

**PSS(1) IBS-S PCP**

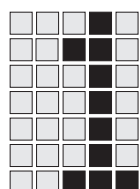


Programmable control systems PSS®

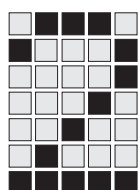
All rights to this documentation are reserved by Pilz GmbH & Co. KG. Copies may be made for internal purposes.

Suggestions and comments for improving this documentation will be gratefully received.

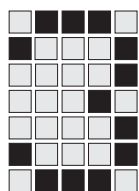
Pilz<sup>®</sup>, PIT<sup>®</sup>, PMI<sup>®</sup>, PNOZ<sup>®</sup>, Primo<sup>®</sup>, PSEN<sup>®</sup>, PSS<sup>®</sup>, PVIS<sup>®</sup>, SafetyBUS p<sup>®</sup>, SafetyEYE<sup>®</sup>, SafetyNET p<sup>®</sup>, the spirit of safety<sup>®</sup> are registered and protected trademarks of Pilz GmbH & Co. KG in some countries.



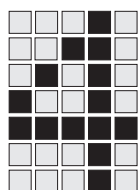
<b>Introduction</b>	1-1
<b>Definition of symbols</b>	1-1



<b>Overview</b>	2-1
<b>Front view</b>	2-2



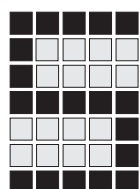
<b>Safety</b>	3-1
<b>Intended application</b>	3-1
<b>Safety regulations</b>	3-1
Safety guidelines	3-1
External supply for the Interbus-S slave	3-2



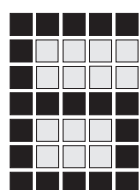
<b>Interbus-S</b>	4-1
<b>Overview</b>	4-1
<b>Configuring the Dual-Port RAM</b>	4-4
<b>Setting the process and parameter data length</b>	4-6
<b>Initialising the Interbus-S slave</b>	4-8
<b>Transferring process data</b>	4-9
Transferring process data without SB 202	4-9
Transferring process data with SB 202	4-11
Error messages in XW 67	4-16
<b>Transferring parameter data</b>	4-17
Operating system objects	4-19
User objects	4-25
Error messages on Interbus	4-35

## Contents

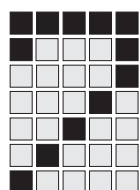
---



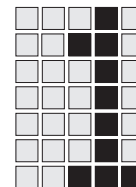
<b>Operation</b>	5-1
Commissioning	5-1
Faults	5-2
Display elements	5-2



<b>Technical Details</b>	6-1
Module code	6-2
Identification code	6-2



<b>Appendix</b>	7-1
-----------------	-----



# Introduction

---

This manual explains the function and operation of the PSS(1) IBS-S PCP module. It contains important information and should be retained for future reference.

The manual is divided into the following chapters:

- 1) Introduction
- 2) Intended application, safety regulations  
Describes the module's application areas and provides information about the safety precautions to be taken during installation and operation.
- 3) Overview  
Provides information about the most significant features of the PSS(1) IBS-S PCP module.
- 4) Interbus-S  
Describes the exchange of data between the module and Interbus-S, via the process data channel and parameter data channel.
- 5) Operation  
Explains how to commission the module and advises on what to do if a fault occurs.
- 6) Technical details
- 7) Appendix

## Definition of Symbols

Information in this manual which is of particular importance can be identified as follows:



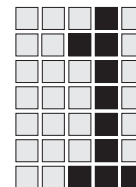
### **DANGER!**

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



### **WARNING!**

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



# Introduction

---



## **CAUTION!**

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



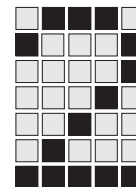
## **NOTICE**

This describes a situation in which the unit(s) could be damaged and also provides information on preventive measures which can be taken.



## **INFORMATION**

This gives advice on applications and provides information on special features, as well as highlighting areas within the text which are of particular importance.



## Overview

---

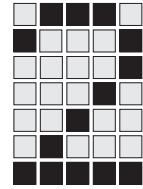
For non-safety-related applications, the PSS(1) IBS-S PCP module can be incorporated into the remote bus as an Interbus-S slave. This means that data can be transferred via both the process and the parameter data channel. The Interbus-S slave meets the requirements of the Interbus-S Club, details of which can be found in the Appendix.

The Interbus-S slave can be fed from the PSS power supply or from a separate external supply. If the module is connected to an external power supply it will automatically use this as its source.

The external supply voltage on the PSS and Interbus-S slave means that the Interbus-S can be operated even when no voltage is applied to the PSS. However, if there is no power to the PSS, it will not be possible to exchange data with the Interbus-S slave.

The PSS(1) IBS-S PCP module can be used as a replacement for its predecessor, the PSS(1) IBS-S. Both modules are fully compatible, provided the rotary switch for setting the parameter data length on the PSS(1) IBS-S PCP is set to “30x 150” and the Interbus-S slave is fed from the PSS power supply. It is not necessary to amend the user program unless you wish to take advantage of the option to use an external power supply for the Interbus-S slave, in which case you should replace SB 202 (IBS:SLAV) with SB 202 (IBS:SL\_2) or program your own check for data consistency.

