deactivated at this station.

There are two options for startup:

1 Deactivating terminals in the hardware configuration by default
   - This option can be implemented easily. After setting the "Module activation" parameter to "inactive", the configuration can be downloaded to the control system as usual, see "Activating/deactivating a terminal" on page 4-2.

2 Via DP/V1 or DP/V0 with maximum configuration and final specification in the application
   - Specify the maximum configuration in the hardware configuration and download it.
   - Switch to cyclic data exchange.
   - On the I/O terminals, no data is exchanged as long as the connected configuration does not correspond to the available, activated terminals.
   - Specify the configuration connected to the station in a non-volatile memory via index 6 or index 7.
   - Data exchange is started.

For the second option, the example from the figure on page 4-6 is used and the DI8 as well as the DO8 are deactivated via DP/V0 and DP/V1. In each case, the data to be written is indicated. For additional information about the protocol, please refer to Section "Acyclic communication in DP/V1 mode" on page 3-4 and Section "Acyclic communication in DP/V0 mode via process data" on page 3-13. Here, you will find further examples for communication via DP/V0 and DP/V1. The individual indices are explained in Section "Slot 0" on page B-1.

Which option will be used in the end depends on the support of DP/V1 in the master. If DP/V1 is not (or inadequately) supported by the master, DP/V0 should be used. For SIMATIC® STEP 7® CPUs, functions are available that have been prepared by Phoenix Contact.

4.3.3 Specifying the active configuration via DP/V0

For DP/V0, the activation status can be accessed via Index 6 or 7. The structure of the objects differs slightly, therefore typical access is illustrated for both objects.

Access via Index 6

Please observe the structure of index 6 when accessing it:

Byte 1								Byte 2								Bytes 3 ... 7	Byte 8							
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	...	57	58	59	60	61	62	63	x

Bit = 0: Terminals and slot inactive

Bit = 1: Terminals and slot active

In this example there are 7 connected terminals. Terminals 1, 2, 3, 4, and 7 should be active. Terminals 5 and 6 are inactive.

The resulting data for object 6 is as follows:

$F2_{hex}$, 00_{hex}, 00_{hex}, 00_{hex}, 00_{hex}, 00_{hex}, 00_{hex}, 00_{hex}

These 8 data bytes are described below as an example for two possible data widths of the VC1 device.

> If a new activation status is described via index 6 or index 7, it only becomes valid when the bus is restarted.

Two corresponding examples:

If, as described in this example, the configuration and the available terminals differ, because, e.g., two terminals are not connected, the bus is read in continuously until it is matches the configuration. The activation status on index 6 is evaluated automatically.

If the active bus configuration is extended and if previously inactive terminals are attached at the end, these terminals are first activated via index 6. Next time the bus is started, the new terminals are integrated into the data traffic. The bus can be started via index 4, bit 0. Please note that during a restart the terminal output data is reset to its terminal-specific reset values.

If the active bus configuration is extended and previously inactive terminals are attached at the end, terminal activation and bus restart can be implemented simultaneously via index 7. The new terminals are integrated into data traffic. Please note that during a restart the terminal output data is reset to its terminal-specific reset values.

4 words VC1

Write request (master -> slave) - Start fragment

Data (4 words VC1)	Data structure
14 00 00 00 06 00 08 F2	Write/Slot/Invoke ID/Index high/Index low/Subindex/Length/1 byte of data

Write response (slave -> master)

Data (4 words VC1)	Data structure
14 I 00 00 00 00 00 00 00	Write response/7 bytes unused

Write request (master -> slave) - End fragment

Data (4 words VC1)	Data structure
40 00 00 00 00 00 00 00	Write/7 bytes of data

Write response (slave -> master)

Data (4 words VC1)	Data structure
84 I 00 00 00 00 00 00 00	Write response/Status/6 bytes unused

8 words VC1

Write request (master -> slave) - Start fragment

Data (8 words VC1)	Data structure
14 00 00 00 06 00 08 F2 00 00 00 00 00 00 00 I 00	Write/Slot/Invoke ID/Index high/Index low/Subindex/Length/8 bytes of data/1 byte unused

Write response (slave -> master)

Data (8 words VC1)	Data structure
14 I 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	Write response/15 bytes unused

Access via index 7

Please observe the structure of index 7 when accessing it:

Byte 1	08_{hex}							
Byte 2	00_{hex}							
Byte 3	07_{hex}							
Byte 4	Length of data n							
Byte 5	1	2	3	4	5	6	7	8
...								
Byte 4+n	x	x	x	x	x	x	x	x

Bit = 0: Terminal and slot inactive

Bit = 1: Terminal and slot active

In this example there are 7 connected terminals. Terminals 1, 2, 3, 4, and 7 should be active. Terminals 5 and 6 are inactive.

