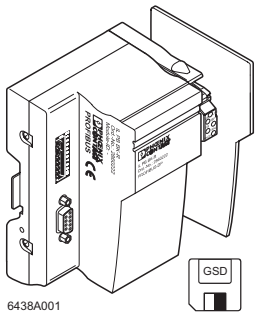
IL PB BK - Module Formats

Description of Module Formats in the GSD File for the PROFIBUS DP Bus Interface Module



6438A001

User Reference 6488A 11/2001

This user reference is only valid in association with the "Configuring and Installing the PROFIBUS DP Bus Interface Module for the Inline Product Range" User Manual IL PB BK UM E.

This user reference includes the basics of the IL PB BK UNIMOD-S7 (Use of a Universal Module on the PROFIBUS DP Bus Interface Module Under SIMATIC® STEP 7®) and the IL PB BK UNIMOD-S7 (2) ("Packing" Process Data on the PROFIBUS DP Bus Interface Module Under SIMATIC® STEP 7®) user references.

All documentation is also available on the Internet at: **http://www.phoenixcontact.com**.

Modules in the INTERBUS Inline product range are referred to in the following as automation terminals (terminals).

Introduction

This user reference is aimed at persons who want to configure an INTERBUS Inline station, with modules that are not yet integrated in the GSD (device database) file, using the PROFIBUS DP Bus Interface Module.
It provides the user with information about the meaning of the individual hexadecimal codes in the GSD file for the module descriptions. It is assumed the user has basic knowledge of the GSD file.

PROFIBUS devices are always equipped with a GSD file. The GSD file contains general information on the manufacturer, the hardware and software version, and supported baud rates. In modular slaves, the GSD file also contains data about the number and type of I/O in the supported modules (terminals).

The module format of each module is defined by a module designation and a specific hexadecimal code.

Module Format of a GSD File

The IL PB BK GSD file contains a list of various modules in different categories. The first distinction is made between digital and analog modules. Further distinctions are then made within these areas. The IL PB BK GSD file, for example, can be opened, processed, and saved with a GSD editor. The following explains how the entries are defined in a GSD file. Particular attention is paid to the module formats and their definition.

"Conventions" for the Module Formats in the GSD File

The following shows an entry in the GSD file regarding a module format:

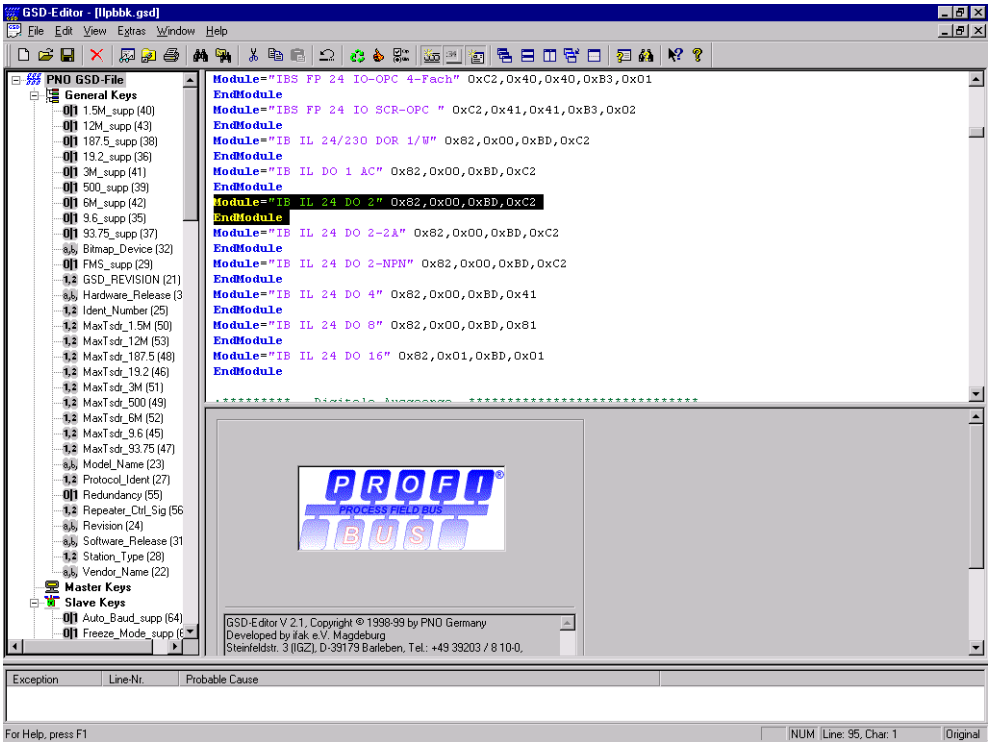


Figure 1 A GSD editor with the IB IL 24 DO 2 terminal selected as an example

Module = "IB IL 24 DO 2" 0x82,0x00,0xBD,0xC2

EndModule

The definition of the module begins with the term "Module". An "=" follows, which introduces a more detailed description of the module.

A precise description of the module is given in “inverted commas” after the equals sign. The example used here ("IB IL 24 DO 2") indicates a digital output terminal with two channels.

A precise description of the terminal in hexadecimal codes follows the terminal designation. These codes, which describe the module format, are based on a convention, by which all formats can be specified. For digital output terminals, there are four hexadecimal values separated by commas:

"0x82,0x00,0xBD,0xC2".

The first hexadecimal number is a special identification format. The identification format provides information on the type of terminal, e.g., whether it is an input or an output terminal. In this example, "0x82" indicates an output terminal.

The second hexadecimal number ("0x00") defines the consistency and length of the data.

The third hexadecimal code "0xBD" indicates the ID code of the individual, configured terminal.

In this example, the last code "0xC2" indicates the "length code" of the terminal.

The definition of the individual terminals ends with the term "EndModule".

The individual meanings of these hexadecimal codes can be clearly explained using the various categories of the individual terminals. The module formats are used in the configuration of the individual terminals.

The terminal configurations are used within the framework of configuration telegrams for PROFIBUS DP.

Structure of a Configuration Telegram

PROFIBUS DP distinguishes between the following telegram types: parameterization, configuration, diagnostics, control commands, and user data traffic. After parameterization, the master must send a configuration telegram to the corresponding slave. The master causes the slave to check the sent configuration against the stored configuration.

PROFIBUS DP configuration telegrams have the following basic structure:

SD	LE	LEr	SD	DA	SA	FC	DSAP	SSAP	DU	...	FCS	ED
68h	xx	xx	xx	8x	8x	xx	3Eh	3Eh	xx	...	16h	16h

The top row contains the standard symbolic name of the byte. The bottom row contains the value entered in hexadecimal notation. The "xx" values can vary. The value 3Eh in bytes "DSAP" and "SSAP" indicates that this is a configuration telegram.

The configuration itself can be found in the "DU" byte, provided a standard identification format is used and the input/output data description is combined in one byte.

With a configuration in a standard identification format it is possible to describe up to 16 bytes or words in one byte of the DU. Inputs and outputs with the same formats can be combined in one byte. Otherwise the number of bytes used depends on the number of different bytes/words you want to use, which can be combined in one byte.

If the data description is not to be or cannot be written in one byte, e.g., because the input and output data are not the same length, several bytes are required in the "DU" field of the description.

In special identification format, several bytes are always used for device descriptions in the "DU" field (except for an empty slot). If several devices are connected to a modular system, a description for every module appears in the "DU" block of the configuration telegram.

Only one configuration telegram is available for the entire INTERBUS Inline station. Therefore each station has a limit of 244 bytes.

Meaning of the Configuration Byte in Standard Identification Format

The "DU" byte has the following meaning for the standard format, which is often used in practice:

MSB							LSB
7	6	5	4	3	2	1	0

Bit No.	Value	Meaning
3-0	XX	(Number - 1) of bytes/words in the module, e.g., 5 words correspond to entry = 4
5/4	00	Special identification format
5/4	01	Input module
5/4	10	Output module
5/4	11	Combined I/O module
6	0	Unit is bytes
6	1	Unit is words
7	0	Consistency over 1 byte or 1 word
7	1	Consistency over the total length

Example of a Standard Identification Format

Module With Outputs

Byte No.	Value of the Byte [Bit]	Meaning
1	1010 0001	Output module with 2 bytes of process data, consistency over the total length

Bit 7 is set, which specifies that the data consistency covers the data of the entire module. Bit 6 is equal to 0, which means that the data width is written in bytes. Bits 5/4 =10_{bin} indicate that the module is an output module. In this example, bits 3-0 are equal to $0001_{bin} = 1_{dec}$. Because "Number - 1" is entered for process data bytes/words, the module has 2 data bytes that are to be transmitted consistently.