# Overview



## Front view

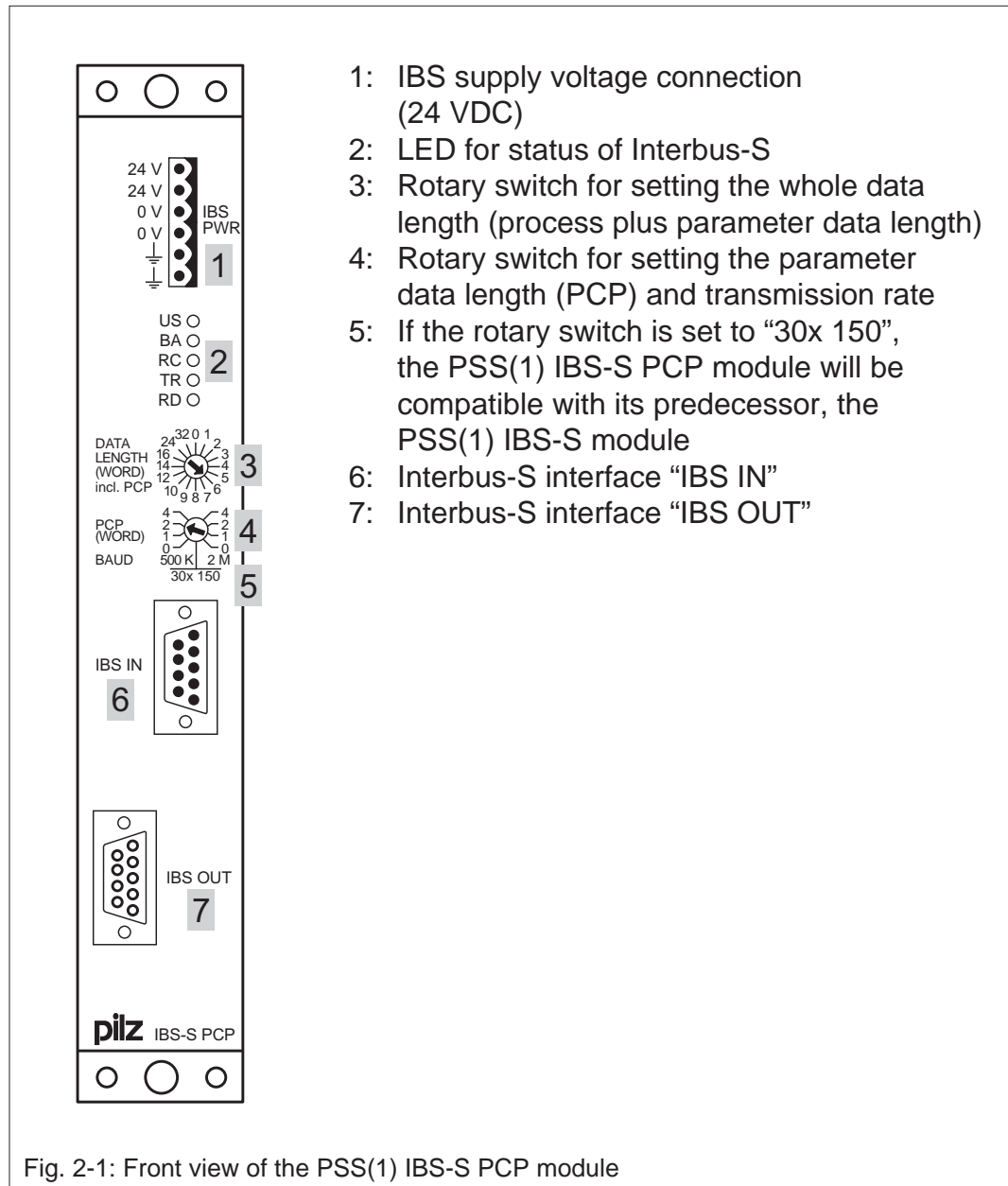
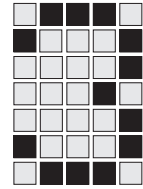


Fig. 2-1: Front view of the PSS(1) IBS-S PCP module





## Intended application

The PSS(1) IBS-S PCP module is a passive subscriber (slave) on Interbus-S. It can be incorporated into the remote bus for non-safety-related applications.

The operating system must be able to support the functionality of the module.

The transfer of process data is supported by the following CPUs:

- PSS CPU
- PSS1 CPU
- PSS SF CPU
- PSS1 SF CPU
- PSS SB CPU
- PSS1 SB CPU
- P9 CPU
- P10 CPU

SB 202 (IBS:SL\_2) can be used to program the transfer of process data. SB 202 (IBS:SLAV) can also be used, but in this case users will need to program their own check for data consistency (this is only important if the Interbus-S slave is to be fed from an external power supply "IBS PWR").

The transfer of parameter data is not supported by the P9 CPU, P10 CPU, PSS SF CPU and PSS1 SF CPU.

On the remaining CPU types in the PSS 3000/3100 series, the transfer of parameters is only supported from the following version numbers:

- PSS CPU from Version 2.1
- PSS1 CPU from Version 1.5
- PSS SB CPU from Version 1.0
- PSS1 SB CPU from Version 1.0

The PSS IBS-S PCP module is designed for use on the module rack for the PSS 3000 series.

The PSS1 IBS-S PCP module is designed for use on the module rack for the PSS 3100 series.

This module is only designed for use in an industrial environment.

Problems of interference could arise if used within a domestic environment.

# Safety

---

## Safety regulations

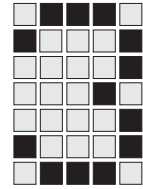
### Safety guidelines

- Before installing and commissioning the module you should read the “PSS 3000/3100 Installation Manual” and “System Manual for the PSS-Range”, or “Pitronik P9 Installation Manual”/“Pitronik P10 Installation Manual” and “CPU System Function (P10)”.
- Switch off the supply voltage before swapping cables and connecting devices.
- Keep as large a distance as possible between the module and any sources of electromagnetic interference (especially frequency converters) or screen the source of interference using a bulkhead.
- Data and supply lines should be installed separately in order to avoid inductive and capacitive transmission (recommended minimum distance: 10 cm).
- Always use metallic or metallised synthetic plugs for Interbus-S remote bus connections.
- The shielded connection on the Interbus-S remote bus cable must be used on both sides of the connector housing.
- Shielded, twisted pair cables must be used for connections to the Interbus-S interfaces.
- Always refer to the Interbus-S specifications provided by Phoenix Contact.
- Always keep within the specifications given under “Technical details” in the Appendix.

### External supply for the Interbus-S slave

The connection of a separate, external power supply for the Interbus-S slave is optional. If you wish to connect an external power supply, follow the instructions below:

- The external power supply must be 24 V. Always check that the voltage from the external supply matches this value.
- To achieve the lowest possible residual ripple ( $< \pm 1.2$  V), we recommend that you install a three-phase bridge rectifier or regulated supply.



- The fuse between the external power supply and the module protects the cabling. The size of the fuse will depend on the cable cross section and on local regulations.
- Always remove the power to modules before they are installed or removed.



- **WARNING!**

Safe electrical isolation must be ensured for the external 24 V supply. Failure to do so could result in electrical shock. Power supplies must conform to DIN VDE 0551 or EN 60742 and DIN VDE 0160.

- The external power supply should be connected to the module as shown in the diagram below.

There must be no direct connection between “N” and the 0 V output on the external power supply.

The 0 V connections on all the 24 V power supplies should be connected and the 0 V mains earthed at a single point. The connection of the 0 V supply to the central bar or earth fault monitor must be in accordance with the relevant national regulations/standards (such as EN 60 204, VDE 0113, NFPA 79-1991:17-7, NEC: Article 250).