The resulting data for object 7 is as follows:
08_{hex}, 00_{hex}, 07_{hex}, 01_{hex}, $F2_{hex}$

As the fourth byte for index 7 contains the length, the number of bytes to be transmitted is only as many as required for the terminals to be activated/deactivated.

These 5 data bytes are described below as an example:

4 words VC1

Write request (master -> slave) - Start fragment

Data (4 words VC1)	Data structure
08 00 07 01 F2 I 00 00 00	Read/write/Index high/Index low/Length/ 1 byte of data/3 bytes unused

Write response (slave -> master)

Data (4 words VC1)	Data structure
88 00 02 05 F2 I 00 00 00	Read/write response/Status/Length/ Number of available terminals/Status of the terminals/3 bytes unused

4.3.4 Specifying the active configuration via DP/V1

For DP/V1 the activation status can be accessed via index 6. Index 7 is not supported here because it was specifically optimized for access via the process data interface.

Access via index 6

Please observe the structure of index 6 when accessing it:

Byte 1								Byte 2								Bytes 3 ... 7	Byte 8							
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	...	57	58	59	60	61	62	63	x

Bit = 0: Terminal and slot inactive

Bit = 1: Terminal and slot active

In this example there are 7 connected terminals. Terminals 1, 2, 3, 4, and 7 should be active. Terminals 5 and 6 are inactive.

The resulting data for object 6 is as follows:
$F2_{hex}$, 00_{hex}, 00_{hex}, 00_{hex}, 00_{hex}, 00_{hex}, 00_{hex}, 00_{hex}

These 8 data bytes are transmitted via DP/V1 in the following.

If a new activation status is described via index 6, it only becomes valid when the bus is restarted.

Two corresponding examples:

If, as described in this example, the configuration and the available terminals differ, because, e.g., two terminals are not connected, the bus is read in continuously until it is matches the configuration. The activation status on index 6 is evaluated automatically.

If the active bus configuration is extended and if previously inactive terminals are attached at the end, these terminals are first activated via index 6. Next time the bus is started, the new terminals are integrated into the data traffic. The bus can be started via index 4, bit 0. Please note that during a restart the terminal output data is reset to its terminal-specific reset values.

Write request (master -> slave)

Data (8 words VC1)	Data structure
5F 00 06 08 F2 00 00 00 00 00 00 00	Write/Slot/Index/Total length of data/ Length/8 bytes of object data

Write response (slave -> master)

Data (8 words VC1)	Data structure
5F 00 06 08	Write/Slot/Index/Length

4.3.5 Summary

Depending on the task and requirements in the control system, the illustrated services (process data interface DP/V0 or DP/V1) can be used to specify the configuration actually used by the application. In this example, the IB IL 24 DI24 and IB IL 24 DO8 terminals are deactivated with the illustrated services and therefore must not be connected. For additional information and examples regarding communication via the process data interface as well as DP/V1, please refer to Section "Acyclic communication (DP/V1 and PCP)" on page 3-1.

The "PD PCP x words" process data interface module does not have to be configured for normal operation. It is only required if you wish to access objects such as index 6 via the process data.

If you wish to use the "PD PCP x words" process data interface module (VC1), it can be configured in any position. However, it is recommended that the last position be used so that the configured slot and the actual slot will always be the same.

5 PROFIsafe application notes

The maximum number of Inline terminals that can be snapped onto the bus coupler depends on the following parameters:

- Maximum number of Inline bus devices that can be snapped on: 62
- Maximum length of the process data channel: 244 bytes
- Maximum length of the parameter channel: 237 bytes

Make sure that no parameters are overwritten.

- When operating safety terminals on the bus coupler, it is recommended that status PDU diagnostics are activated. In this operating mode, the device and parameter errors are forwarded transparently by the safety terminals to the corresponding control system.