Module With Inputs

Byte No.	Value of the Byte [Bit]	Meaning
1	0001 0000	Input module with 1 byte of process data, byte consistency

In this example, bits 5/4 = 01_{bin} indicate that the module is an input module. Bit 6 is equal to 0, which means that the length is specified in bytes in bits 3-0. In this example, the value $0000_{bin} = 0_{dec}$ is entered in bits 3-0. The length is thus 1 byte. 0 in bit 7 indicates consistency over the data unit, which in this example is "bytes".

Module With Inputs and Outputs

Byte No.	Value of the Byte [Bit]	Meaning
1	1111 0011	I/O module with 4 words of process data, consistency over the total length

Bit 5/4 = 11 indicates that the module is an input/output module. Bit 6 indicates that the length is specified in words in bits 3-0. $0011_{bin} = 3_{dec}$ is entered in bits 3-0. The module thus has 4 words, which according to the specification in bit 7 must be transmitted consistently over the entire module.

Meaning of the Configuration Byte in Special Identification Format

In special identification format, additional bytes are transmitted per module. This enables more information about the connected modules to be transmitted. For the PROFIBUS DP Bus Interface Module this refers to the length and ID codes of the connected modules.

The first byte contains information on whether inputs and/or outputs are available and how many manufacturer-specific bytes follow the I/O data description.

Configuration Telegram Structure for a Module With Input or Output Data

Byte 1	Byte 2	Byte 3	Byte 4
Inputs or outputs	Description of the input or output data	ID code	Length code

In the case of a **module with input or output data**, the second byte provides information on the consistency, the unit of length (words or bytes) and the number of input or output words or bytes. The third byte already contains the ID code, while the fourth byte contains the length code.

Configuration Telegram Structure for a Module With Input and Output Data

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Inputs and outputs	Description of output data	Description of input data	ID code	Length code

In the case of a **module with input and output data**, the second byte describes the data in the output direction. The third byte contains information on the data in the input direction. The fourth byte contains the module ID code, while the fifth contains the length code.

Structure of Configuration Byte 1

MSB							LSB
7	6	5	4	3	2	1	0

Bit No.	Value	Meaning
3-0	xx	Number of manufacturer-specific bytes
5+4	00	Always "00" in special identification format
7/6	00	Free slot
7/6	01	The following byte describes the input data.
7/6	10	The following byte describes the output data.
7/6	11	1 byte each of input/output data description follows.

Structure of Configuration Bytes 2 or 2 and 3

The description for the output or input data format in byte 2 or byte 2 and byte 3 has the following format:

MSB							LSB
7	6	5	4	3	2	1	0

Bit No.	Value	Meaning
5-0	xx	(Number - 1) of bytes or words in the module
6	0	Unit is bytes
6	1	Unit is words
7	0	Consistency over 1 byte or 1 word
7	1	Consistency over the entire module

A. Special Identification Format for Individual Modules

The special identification format is used for all modules in the GSD file for the PROFIBUS DP Bus Interface Module. In addition to the individual module descriptions, several modules can also be combined in one description. This will be explained in more detail later. First, some examples will illustrate how an individual module is generally mapped in the GSD file.