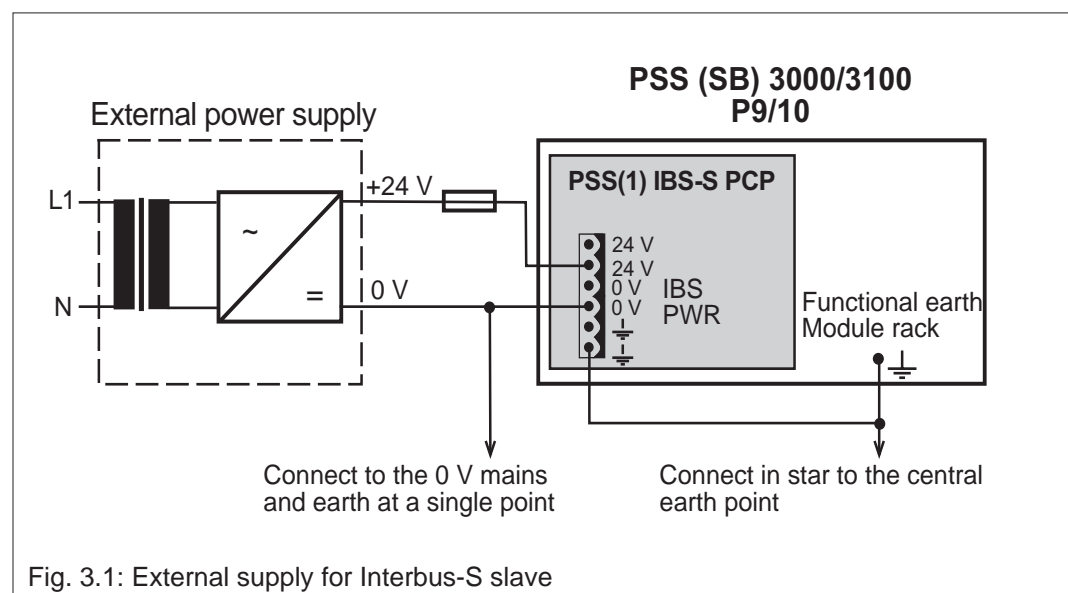
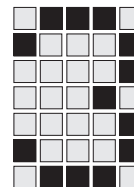


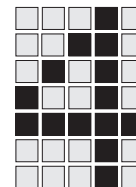
Fig. 3.1: External supply for Interbus-S slave



# Safety

---

**Notes**



# Interbus-S

## Overview

The PSS(1) IBS-S PCP module is designed as an Interbus-S slave. The interfaces IBS-IN (male 9-pin SUB-D connector) and IBS-OUT (female 9-pin SUB-D connector) are used to connect the Interbus-S remote bus.