> The GSD file (electronic device data sheet) is required to configure PROFIBUS devices. This file contains all the safety modules that are currently available.
>
> Make sure you always use the latest GSD file. It can be found on the Internet at www.phoenixcontact.com.
>
> In the unlikely event that the GSD file does not yet contain the configuration data for a new safety module, this can be adjusted manually.
>
> If you have any queries regarding adjusting the GSD file, please contact the Technical Support Center.

The bus coupler supports Inline PROFIsafe modules. These PROFIsafe modules can be operated together with safe and non-safe I/O devices. At present, the following PROFIsafe modules are available:

- IB IL 24 PSDI 8-PAC
- IB IL 24 PSDO 8-PAC
- IB IL 24 PSDOR 4-PAC
- IB IL 24 PSDO 4/4-PAC

The modules map 4 words of process data to the local bus and obtain the F-Parameters and iParameters from the parameter telegram via a 63-byte parameter block. The settings are provided in the GSD file.

> Connected PROFIsafe modules need 30 seconds before they can communicate and be parameterized.

6 IO-Link

In contrast to individual signal wiring, which was previously primarily used, IO-Link uses a 3-wire connection to sensors and actuators. This means that in addition to transmitting a simple switching signal, bidirectional serial communication is also possible. IO-Link is also suitable for mixed operation. If an interface does not support IO-Link, the device automatically switches to SIO mode (Standard Input/Output).

6.1 IO-Link call

Direct communication between the IO-Link client (PROFIBUS DP master) and the IO-Link server (IB IL IOL4 DI2-PAC Inline IO-Link master) can be established. There can be multiple IO-Link masters in an Inline station (depending on the amount of process data).

IOLD objects are accessed with the IO-Link call being a standardized read and write access. It uses the mechanisms of PROFIBUS I&M functions (via DP/V1).

For further information on IO-Link call communication can be found in the AH FLS PB M12 IOL 4 M12 application note.

A Appendix

A 1 Error codes for PCP communication

Table A-1 $06_{hex}/02_{hex}$ (Hardware fault)

Meaning	Access to the object failed due to a hardware fault.
Cause	E.g., I/O voltage not present.
Remedy	Remove the hardware fault.

Table A-2 $06_{hex}/03_{hex}$ (Object access denied)

Meaning	The object has limited access rights.
Cause	It may be a read-only object or it may be password-protected.
Remedy	Check the access rights in the object description.

Table A-3 $06_{hex}/05_{hex}$ (Object attribute inconsistent)

Meaning	A service parameter was specified with an impermissible value.
Cause	E.g., an incorrect length specification or an impermissible subindex.
Remedy	Check the parameters in the object description and send the service again with the corrected values.

Communication error messages

Table A-4 $06_{hex}/06_{hex}$ (Object access unsupported)

Meaning	The service used cannot be applied to this object.
Cause	E.g., a program sequence can be started or stopped, but not read.
Remedy	Check the object description to determine which services are supported for this object.

Table A-5 $06_{hex}/07_{hex}$ (Object non-existent)

Meaning	The object does not exist.
Cause	The "Index" parameter probably contains an invalid value.
Remedy	Check the object index in the object description and send the service again.

Other error messages

Table A-6 $08_{hex}/00_{hex}$ (Application error)

Meaning	Terminal-specific error message; not a communication error
Cause	–
Remedy	Refer to your terminal description.

Depending on the I/O terminal, other specific error codes may also be used. These codes are listed in the relevant data sheet.

A 2 Error codes for DP/V1 and VC1 communication

Always observe the individual represenations in your working environment.

DP/V1 errors:

Function code (response) = DE_{hex} (error read)

Or function code (response) = DF_{hex} (error write)

Error decode = 80_{hex} (DP/V1 communication)

Errors with reference to I/O terminal:

Status 44_{hex} indicates an error
- For DP/V1 on byte 3 of the data block
- For VC1 byte 2 in the response