Output Module

Byte No.	Value of the Byte [Bit]	Meaning
1	1000 0010	Special identification format for **one** output module
2	kwnn nnnn	Consistency, unit, and length of the output process data
3	iiii iiii	ID code of the module (see data sheet)
4	llll llll	Length code of the module (see data sheet)

Explanations:

k = 0 indicates consistency over the data unit (w)

k = 1 indicates consistency over the entire process data area of the terminal

w = 0 indicates the length specification is in bytes

w = 1 indicates the length specification is in words (2 bytes)

n = (Number - 1) of bytes or words in binary notation

i = Wildcard for the module ID code

l = Wildcard for the module length code

Example: IB IL DO 4

4 bits of output data, ID code BD_{hex},
length code 41_{hex} -> 0x82,0x00,0xBD,0x41

Byte No.	Value of the Byte [Bit]	Meaning
1	**1000 0010**	Special identification format for **one** output terminal
2	0000 0000	Consistency over the unit, unit is bytes, length 1
3	1011 1101	ID code of the terminal: BD_{hex}
4	0100 0001	Length code of the terminal: 41_{hex}

Example: IB IL AO 1/SF

1 word of output data, ID code $7D_{hex}$,
length code 01_{hex} -> 0x82,0x40,0x7D,0x01

Byte No.	Value of the Byte [Bit]	Meaning
1	1000 0010	Special identification format for **one** output terminal
2	0100 0000	Consistency over the unit, unit is words, length 1
3	0111 1101	ID code of the terminal: $7D_{hex}$
4	0000 0001	Length code of the terminal: 01_{hex}

Input Module

Byte No.	Value of the Byte [Bit]	Meaning
1	0100 0010	Special identification format for **one** input module
2	kwnn nnnn	Consistency, unit, and length of the input process data
3	iiii iiii	ID code of the module (see data sheet)
4	llll llll	Length code of the module (see data sheet)

Explanations:

k = 0 indicates consistency over the data unit (w)

k = 1 indicates consistency over the entire process data area of the terminal

w = 0 indicates the length specification is in bytes

w = 1 indicates the length specification is in words (2 bytes)

n = (Number - 1) of bytes or words in binary notation

i = Wildcard for the module ID code

l = Wildcard for the module length code

Example: IB IL 24 DI 16

16 bits of input data, ID code BE_{hex},
length code 01_{hex} -> 0x42,0x01,0xBE,0x01

Byte No.	Value of the Byte [Bit]	Meaning
1	0100 0010	Special identification format for **one** input terminal
2	0000 0001	Consistency over the unit, unit is bytes, length 2
3	1011 1110	ID code of the terminal: BE_{hex}
4	0000 0001	Length code of the terminal: 01_{hex}

Combined I/O Module

Byte No.	Value of the Byte [Bit]	Meaning
1	1100 0010	Special identification format for **one** combined I/O module
2	kwnn nnnn	Consistency, unit, and length of the output process data
3	kwnn nnnn	Consistency, unit, and length of the input process data
4	iiii iiii	ID code of the module (see data sheet)
5	llll llll	Length code of the module (see data sheet)

Explanations:

k = 0 indicates consistency over the data unit (w)

k = 1 indicates consistency over the total length of the process data area

w = 0 indicates the length specification is in bytes

w = 1 indicates the length specification is in words (2 bytes)

n = (Number - 1) of bytes or words in binary notation

i = Wildcard for the module ID code

l = Wildcard for the module length code

Example: IB IL 400 ELR 1-3A

1 byte of input and output data, ID code BF_{hex}, length code 81_{hex} -> 0xC2,0x00,0x00,0xBF,0x81

Byte No.	Value of the Byte [Bit]	Meaning
1	1100 0010	Special identification format for **one** combined I/O terminal
2	0000 0000	Output data: consistency over the unit, unit is bytes, length 1
3	0000 0000	Input data: consistency over the unit, unit is bytes, length 1
4	1011 1111	ID code of the terminal: BF_{hex}
5	1000 0001	Length code of the terminal: 81_{hex}

Example: IB IL CNT

2 words of input and output data, ID code BF_{hex},
length code 02_{hex} -> 0xC2,0xC1,0xC1,0xBF,0x02

Byte No.	Value of the Byte [Bit]	Meaning
1	1100 0010	Special identification format for **one** combined I/O terminal
2	1100 0001	Output data: consistency over the total length, unit is words, length 2
3	1100 0001	Input data: consistency over the total length, unit is words, length 2
4	1011 1111	ID code of the terminal: BF_{hex}
5	0000 0010	Length code of the terminal: 02_{hex}

B. Combining Identical Modules to Form a Virtual Module

Combining modules (packing) to form a virtual module is particularly useful if you want to save memory and address space in the control system. Adjacent modules with the same ID and length code, with a single address requirement in one direction of no more than 4 bits, can be combined to save space.