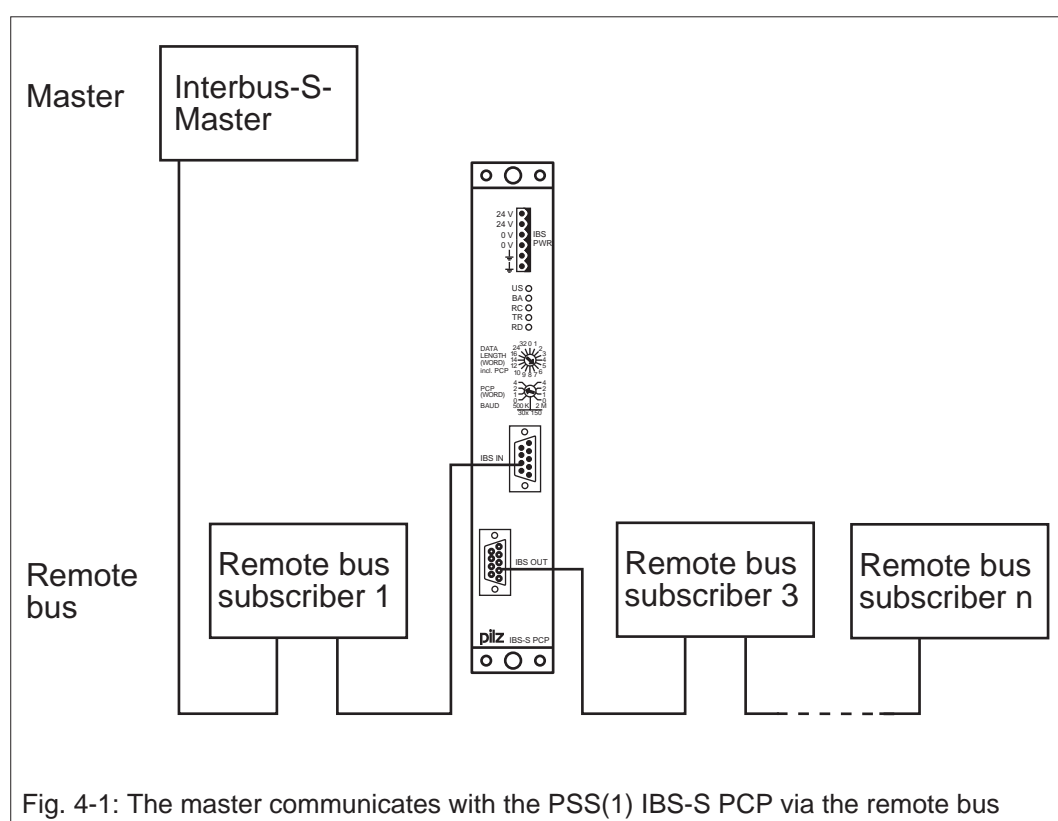


Fig. 4-1: The master communicates with the PSS(1) IBS-S PCP via the remote bus

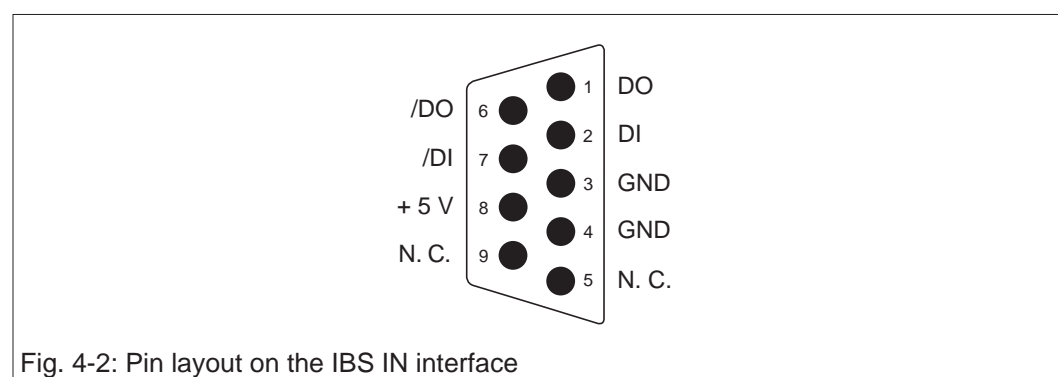


Fig. 4-2: Pin layout on the IBS IN interface

# Interbus-S

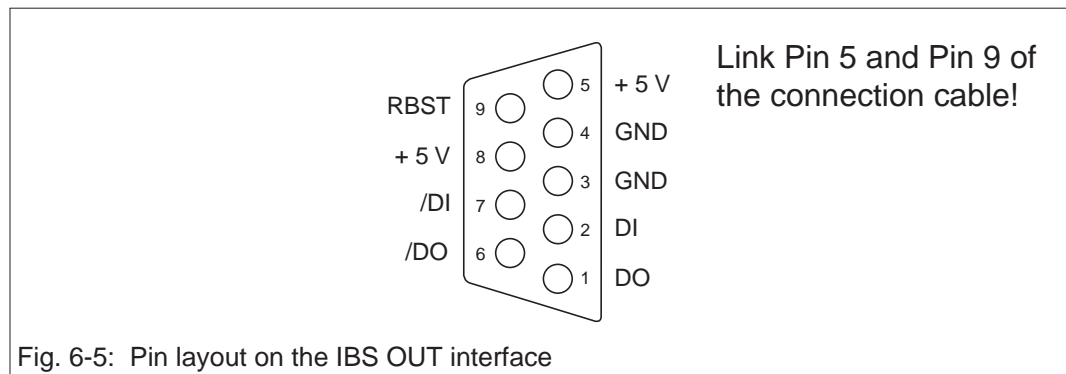


Fig. 6-5: Pin layout on the IBS OUT interface

The Interbus-S slave can be fed from the PSS power supply or from an external supply. An external power supply has the advantage that the Interbus-S slave can continue to operate even when no voltage is applied to the PSS.

If an external power supply is connected to the module it will automatically use this as its source. The Interbus connection will be interrupted as the module switches over.

As soon as the supply is returned to the Interbus-S slave it will automatically be reinitialised and the Interbus-S can be operated once more. The Interbus-S master will receive an error message.

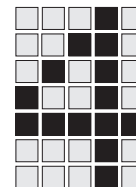
The Interbus-S slave has a process and a parameter data channel. The data length of the process and parameter data may be a combined total of 32 words and can be set via two rotary switches.

Depending on the parameter data length that has been set, the identification code of the Interbus-S slave will be as follows:

Parameter data length	Identification code
0	3 (03 <sub>hex</sub> )
1	243 (F3 <sub>hex</sub> )
2	240 (F0 <sub>hex</sub> )
4	241 (F1 <sub>hex</sub> )

The rotary switch for setting the parameter data length is also used to set the transmission rate. It is possible to select either 500 kBaud or 2 MBaud.

A Dual-Port RAM (DPR) forms the interface between the PSS and the Interbus-S slave. This is where all the information necessary for data



---

transfer is stored, together with data received and data to send. The PSS addresses the Dual-Port RAM as an addressable module via the standard bus. The DPR occupies 256 words.

The DPR is administered by a microprocessor. If the PSS stores send data and a send request in the Dual-Port RAM, these will be transferred to the Interbus-S remote bus subscriber via the microprocessor. The microprocessor will also store data received from the remote bus in the Dual-Port RAM.

The Interbus-S slave must be initialised before data is transferred. Initialisation will occur automatically once the supply voltage to the Interbus-S slave is switched on and the self check has proved to be error-free. During initialisation, the values set at the rotary switches for the process and parameter data length/transmission rate will be read in. If data lengths need amending after initialisation, this will need to be done via the user program – adjusting the rotary switch will have no effect.

The transfer of process data can be managed via a user-generated program or via standard function block SB 202 (IBS:SL\_2).

SB 202 (IBS:SLAV) can also be used, but in this case users will need to program their own check for data consistency (this is only important if the Interbus-S slave is to be fed from an external power supply “IBS PWR”).

The transfer of parameter data (operating system objects and user objects) is supported by the operating system (see also “Intended application” on page 3-1).

The parameter data channel on the Interbus-S slave has server functionality, i.e. the Interbus-S slave prepares parameter data only on request from the Interbus-S master. This means the following parameter data utilities are available: initiate, abort, reject, read, write, identify, get-OV and status.

The telegram length of the parameter channel is 240 bytes for send data and 240 bytes for receive data.

# Interbus-S

---

## Configuring the Dual-Port RAM

The Dual-Port RAM forms the interface between the PSS and Interbus-S slave. It is addressed by the PSS under addresses XW 0 to XW 255.

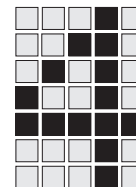


### **NOTICE**

**The word addresses stated in this chapter are only valid if the module's start address equals 0. If the module has a different start address, this figure will need to be added to the word addresses.**

The key to the words can be found in the tables overleaf.





XW	Key
0	<p>Bit 0: Write flag Set by the user program if process data has been stored in the RAM for sending. When the Interbus-S slave has accepted the data it will reset the bit.</p> <p>Bit 1: Error flag Set by the Interbus-S slave if an error has occurred during data transfer.</p> <p>Bit 2: Flag for amending the process data length Set by the user program if the process data length is to be amended. Once the change has been made, the Interbus-S slave will reset the bit.</p>
1	<p>Bit 0: Read flag Set by the Interbus-S slave if process data has been stored in the RAM for reading. When the user program has read the data it will reset the bit.</p> <p>Bit 1: Error flag Set by the Interbus-S slave if an error has occurred during data transfer.</p>
2 ... 33	Process data that is to be transferred from the PSS to the Interbus-S slave
34 ... 65	Process data that has been received from the Interbus-S slave and is to be transferred to the PSS
66	Process data length, entered when the Interbus-S slave is initialised or when the value is amended in the user program
67	Error code for the process data channel
68	If the process data channel is ready for operation, the Interbus-S slave will write 534B <sub>hex</sub> in XW 68
69	If the parameter data channel is ready for operation, the Interbus-S slave will write 4F4B <sub>hex</sub> in XW 69
70	Parameter data length, entered when the Interbus-S slave is initialised
71 ... 255	Reserved for the transfer of parameter data

# Interbus-S

## Setting the process and parameter data length


The data length of the process and parameter data may be a combined total of 32 words and can be set via two rotary switches on the front of the module.