Table A-7 Error codes for DP/V1 and VC1 communication

Error_Code_1	Error_Code_2	Meaning
$B0_{hex}$	0	Index invalid
$B1_{hex}$	0	Invalid data length when writing
$B2_{hex}$	0	Invalid device number
$B5_{hex}$	0	Status conflict, last read/write not finished yet
$B6_{hex}$	0	Access to device or index not permitted
$B7_{hex}$	0	Invalid parameter
$C3_{hex}$	0	(Internal) resource not available
$D1_{hex}$	0	PCP connection not established
$D2_{hex}$	0	PCP not supported
$D4_{hex}$	0	Incorrect service code
$D5_{hex}$	0	Incorrect sequence of fragments
$D6_{hex}$	0	Incorrect data length during access
$D7_{hex}$	0	PCP PDU size of 64 bytes (58 bytes of user data) per communication exceeded
DC_{hex}	0	(Internal) timeout while reading
DD_{hex}	0	(Internal) error while sending a request
DE_{hex}	0	(Internal) error while receiving a service

A 3 Error codes for F-Parameters

F-Parameters contain information to adapt the PROFIsafe layer to particular customer specifications. Parameterization is also checked on a separate path of the PROFIsafe layer.

For Status-PDU diagnostics the following parameter errors can be distinguished in the "PROFIsafe F-Parameter error" status type:

Table A-8 F-Parameter parameter errors

Error code		Error cause	Remedy
dec	**hex**		
64	40	The parameterized F_Destination_Address does is not match the PROFIsafe address set at the safety module (F module).	Match the PROFIsafe address of the safety module and the value in F_Destination_Address.
65	41	Invalid parameterization of F_Destination_Address. Addresses 0000_{hex} and $FFFF_{hex}$ are not permitted.	Correct value.
66	42	Invalid parameterization of F_Source_Address. Addresses 0000_{hex} and $FFFF_{hex}$ are not permitted.	Correct value.
67	43	Invalid parameterization of F_WD_Time. A monitoring time of 0 ms is not permitted.	Correct value.
68	44	Invalid parameterization of F_SIL. The required SIL cannot be supported by the safety module (F module).	Use device with the required SIL.
69	45	Invalid parameterization of F_CRC_Length. The CRC length generated by the safety module (F module) does not correspond to the required length.	Check device description.
70	46	Version of F-Parameter set invalid. Version of the safety module (F module) does not match the required version.	Check device description.
71	47	The checksum determined by the safety module (F module) from the PROFIsafe parameters (CRC1) does not match the CRC1 transmitted with the parameter telegram.	Check F-Parameters, repeat calculation

Please refer to the corresponding data sheets or user manuals for detailed information on the iParameters of the safety modules used.

A 4 Format of the parameter telegram

This section provides a detailed description of the format of the parameters for the bus coupler and the I/O terminals. This may be useful when setting parameters using acyclic services or if there is no user interface for the simple selection of parameters.

Table A-9 Parameters for the bus coupler

Byte	Bit	Meaning
Bytes 1 to 7		DP standard
Bytes 8 to 10		DP/V1 standard
Byte 11		Control byte
	Bit 7	0: Hide channel-specific diagnostics 1: Show channel-specific diagnostics
	Bit 6	0: Do not rotate DI32 and DO32 data 1: Rotate DI32 and DO32 data
	Bit 5	0: SET_PRM does not overwrite dynamic configuration 1: SET_PRM overwrites dynamic configuration
	Bit 4	0: Do not rotate DI16 and DO16 data 1: Rotate DI16 and DO16 data
	Bit 3	Reserved (set to 0)
	Bit 2	Reserved (set to 0)
	Bit 1	0: Automatically acknowledge peripheral faults 1: Manually acknowledge peripheral faults
	Bit 0	0: Stop local bus in the event of terminal failure 1: Operate local bus with available terminals

The data for the configuration and the failsafe value can be found in the terminal-specific data sheets.

Table A-10 Parameters for the I/O devices byte 1

Byte	Bit	Meaning
Byte 1	Bit 7 to bit 6	00: Start block ID for device
	Bit 5 to bit 4	Configuration
		00: No configuration (e.g., DO)
		01: Permanent configuration
		10: Temporary configuration
	Bit 3 to bit 2	Failsafe value
		00: No failsafe value (e.g., DI)
		01: Output zero
		10: Maintain value
		11: Apply value from data field
	Bit 1	Extended functions
		0: No function block
		1: Function block (also several)
	Bit 0	Activation
		0: Activated
		1: Disabled

Following a configuration block:

Byte x	Bit 7 to bit 6	01: Configuration block ID
	Bit 5 to bit 0	Length of the data block
Byte x+1 to n		n data bytes

Following a failsafe block:

Byte x	Bit 7 to bit 6	10: Failsafe value block ID
	Bit 5 to bit 0	Length of the data block
Byte x+1 to n		n data bytes

If one or more function blocks are present, a header byte for the function block follows:

Byte x	Bit 7 to bit 6	11: Function block ID
	Bit 5 to bit 0	Entire function block length
		An additional byte is automatically used if the total length of the function block is exceeded. This byte specified the actual length of the function block. The first byte (header byte for the function blocks) is set to FF_{hex}.