Even modules that are not the same but whose process data length is greater than or equal to 8 bits can be combined. This is particularly useful if the same combination of specific modules is configured repeatedly. This module combination need only be defined once in the GSD and can then be used repeatedly in a process. This saves time and reduces configuration errors.

Please refer to the user reference IL PB BK UNIMOD-S7 (2) ("Packing" Process Data on the PROFIBUS DP Bus Interface Module Under SIMATIC® STEP 7®).

Virtual Module Consisting of Packed, Identical Output Modules

Byte No.	Value of the Byte [Bit]	Meaning
1	1000 0011	Special identification format for packed, identical output modules
2	kwnn nnnn	Consistency, unit, and length of the output process data
3	iiii iiii	ID code of the module (see data sheet)
4	llll llll	Length code of the module (see data sheet)
5	mmmm mmmm	Number of packed modules

Explanations:

k = 0 indicates consistency over the data unit (w)

k = 1 indicates consistency over the entire process data area of the virtual module

w = 0 indicates the length specification is in bytes

w = 1 indicates the length specification is in words (2 bytes)

n = (Number - 1) of bytes or words in binary notation

i = Wildcard for the module ID code

l = Wildcard for the module length code

m = Number of modules packed in binary notation

Example: 3 x IB IL 24 DO 2

3 x 2 bits of output data, ID code BD_{hex},
length code $C2_{hex}$ -> 0x83,0x00,0xBD,0xC2,0x03

Byte No.	Value of the Byte [Bit]	Meaning
1	1000 0011	Special identification format for packed, identical output terminals
2	0000 0000	Consistency over the unit, unit is bytes, length 1
3	1011 1101	ID code of the terminal: BD_{hex}
4	1100 0010	Length code of the terminal: $C2_{hex}$
5	0000 0011	Number of packed terminals: 3_{dec}

Virtual Module Consisting of Packed, Identical Input Modules

Byte No.	Value of the Byte [Bit]	Meaning
1	0100 0011	Special identification format for packed, identical input modules
2	kwnn nnnn	Consistency, unit, and length of the input process data
3	iiii iiii	ID code of the module (see data sheet)
4	llll llll	Length code of the module (see data sheet)
5	mmmm mmmm	Number of packed modules

Explanations:

k = 0 indicates consistency over the data unit (w)

k = 1 indicates consistency over the entire process data area of the virtual module

w = 0 indicates the length specification is in bytes

w = 1 indicates the length specification is in words (2 bytes)

n = (Number - 1) of bytes or words in binary notation

i = Wildcard for the module ID code

l = Wildcard for the module length code

m = Number of modules packed in binary notation

Example: 2 x IB IL 24 DI 4

2 x 4 bits of input data, ID code BE_{hex}
length code 41_{hex} -> 0x43,0x00,0xBE,0x41,0x02

Byte No.	Value of the Byte [Bit]	Meaning
1	0100 0011	Special identification format for packed, identical input terminals
2	0000 0000	Consistency over the unit, unit is bytes, length 1
3	1011 1110	ID code of the terminal: BE_{hex}
4	0100 0001	Length code of the terminal: 41_{hex}
5	0000 0010	Number of packed terminals: 2_{dec}

Virtual Module Consisting of Packed, Identical I/O Modules

Byte No.	Value of the Byte [Bit]	Meaning
1	1100 0011	Special identification format for packed, identical I/O modules
2	kwnn nnnn	Consistency, unit, and length of the output process data
3	kwnn nnnn	Consistency, unit, and length of the input process data
4	iiii iiii	ID code of the module (see data sheet)
5	llll llll	Length code of the module (see data sheet)
6	mmmm mmmm	Number of packed modules