The whole data length (process and parameter data length) is set using the upper switch. The lower switch is used to set the data length for the parameter data channel (PCP). The process data length is the difference between the whole data length and the parameter data length. For example, if the whole data length is set to 10 words and the parameter data length to 2 words, the process data length will be 8 words.

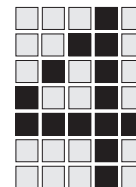
The switch for the whole data length has 16 settings, the switch for the parameter data length has 10 settings. Of the 10 settings on the lower switch, only 8 settings are actually valid, enabling 4 different parameter data lengths to be selected and the transmission rate to be set simultaneously.

The table below shows all the permitted settings:

		Switch setting for DATA LENGTH															
		0	1	2	3	4	5	6	7	8	9	10	12	14	16	24	32
Switch setting for parameter data length (PCP)	0	0	1	2	3	4	5	6	7	8	9	10	12	14	16	24	32
	1	0	1	2	3	4	5	6	7	8	9	11	13	15	23	31	
	2	0	1	2	3	4	5	6	7	8	10	12	14	22	30		
	4	0	1	2	3	4	5	6	8	10	12	20	28				
	Other																

 Invalid setting

 Process data length in words  
 2 Parameter data length in words



The settings at the rotary switches are read in as the Interbus-S slave is initialised. If a setting is invalid, the Interbus-S slave will not be able to operate on the remote bus.

The error code (22<sub>hex</sub>) for the error can be found in XW 67. The fault must be rectified and initialisation restarted by switching the Interbus-S slave supply (or PSS supply) off and then on again.

### **Changing the set data lengths:**

If you wish to amend the values set for the process and parameter data length, you will need to correct the switch settings and re-initialise by switching the Interbus-S slave supply (or PSS supply) off and then on again. This will interrupt the Interbus transmission. When you amend the parameter data length, the identification code of the Interbus-S slave will also change and will need to be read in again from the Interbus-S.

If you only wish to change the process data length, this can be done through the user program rather than having to re-initialise. Write the required process data length in XW 66 and set bit 2 from XW 0 to “1”. The Interbus-S slave will then check the value you have entered.

- If the value is incorrect, the error will be registered in XW 67. Rectify the error and set bit 2 from XW 0 to “1” again.
- If the value entered in XW 66 is identical to the old value for the process data length, nothing will happen.
- If the value is correct, the Interbus-S slave will interrupt the Interbus-S transmission and reset the process data length. The Interbus-S slave will then reset bit 2 from XW 0 to “0”.



### **INFORMATION**

The value for the current process data length is stored in XW 66 and the value for the current parameter data length is stored in XW 70.

If the process data length is amended in the user program, the value in XW 66 will not match the process data length set on the rotary switch.

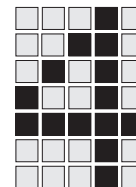
# Interbus-S

---

## Initialising the Interbus-S slave

The Interbus-S slave has to be initialised before data transfer can take place. This occurs automatically on power up or when voltage is returned to the Interbus-S supply.

Sequence	Word configuration	
A self-check is performed when the Interbus-S supply is switched on or power is returned. If no error is found, the RAM will be initialised with "0".	XW0 . . . XW255	<div style="border: 1px solid black; padding: 2px; display: inline-block; text-align: center;">0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; text-align: center;">0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</div>
In the case of an error:	XW67	<div style="border: 1px solid black; padding: 2px; display: inline-block; text-align: center;">Error code</div>
The process data length set on the rotary switch is stored in:	XW66	<div style="border: 1px solid black; padding: 2px; display: inline-block; text-align: center;">Process data length</div>
The parameter data length set on the rotary switch is stored in:	XW70	<div style="border: 1px solid black; padding: 2px; display: inline-block; text-align: center;">Parameter data length</div>
If the module is ready for operation it registers (534B hex.):	XW68	<div style="border: 1px solid black; padding: 2px; display: inline-block; text-align: center;">0 1 0 1 0 0 1 1 0 1 0 0 1 0 1 1</div>



## Transferring process data without SB 202

### Data consistency

If the supply to the Interbus-S slave “IBS PWR” fails, it is possible that the PSS could read corrupted process data from the RAM because, if the supply fails, the bits in the RAM do not suddenly change their status.



#### NOTICE

In order to ensure data consistency, the user program must check whether the Interbus-S slave supply voltage is present or whether the Interbus-S is ready for operation (XW 68 = 534B<sub>hex</sub>), before and after data is read from the RAM. If the Interbus-S slave is fed from the PSS supply, data consistency is automatically guaranteed.

Data consistency in the process data channel on the Interbus-S master will depend on the manufacturer and/or specification of the master module.

### Transferring process data from the PSS to Interbus-S

Sequence	Word configuration																	
Check XW68 for 534B <sub>hex</sub> (ready for operation)	XW68 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td></tr></table>	0	1	0	1	0	0	1	1	0	1	0	0	1	0	1	1	
0	1	0	1	0	0	1	1	0	1	0	0	1	0	1	1			
Check XW67 for 0 (free from error)	XW67 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
Write send data to the RAM:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td style="width: 100px;">XW2</td><td style="text-align: center;">Data 1</td></tr> <tr><td style="width: 100px;">XW3</td><td style="text-align: center;">Data 2</td></tr> <tr><td></td><td style="text-align: center;">⋮</td></tr> <tr><td style="width: 100px;">XW33</td><td style="text-align: center;">Data 32</td></tr> </table>	XW2	Data 1	XW3	Data 2		⋮	XW33	Data 32									
XW2	Data 1																	
XW3	Data 2																	
	⋮																	
XW33	Data 32																	
If the data is ready for transfer:	XW0 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>1</td></tr></table>	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1		
After the data is transferred the Interbus-S slave will set:	XW0 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>0</td></tr></table>	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0		

x: unchanged

# Interbus-S

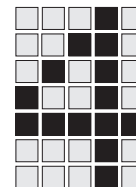
If an error occurs during the transfer of data from the PSS to Interbus-S, the Interbus-S slave will set bit 1 from XW 0. The error code can be read from XW 67.