After that the function blocks are decoded:

a) PCP function block

Byte x+1	Bit 7 to bit 6	01: PCP function block
	Bit 5 to bit 0	PCP function block length
Byte x+2		Invoke ID
Byte x+3		Index high byte
Byte x+4		Index low byte
Byte x+5		Subindex
Byte x+6 to x+6+n		n data bytes

b) Additional function block

Byte x+1	Bit 7 to bit 6	10: Additional function block
	Bit 5 to bit 0	Length of the additional function block
Byte x+2	Bit 0	1: Rotate I/O data permitted 0: Rotate I/O data not permitted
	Bit 7 to bit 1	Reserved

c) ID function block (n <= 16)

Byte x+1	Bit 7 to bit 6	11: ID function block
	Bit 5 to bit 0	Length of the ID function block
Byte x+2	Bit 7 to bit 0	Order No. (last - n sign), ASCII
Byte x+2+n	Bit 7 to bit 0	Order No. (last sign), ASCII

B Object dictionaries

Due to the complexity of the station, the station is divided into slots 0 to 63. The slots represent the individual slots of the station. Slot 0 represents the bus coupler, slot 1 the integrated inputs and outputs on the bus coupler, and slots 2 to 63 the additional bus devices.

To parameterize the station, each slot offers one or more object references, which can be used to acyclically parameterize the bus coupler and the bus devices. In the following section, these objects are explained in more detail in reference to the slots.

B 1 Slot 0

The following station-specific indices are implemented on slot 0:

Index 3:	**PDU length**
Function:	Data width of the virtual C1 module (process data interface for acyclic communication) in bytes
Access:	Read
Length:	1 byte
Note:	For DP/V0 access only

Index 4:	**Master control**
Function:	Control bits for station
Access:	Write
Length:	1 byte
Structure:	Bit 0: Restart local bus
	Bit 1: Acknowledge PF
	Bit 2: Update diagnostics (evaluation of index 18 on PCP terminals)
	Bit 3: Unlock dynamic configuration
	Bit 4: Reserved (set to 0)
	Bit 5: Reserved (set to 0)
	Bit 6: Reserved (set to 0)
	Bit 7: Reserved (set to 0)

Bit 0 can be used to restart the local bus at any time. If a new activation status has been specified via index 6, it is used for this restart. Please note that during a restart the terminal output data is reset to its terminal-specific reset values.

Bit 1 can be used to reset peripheral faults that have to be acknowledged. This function has been prepared for future terminals.

Set bit 2 if the diagnostics of all connected terminals are to be read in again. This is only useful if terminals are connected for which object 18_{hex} (diag state) is implemented. Usually this includes all devices that use PCP. Therefore check in the relevant data sheet whether the terminal supports PCP.

Bit 3 can be used in the context of dynamic configuration. If a new activation status is specified via object 6 or 7, it is mandatory. If the connection to the PROFIBUS master is then interrupted and if the original parameterization is transmitted by the master during restart, the activation status is maintained. The dynamic configuration does not have to be executed again.

However, if the activation status is reset during a restart, set bit 3 once following dynamic configuration via index 6 or 7.