Explanations:

k = 0 indicates consistency over the data unit (w)

k = 1 indicates consistency over the entire process data area of the virtual module

w = 0 indicates the length specification is in bytes

w = 1 indicates the length specification is in words (2 bytes)

n = (Number - 1) of bytes or words in binary notation

i = Wildcard for the module ID code

l = Wildcard for the module length code

m = Number of modules packed in binary notation

Example: 2 x IB L2 BOX 24 DIO 2/2/4,

2 x 4 bits of I/O data, ID code $B3_{hex}$,
length code 41_{hex} -> 0xC3,0x00,0x00,0xB3,0x41,0x02

Byte No.	Value of the Byte [Bit]	Meaning
1	1100 0011	Special identification format for packed, identical I/O terminals
2	0000 0000	Output data: consistency over the unit, unit is bytes, length 1
3	0000 0000	Input data: consistency over the unit, unit is bytes, length 1
4	1011 0011	ID code of the terminal: $B3_{hex}$
5	0100 0001	Length code of the terminal: 41_{hex}
6	0000 0010	Number of packed terminals: 2_{dec}

C. Combining Non-Identical Modules to Form Virtual Modules

Packing non-identical modules to form a virtual module is particularly useful in two situations.

1. To save address and memory space by combining modules with a single data width of no more than 4 bits in a single module. In this way the bytes are used to their full potential.
2. To reduce the configuration effort for similar repetitive structures and to avoid configuration errors.

Do not combine more than seven packed Inline terminals.

Virtual Module Consisting of Packed, Non-Identical Output Modules

Byte No.	Value of the Byte [Bit]	Meaning
1	1000 xxxx	Special identification format for packed, non-identical output modules
2	kwnn nnnn	Consistency, unit, and length of the output process data
3	iiii iiii	ID code of the first module (see data sheet)
4	llll llll	Length code of the first module (see data sheet)
5	iiii iiii	ID code of the second module (see data sheet)
6	llll llll	Length code of the second module (see data sheet)
y-1	iiii iiii	ID code of the last packed module (see data sheet)
y	llll llll	Length code of the last packed module (see data sheet)

Explanations:

x = number of modules multiplied by 2, with a maximum of seven modules, this gives: x = a maximum of 14 in binary notation

k = 0 indicates consistency over the data unit

k = 1 indicates consistency over the entire process data area of the virtual module

w = 0 indicates the length specification is in bytes

w = 1 indicates the length specification is in words (2 bytes)

n = (Number - 1) of bytes or words in binary notation

i = Wildcard for the module ID code

l = Wildcard for the module length code

y = Number of bytes in special identification format

Example: IB IL 24 DO 2 and IB IL 24 DO 4

2 bits and 4 bits of output data, ID code BD_{hex} for each,
length code $C2_{hex}$ and 41_{hex} -> 0x84,0x00,0xBD,0xC2,0xBD,0x41

Byte No.	Value of the Byte [Bit]	Meaning
1	1000 0100	Special identification format for two packed, non-identical output terminals
2	0000 0000	Consistency over the unit, unit is bytes, length 1
3	1011 1101	ID code of the first terminal: BD_{hex}
4	1100 0010	Length code of the first terminal: $C2_{hex}$
5	1011 1101	ID code of the second terminal: BD_{hex}
6	0100 0001	Length code of the second terminal: 41_{hex}

Virtual Module Consisting of Packed, Non-Identical Input Modules

No.	Value of the Byte [Bit]	Meaning
1	0100 xxxx	Special identification format for packed, non-identical input modules
2	kwnn nnnn	Consistency, unit, and length of the input process data
3	iiii iiii	ID code of the first module (see data sheet)
4	llll llll	Length code of the first module (see data sheet)
5	iiii iiii	ID code of the second module (see data sheet)
6	llll llll	Length code of the second module (see data sheet)
y-1	iiii iiii	ID code of the last packed module (see data sheet)
y	llll llll	Length code of the last packed module (see data sheet)

Explanations:

x = number of modules multiplied by 2, with a maximum of seven modules, this gives:
x = a maximum of 14 in binary notation

k = 0 indicates consistency over the data unit

k = 1 indicates consistency over the entire process data area of the virtual module

w = 0 indicates the length specification is in bytes

w = 1 indicates the length specification is in words (2 bytes)

n = (Number - 1) of bytes or words in binary notation

i = Wildcard for the module ID code

l = Wildcard for the module length code

y = Number of bytes in special identification format

Example: IB IL 24 DI 8, IB IL 24 DI 4, and IB IL 24 DI 2,

8 bits, 4 bits, and 2 bits of input data, ID code BE_{hex} for each,
length code 81_{hex}, 41_{hex} and $C2_{hex}$ -> 0x46,0x01,0xBE,0xB1,0xBE,0x41,0xBE,0xC2