## Transferring process data from Interbus-S to the PSS

Sequence	Word configuration																																																
Check XW68 for 534B <sub>hex</sub> (ready for operation)	XW68 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td></tr></table>	0	1	0	1	0	0	1	1	0	1	0	0	1	0	1	1																																
0	1	0	1	0	0	1	1	0	1	0	0	1	0	1	1																																		
Check XW67 for 0 (free from error)	XW67 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																																
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																																		
Check whether the Interbus component has written data to the RAM, i.e. whether bit 0 from XW 1 is set:	XW1 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>1</td></tr></table>	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1																																
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1																																		
Read data from the RAM:	XW34 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td colspan="16" style="text-align: center;">Data 1</td></tr></table> XW35 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td colspan="16" style="text-align: center;">Data 2</td></tr></table> ⋮ XW65 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td colspan="16" style="text-align: center;">Data 32</td></tr></table>	Data 1																Data 2																Data 32															
Data 1																																																	
Data 2																																																	
Data 32																																																	
After the data is transferred, the PSS will set (ready for new data):	XW1 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>0</td></tr></table>	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0																																
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0																																		
Check XW68 for 534B <sub>hex</sub> (data consistency). If there is any deviation, the data read in is invalid.	XW68 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td></tr></table>	0	1	0	1	0	0	1	1	0	1	0	0	1	0	1	1																																
0	1	0	1	0	0	1	1	0	1	0	0	1	0	1	1																																		

x: unchanged

If an error occurs during the transfer of data from Interbus-S to the PSS, the Interbus-S slave will set bit 1 from XW 1. The error code can be read from XW 67.



---

## Transferring process data with SB 202 (IBS:SL\_2)

Standard function block SB 202 (IBS:SL\_2) is available for communication between the PSS CPU and the Interbus-S slave. This block can be used on the PSS SB 3006 IBS-S for both modules PSS(1) IBS-S PCP and PSS(1) IBS-S.

Standard function block SB 202 (IBS:SLAV) for the P9/P10 has the same number and the same input/output parameters, but has reduced functionality (it does not test the data consistency, etc.).

### Function

- Transfers process data between the Dual-Port RAM of the Interbus-S slave and the PSS CPU
- Establishes the data blocks containing receive and send data
- Checks the process data width
- States the start address
  - PSS SB 3006 IBS-S:* the PSS Interbus-S slave must have the start address "0"; it requires 256 words (XW 0 ... XW 255)
  - PSS(1) IBS-S PCP:* the Interbus-S slave for these modules requires a total of 256 words of free address space. This address space must not be assigned twice!
  - PSS(1) IBS-S:* the Interbus-S slave for these modules requires a total of 128 words of free address space. This address space must not be assigned twice!
- Checks the data consistency



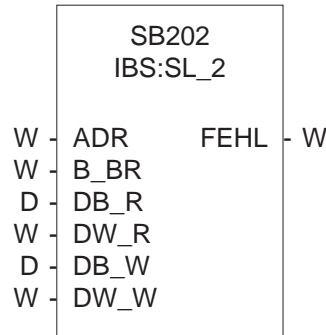
### INFORMATION

This block does not support the transfer of parameter data (see also page 4-17).

# Interbus-S

---

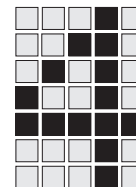
## Block header



## Input parameters

- **ADR:** Offset for the module's free address, permitted range: 0 ... 32.640 ("0" must be entered on the PSS SB 3006 IBS-S ), Input base: 256 on the *PSS SB 3006 IBS-S/ PSS(1) IBS-S PCP* and 128 on the *PSS(1) IBS-S*
- **B\_BR:** Process data length of the Interbus-S, permitted range: 1... 32 (see page 4-6 for details of permitted settings). The entry must match the setting on the rotary switch and the value stated in XW 66.
- **DB\_R:** Data block for storing data to be sent from the Interbus-S slave to the Interbus-S master; permitted setting: DB 10 ... 255
- **DW\_R:** Data word from which data to be sent to the Interbus-S master is to be stored in *DB\_R*; permitted settings: 1 ... 1023  
The number of data words required will correspond to the process data length. The declared length of the block (read/write) must therefore be at least *DW\_R* plus the process data length. DW 0 from *DB\_R* is reserved and not available for use.
- **DB\_W:** Data block for storing data that the Interbus-S slave is to receive from the Interbus-S master; permitted setting: DB 10 ... 255
- **DW\_W:** Data word from which data received from the Interbus-S master is to be stored in *DB\_W*; permitted settings: 1 ... 1023  
The number of data words required will correspond to the process data length. The declared length of the block (read/write) must therefore be at least *DW\_W* plus the process data length.





## Output parameters

- *Fehl*: Error messages are displayed in the high byte and low byte of output parameter *Fehl*. Please refer to the table below.

<i>Fehl</i> Low byte	Key
<div style="border: 1px solid black; padding: 2px; display: inline-block;">00000000</div>	No error
Bit <div style="display: inline-block; vertical-align: middle; margin-left: 10px;">             3 2 1  <div style="border: 1px solid black; padding: 2px; display: inline-block;">00001110</div> </div>	Interbus-S is inactive, no data is being transferred. Remedy: The Interbus-S slave reactivates automatically.
Bit <div style="display: inline-block; vertical-align: middle; margin-left: 10px;">             3 2 1 0  <div style="border: 1px solid black; padding: 2px; display: inline-block;">00001111</div> </div>	Interbus-S cable connection is defective or the Interbus-S master is in reset. Remedy: The Interbus-S slave reactivates automatically.
Bit <div style="display: inline-block; vertical-align: middle; margin-left: 10px;">             4 3  <div style="border: 1px solid black; padding: 2px; display: inline-block;">00011000</div> </div> Bit <div style="display: inline-block; vertical-align: middle; margin-left: 10px;">             4 3 0  <div style="border: 1px solid black; padding: 2px; display: inline-block;">00011001</div> </div> Bit <div style="display: inline-block; vertical-align: middle; margin-left: 10px;">             4 3 1  <div style="border: 1px solid black; padding: 2px; display: inline-block;">00011010</div> </div> Bit <div style="display: inline-block; vertical-align: middle; margin-left: 10px;">             4 3 1 0  <div style="border: 1px solid black; padding: 2px; display: inline-block;">00011011</div> </div>	Initialisation error, RAM may be defective. Remedy: Reset the Interbus-S slave. If the RAM is defective, please contact Pilz.
Bit <div style="display: inline-block; vertical-align: middle; margin-left: 10px;">             5  <div style="border: 1px solid black; padding: 2px; display: inline-block;">00100000</div> </div>	Too high a data length entered in XW 66 when the process data length was amended via the user program. Remedy: Correct the setting on the rotary switch and reset the Interbus-S slave or amend the process data length in the user program.
Bit <div style="display: inline-block; vertical-align: middle; margin-left: 10px;">             5 1  <div style="border: 1px solid black; padding: 2px; display: inline-block;">00100001</div> </div>	Invalid data length entered in XW 66 when the process data length was amended via the user program. Remedy: Correct the setting on the rotary switch and reset the Interbus-S slave or amend the process data length in the user program.
Bit <div style="display: inline-block; vertical-align: middle; margin-left: 10px;">             5 1  <div style="border: 1px solid black; padding: 2px; display: inline-block;">00100010</div> </div>	XW 66 contains an invalid process data length. Remedy: Correct the setting on the rotary switch and reset the Interbus-S slave or amend the process data length in the user program.