In this way, both options are available:

– Restart with original configuration
– Restart with last dynamic configuration

Index 5: **PCP status**

Function: Position and communication status of PCP terminal

Access: Read

Length: 3 bytes per PCP terminal -> 48 bytes, maximum

Structure: Byte 1: Position in the station (slot number)

Byte 2: Status of PCP connection

– 00_{hex} No connection
– 01_{hex} Connection OK
– FF_{hex} Error during connection establishment

Byte 3: Reserved

Index 6: **Terminal activation**

Function: Activation/deactivation of I/O terminals

Access: Read and write

Length: 8 bytes

Structure: See below

Byte 1								Byte 2								Bytes 3 ... 7	Byte 8							
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	...	57	58	59	60	61	62	63	x
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0									

Bit = 0: Terminal and slot inactive

Bit = 1: Terminal and slot active

Index 7: **Terminal activation/restart**

Function: Activation/deactivation of I/O terminals, bus restart (write), number of terminals, and status (read)

Access: Simultaneous read and write access via DP/V0 with command 08

Length: 3-byte header, 1-byte length, up to n = 8 bytes for up to 63 terminals

Structure: See below

Write:

Byte 1	08_{hex}							
Byte 2	00_{hex}							
Byte 3	07_{hex}							
Byte 4	Length of the following data n							
Byte 5	1	2	3	4	5	6	7	8
...								
Byte 4+n	x	x	x	x	x	x	x	x

Bit = 0: Terminal and slot inactive

Bit = 1: Terminal and slot active

Read (write response):

Byte 1	88_{hex}							
Byte 2	Status							
Byte 3	Length of the following data m							
Byte 4	Number of available terminals (k)							
Byte 5	1	2	3	4	5	6	7	8
...								
Byte 3+m	x	x	x	x	x	x	x	x

Bit = 0: Terminal and slot inactive

Bit = 1: Terminal and slot active

This object is specifically intended for use under DP/V0. The data length for write access always depends on the number of configured terminals. The number of activated/deactivated terminals is completely irrelevant. If, for example, 15 terminals were configured, 4 of which are to be deactivated and 11 of which are to be activated, only the number of configured terminals is relevant. In this case, 2 bytes are required in order to display 15 terminals. If 25 terminals are configured, 4 bytes are required. In the case of write access, at least as many bytes must be used as are necessary for the activation status for each device to be transmitted. More bytes can also be transmitted, however only up to 8 bytes, as the station can only manage a maximum of 63 terminals.

In the case of read access, the amount of data is based on the number of configured and available devices. If, for example, k = 18 devices are configured, you will receive the response m = 3+1 bytes.

The following is valid for bytes in position 5 and up:
Status and specification are the same:

– Activated and connected: Write: "1“; Read: "1"
– Not activated and not connected: Write: "0“; Read: "0"

Status and specification inconsistent:

– Activated but not connected: Write: "1"; Read: „0“
– Not activated but connected: Write "0"; Read: "1"

As the specification must match the connected terminals, by comparing the desired state in the output data and the real status in the input data in addition to simply specifying the activation, you can determine which terminals are present or not.

An example for access to index 7 is described in Section 4.3.3 „Specifying the active configuration via DP/V0“.

Index 12:	**Terminal diagnostics**
Function:	Station diagnostics byte 7 and up (without standard diagnostics)
Access:	Read
Length:	72 bytes, maximum
Structure:	As described in Section "Diagnostics" on page 2-1

Terminal diagnostics can only be used via the acyclic services of DP/V1.

Index 20:	**Cycle count**
Function:	Cycle counter (all cycles)
Access:	Read and write
Length:	8 bytes
Note:	For write access, all counters (index 20 to 25) are set to 0

Index 21:	**Cycle error count**
Function:	Cycle counter (all faulty cycles)
Access:	Read and write
Length:	8 bytes
Note:	For write access, all counters (index 20 to 25) are set to 0

Index 22: **ID cycle count**
Function: Cycle counter (all ID cycles)
Access: Read and write
Length: 8 bytes
Note: For write access, all counters (index 20 to 25) are set to 0

Index 23: **ID cycle error count**
Function: Cycle counter (all faulty ID cycles)
Access: Read and write
Length: 8 bytes
Note: For write access, all counters (index 20 to 25) are set to 0

Index 24: **Data cycle count**
Function: Cycle counter (all data cycles)
Access: Read and write
Length: 8 bytes
Note: For write access, all counters (index 20 to 25) are set to 0

Index 25: **Data cycle error count**
Function: Cycle counter (all faulty data cycles)
Access: Read and write
Length: 8 bytes
Note: For write access, all counters (index 20 to 25) are set to 0

The objects on slot 0 can be read and written with a single access attempt via DP/V1.