Byte No.	Value of the Byte [Bit]	Meaning
1	0100 0110	Special identification format for three packed, non-identical input modules
2	0000 0001	Consistency over the unit, unit is bytes, length 2
3	1011 1110	ID code of the first terminal: BE_{hex}
4	1000 0001	Length code of the first terminal: 81_{hex}
5	1011 1110	ID code of the second terminal: BE_{hex}
6	0100 0001	Length code of the second terminal: 41_{hex}
7	1011 1110	ID code of the third terminal: BE_{hex}
8	1100 0010	Length code of the third terminal: $C2_{hex}$

Virtual Module Consisting of Packed, Non-Identical Input and Output Modules

Byte No.	Value of the Byte [Bit]	Meaning
1	1100 xxxx	Special identification format for packed, non-identical I/O modules
2	kwnn nnnn	Consistency, unit, and length of the output process data
3	kwnn nnnn	Consistency, unit, and length of the input process data
4	iiii iiii	ID code of the first module (see data sheet)
5	llll llll	Length code of the first module (see data sheet)
6	iiii iiii	ID code of the second module (see data sheet)
7	llll llll	Length code of the second module (see data sheet)
y-1	iiii iiii	ID code of the last packed module (see data sheet)
y	llll llll	Length code of the last packed module (see data sheet)

Explanations:

x = number of modules multiplied by 2, with a maximum of seven modules, this gives:
x = a maximum of 14 in binary notation

k = 0 indicates consistency over the data unit

k = 1 indicates consistency over the entire process data area of the virtual module

w = 0 indicates the length specification is in bytes

w = 1 indicates the length specification is in words (2 bytes)

n = (Number - 1) of bytes or words in binary notation

i = Wildcard for the module ID code

l = Wildcard for the module length code

y = Number of bytes in special identification format

Example: IB IL 24 DO 4 and IB IL 24 DI 2

4 bits of output data, 2 bits of input data, ID code BD_{hex} and BE_{hex},
length code 41_{hex} and $C2_{hex}$ -> 0xC4,0x00,0x00,0xBD,0x41,0xBE,0xC2

Byte No.	Value of the Byte [Bit]	Meaning
1	1100 0100	Special identification format for packed, non-identical I/O terminals
2	0000 0000	Output data: consistency over the unit, unit is bytes, length 1
3	0000 0000	Input data: consistency over the unit, unit is bytes, length 1
4	1011 1101	ID code of the first terminal: BD_{hex}
5	0100 0001	Length code of the first terminal: 41_{hex}
6	1011 1110	ID code of the second terminal: BE_{hex}
7	1100 0010	Length code of the second terminal: $C2_{hex}$

Ordering Data for Documentation

Using the information on the meanings of the individual hexadecimal codes in the GSD file for the described modules, you should now be able to configure an Inline station, with modules that have not yet been integrated in the GSD file using the PROFIBUS DP Bus Interface Module. For more detailed information, please refer to the following documentation.

Description	Order Designation	Order No.
Configuring and Installing the PROFIBUS DP Fieldbus Coupler for the Inline Product Range (English)	IL PB BK UM E	26 98 10 6
CD-ROM with all Inline, Loop 2, and other data sheets	CD IBS DB ELDOC	27 45 60 6
Use of a Universal Module on the PROFIBUS DP Bus Interface Module Under SIMATIC® STEP 7®	IL PB BK UNIMOD-S7	90 06 25 2
"Packing" Process Data on the PROFIBUS DP Bus Interface Module Under SIMATIC® STEP 7®	IL PB BK UNIMOD-S7 (2)	90 08 76 9