# Interbus-S

<i>Fehl</i> High byte	Key
00000000	No error
Bit 1 00000010	Invalid value for <i>DW_R</i> Remedy: Correct and reset the Interbus-S slave
Bit 2 00000100	Invalid value for <i>DW_W</i> Remedy: Correct and reset the Interbus-S slave
Bit 3 00001000	Invalid value for start address Remedy: Correct and reset the Interbus-S slave
Bit 4 00010000	Interbus wiring: fatal error Remedy: Reset the Interbus-S slave
Bit 5 00100000	Waiting for initialisation to finish
Bit 6 01000000	Data width in XW 66 is different from <i>B_BR</i> Remedy: Correct the setting on the rotary switch and reset the Interbus-S slave, amend the process data length in the user program or amend <i>B_BR</i> .
Bit 7 10000000	Interbus-S is not operating.

## Blocks required

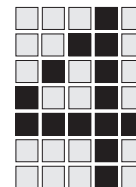
None

## Temporary flags

MW 61.00  
MW 61.16  
MW 62.00  
MW 62.16  
MW 63.00  
MW 63.16

## Reserved flags

None



## Operating system flags

MW 114.00  
MW 114.16

## Error messages

See output parameter FEHL

## Programming guidelines

- We recommend that the data words of the data blocks named in *DB\_R* and *DB\_W* are pre-assigned the value 0.
- One data block can be used simultaneously as *DB\_R* and *DB\_W*.



### NOTICE

SB 202 (IBS:SL\_2) must be called up as part of each user program cycle.



### INFORMATION

Data word DW 0 from data block *DB\_R* is reserved and must not be used. However, DW 0 must be initialised:

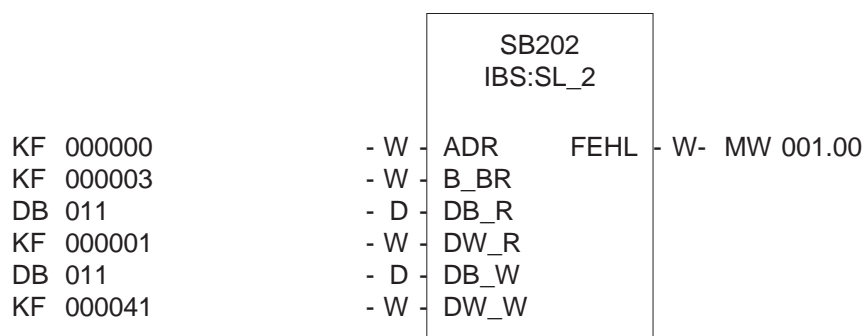
- P9/P10: In OB 020 or OB 021, initialise DW 0 of *DB\_R* with "0".
- PSS 3000/3100 series: In OB 020, initialise DW 0 of *DB\_R* with "0".

## Program example

The program extract given below explains how to set parameters for SB 202 (IBS:SL\_2).

The process data length is 3 words. The data to be sent to the Interbus-S master is to be stored in DB 011, from DW 001 onwards, and the data to be received from the Interbus-S master is to be stored in DB 011, from DW 041 onwards.

```
:CAL SB 202
```



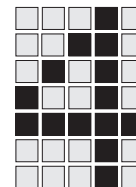
# Interbus-S

---

## Error messages in XW 67

Any errors that occur during the transfer of process data will be entered in XW 67 of the Dual-Port RAM.

Error code	Key
0	No error
0F <sub>hex</sub>	Interbus-S is inactive, no data is being transferred. Remedy: The Interbus-S slave reactivates automatically.
0E <sub>hex</sub>	Interbus-S cable connection is defective or the Interbus-S master is in reset. Remedy: The Interbus-S slave reactivates automatically.
18 ... 1B <sub>hex</sub>	Initialisation error, RAM may be defective. Remedy: Reset the Interbus-S slave. If the RAM is defective, please contact Pilz.
20 <sub>hex</sub>	Too high a data length entered in XW 66 when the process data length was amended via the user program. Remedy: Correct the setting on the rotary switch and reset the Interbus-S slave or amend the process data length in the user program.
21 <sub>hex</sub>	Invalid data length entered in XW 66 when the process data length was amended via the user program. Remedy: Correct the setting on the rotary switch and reset the Interbus-S slave or amend the process data length in the user program.
22 <sub>hex</sub>	XW 66 contains an invalid process data length. Remedy: Correct the setting on the rotary switch and reset the Interbus-S slave or amend the process data length in the user program.



## Transferring parameter data

As the module acts as a server, in the parameter data channel it must only react to requests from other Interbus-S subscribers.

Two types of communication objects are available, both of which can be polled via the Interbus-S parameter data channel:

- Operating system objects and
- User objects.

The following section describes these objects and explains how they are transferred. The telegram length of the parameter data channel is 240 bytes for send data and 240 bytes for receive data.



### INFORMATION

The transfer of parameter data is not supported by all CPUs. Please refer to the section entitled “Intended application” on page 3-1.



### NOTICE

When transferring user or operating system objects that are “octet strings”, the high byte and low byte of the data words are transposed. This must be taken into account when interpreting the data!

**Example:** An octet string user object is 8 bytes in length and occupies words DW 150 to DW 153 within a data block. It is made available on the Interbus-S after a read request.

User object in the PSS

DW150	Byte 1 (High byte)	Byte 0 (Low byte)
DW151	Byte 3	Byte 2
DW152	Byte 5	Byte 4
DW153	Byte 7	Byte 6

# Interbus-S

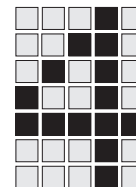
---

User object in Interbus-S (Read confirmation)

Command code	
Number of words following	
Instruction number	Communications reference
"0", if "Positive Confirmation"	
Reserved	Amount of parameter data following, in bytes
Byte 0	Byte 1
Byte 2	Byte 3
Byte 4	Byte 5
Byte 6	Byte 7

## Data consistency

Data consistency in the parameter data channel is ensured if the Interbus-S master does not send new requests to the Interbus-S slave until the previous request has been answered.