Index 255: **Identification & Maintenance functions**

Function: Read and write I & M functions

Access: Read and write

Length: 64 bytes

Note: Only possible via DP/V1

For this object, each access attempt - read or write - should be implemented in two stages (in accordance with specification IEC 61158-6, Section 6.1).

Each PROFIBUS device is equipped with an electronic name plate for unique identification. This nameplate can be accessed, for example, during startup or for maintenance purposes. Depending on the selected I & M function, either only read or read/write access is possible.

The following I & M function are supported:

Table B-1 I & M functions (basic data)

I & M basic data	Access	Description
Header		
Manufacturer specification	Read	IL PB BK DIO
I & M block		
MANUFACTURER_ID	Read	Manufacturer identification Phoenix Contact = $00B0_{hex}$
ORDER_ID	Read	Terminal order number
SERIAL_NUMBER	Read	Production serial number for unique identification
HARDWARE_REVISION	Read	Revision of hardware $XXXX_{hex}$
SOFTWARE_REVISION	Read	Version of software $XXXX_{hex}$
REVISION_COUNTER	Read	Number of modified versions $XXXX_{hex}$
PROFILE_ID	Read	$F600_{hex}$
PROFILE_SPECIFIC_TYPE	Read	0003_{hex}
IM_VERSION	Read	0101_{hex}
IM_SUPPORTED	Read	$001F_{hex}$

Tabelle 0-1 I & M functions (option)

I & M1 (option)	Access	Description
Header		
Manufacturer specification	Read	IL PB BK DIO
I & M block		
TAG_FUNNCTION	Read/write	Specify a system-wide unique identification for the terminal here.
TAG_LOCATION	Read/write	Enter the installation location of the terminal here.

Table B-2 I & M2 functions (option)

I & M2 (option)	Access	Description
Header		
Manufacturer specification	Read	IL PB BK DIO
I & M block		
INSTALLATION_DATE	Read/write	Specify the date when the terminal was installed here.
RESERVED	Read/write	Not specified at the moment.

Table B-3 I & M3 functions (option)

I & M3 (option)	Access	Description
Header		
Manufacturer specification	Read	IL PB BK DIO
I & M block		
DESCRIPTOR	Read/write	Specify a general comment on the terminal here.

Table B-4 I & M4 functions (option)

I & M4 (option)	Access	Description
Header		
Manufacturer specification	Read	IL PB BK DIO
I & M block		
SIGNATURE (security)	Read/write	The hardware configurator can be used to store a security code as a reference for certain parameterizations. The signature parameters allow access to the terminal together with the system identification, consisting of MANUFACTURER_ID, ORDER_ID and SERIAL_ID.

Example:

Read I & M1

1. a) Send the request as a DP/V1 write request (I & M read call) to slot 0
   b) Receive DP/V1 write response
2. a) Send a DP/V1 read to slot 0
   b) Receive DP/V1 read response

Write I & M1

1. a) Send the request as a DP/V1 write request (I & M write call) to slot 0
   b) Receive DP/V1 write response
2. a) Send a DP/V1 read to slot 0
   b) Receive DP/V1 read response

Please note, that when writing to I & M functions I & M1 to I & M4, the nonvolatile memory of the device is accessed. The memory is designed for a maximum of 100,000 write access operations.

B 2 Slot 1

On slot 1, indices are implemented with regard to the integrated DI8:

Index 13:	**PD IN**
Function:	Input data of the integrated DI8
Access:	Read
Length:	1 byte

B 3 Slots 2 to 63

On slots 2-63, indices are implemented with regard to the I/O terminals that can be connected:

Index 13:	**PD IN**
Function:	Input data on the slot of connected terminals
Access:	Read
Length:	0 to 128 bytes

Index 47:	**PCP access**
Function:	Read and write PCP data on connected terminals via DP/V1
Access:	Read and write
Length:	Depends on the command and the PCP object
Structure:	See Section 3.4, page 3-4 and onwards.
Note:	During access via DP/V0, the PCP objects can be accessed directly, i.e., without being diverted via index 47.