Phoenix Contact GmbH & Co. KG
Flachsmarktstr. 8
32825 Blomberg
Germany

+49 - 52 35 - 30 0

+49 - 52 35 - 34 12 00

www.phoenixcontact.com

Worldwide locations:
www.phoenixcontact.com/salesnetwork

SCATTERGOOD & JOHNSON LTD
ELECTRICAL ENGINEERING & FLUID CONTROL DISTRIBUTORS
Est.1899

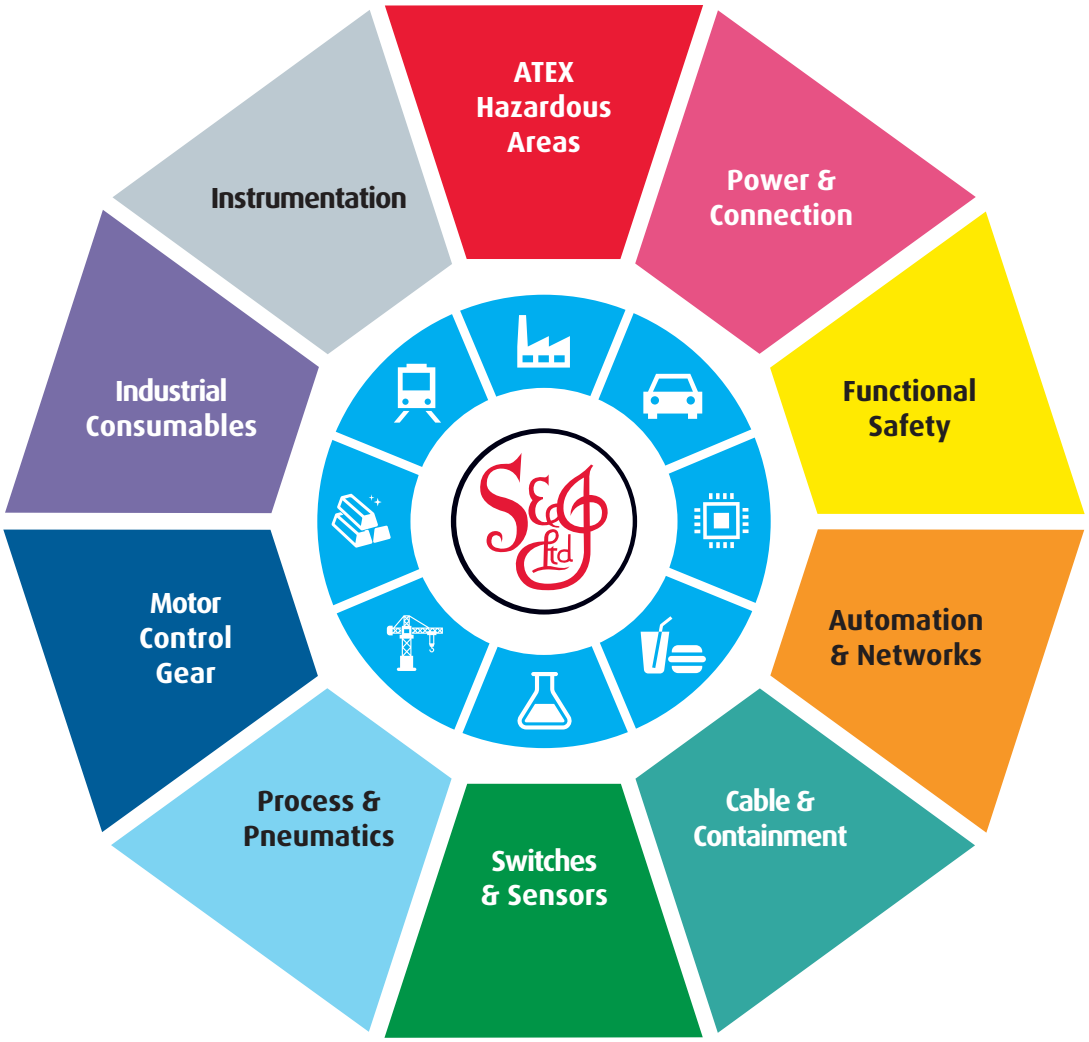
At Scattergood & Johnson Ltd, we pride ourselves on being a technical distributor to specialist industries.

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