SCATTERGOOD & JOHNSON LTD

ELECTRICAL ENGINEERING & FLUID CONTROL DISTRIBUTORS

Est.1899

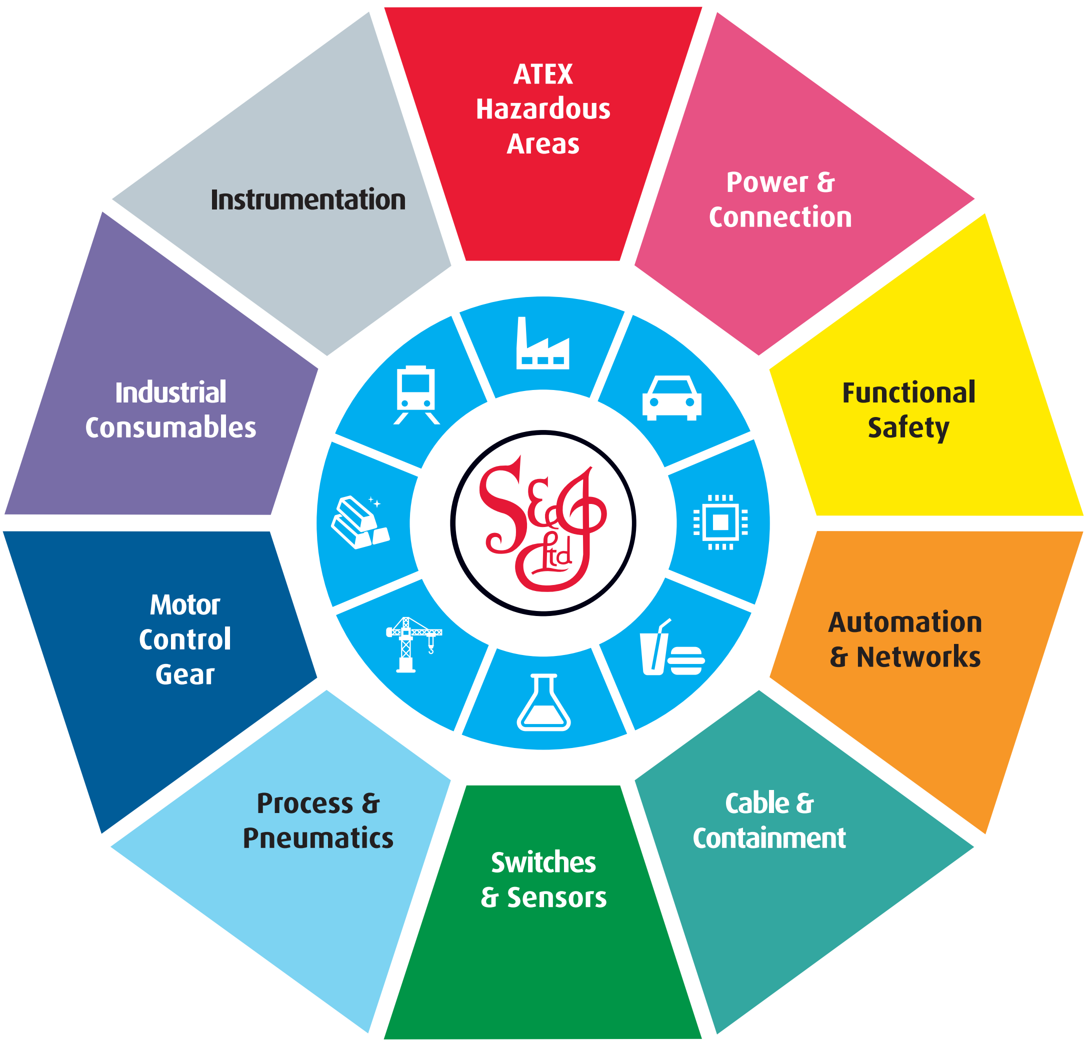
At Scattergood & Johnson Ltd, we pride ourselves on being a technical distributor to specialist industries.

Working with a range of quality product suppliers across a number of specialist markets, we are not your average 'box shifter' - we are your technical and supply chain partner.

We fully support every product we sell - for free! Our internal team and external sales engineers can answer any product or application question, no matter the complexity.

Backing up this technical ability is a range of 50,000+ products available from stock for nationwide next day delivery (same day if required!), or you can collect what you need from any of our trade counters around the UK.

Select your specialist interest below to learn more about how we can help.



Online, In Branch and On the Road - Scattergood & Johnson Ltd, there when you need us.

www.scatts.co.uk