## Operating system objects

Operating system objects contain information about the error stack and PSS FS/ST status flags, as well as information on the error stacks of the devices attached to SafetyBUS p.

Index (hex)	Data type	Length (Byte)	Access rights	Name
5000	Octet string	130	Ra	PSS error stack
5001	Unsigned8	1	Ra	FS status flag
5002	Unsigned8	1	Ra	ST status flag
5003	Unsigned8	1	Wg	SafetyBUS p device address
5004	Octet string	16 x 128	Ra	SafetyBUS p error stack
5005	Octet string	56	Ra	SafetyBUS p diagnostic data

Ra: Interbus-S master has read access

Wg: Interbus-S master has write access

## Description of operating system objects

### PSS error stack

The error stack contains the PSS error messages. An error stack entry is composed of 4 words:

- 1st word: Error class of error
- 2nd word: Error number
- 3rd word: Error location
- 4th word: Error parameter

If the operating system object "PSS error stack" is read, 130 bytes will be transmitted. The first 128 bytes contain 16 error messages. The most recent error will be stored in the first 4 words, the next most recent error will be stored in the next 4 words, etc.

The 129th byte contains the version status of the FS operating system and the 130th byte the version status of the ST operating system.

# Interbus-S

---

## FS status flags

FS status flags provide information on the current status of the PSS FS section.

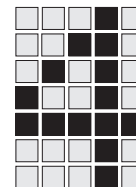
- Bit 0 (M 113.00)  
If the FS section is in a “RUN” condition, this bit will be “1”.
- Bit 1 (M 113.01)  
If an error has occurred in the FS section, this bit will be “1”.
- Bit 2 (M 113.02)  
If the FS section has switched to a “STOP” condition as the result of a Stop command, this bit will be “1”.
- Bit 3 (M 113.03)  
When the FS section has transferred from a “STOP” to a “RUN” condition, this bit will be “1” for one cycle.
- Bit 4 (M 113.04)  
When the FS section has transferred from an “OFF” to a “RUN” condition, this bit will be “1” for one cycle.

## ST status flags

ST status flags provide information on the current status of the PSS ST section.

- Bit 0 (M 112.00)  
If the ST section is in a “RUN” condition, this bit will be “1”.
- Bit 1 (M 112.01)  
If an error has occurred in the ST section, this bit will be “1”.
- Bit 2 (M 112.02)  
If the ST section has switched to a “STOP” condition as the result of a Stop command, this bit will be “1”.
- Bit 3 (M 112.03)  
When the ST section has transferred from a “STOP” to a “RUN” condition, this bit will be “1” for one cycle.
- Bit 4 (M 112.04)  
When the ST section has transferred from an “OFF” to a “RUN” condition, this bit will be “1” for one cycle.
- Bit 5 (M 112.05)  
After a general reset, this bit will be “1” for one cycle.





### NOTICE

The following CPUs will read incorrect values for the ST status flags:

- PSS CPU, Version 2.1
- PSS1 CPU, Version 1.5
- PSS SB CPU, Version 1.0
- PSS1 SB CPU, Version 1.0



### Device address and SafetyBUS p error stack

#### INFORMATION

These operating system objects will only be available on CPUs with SafetyBUS p capabilities (they are not yet available on Version 1.0/1.1 of the PSS(1) SB CPU).

The device address contains the address of a device on SafetyBUS p, following on from which the error stack is transmitted. This error stack may be a maximum of 2 KBytes in size. The error stack is transferred as an array, with a maximum of 16 units of 128 bytes each. Only the section of the error stack that contains error messages will be transferred.

Structure of the transferred data:

Total length of SafetyBUS p error stack including device ID in words (max. 1024)	
	Length of device ID in words (max. 16 words)
Device ID	
Reserved	Error class
Reserved	Error number
Reserved	Amount of error parameter data following, in bytes (max. 4 bytes)
Error parameter 1	Error parameter 2
Error parameter 3	Error parameter 4
... additional error entries (in reverse order to which they occurred)	

} 1st error entry (most recent error)

# Interbus-S

---

Sequence when calling up the SafetyBUS p error stack

First the PSS Interbus-S master has to advise on which device on the SafetyBUS p the error stack is to be read from. This is done via a write request to operating system object “5003<sub>hex</sub>”, stating the device address.

The first read request to operating system object “5004<sub>hex</sub>” with sub-index “1” must be made immediately after the PSS confirms that the device address has been transferred successfully. The first section of the error stack will then be transferred.

The first word contains the total length of the error stack. From this it is possible to calculate how many requests will be necessary to read out the complete error stack.

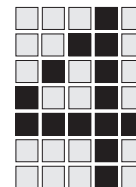
Requests to elements (sub-indices) must be made in ascending order. The time between requests must be less than a minute, otherwise the transfer of the error stack will be aborted (timeout).

Once the transfer of the error stack is complete or the transfer aborted, the device address will be deleted. This means that the device address must always be transferred before the first request to read the error stack.

The transfer of the error stack can be aborted for three reasons:

- If a timeout occurs
- If the user attempts to read one more sub-index than is available
- If the SafetyBUS p is not ready for operation.

Each aborted attempt will produce an error message.



## SafetyBUS p diagnostic data

### INFORMATION

These operating system objects are only available on CPUs with SafetyBUS p capabilities (on the PSS(1) SB CPU, from Version 1.2).

This operating system object contains diagnostic data for the whole SafetyBUS p bus system and also specific CAN diagnostic data. 56 bytes are transmitted when operating system object 5005 is read. Contents of operating system object:

- Byte 0 ... 7: Bus subscriber registering the error
- Byte 8 ... 15: Bus subscriber whose I/O-Groups are not in a RUN condition
- Byte 16 ... 23: Bus subscribers actively involved in bus traffic
- Byte 24 ... 31: Bus subscribers that registered after the transmission rate was switched
- Byte 32 ... 39: Bus subscribers that registered before the transmission rate was switched
- Byte 40 ... 55: CAN diagnostic data

SafetyBUS p diagnostic data is bit-coded. One bit is allocated to each bus subscriber for each diagnostic topic. If the bit equals "1", the above statement will apply to the bus subscriber.

Example: "Bus subscriber registering the error":

Byte	7	6	5	4	3	2	1	0
Bit	7 ... 0	7 ... 0	7 ... 0	7 ... 0	7 ... 0	7 ... 0	7 ... 0	7 ... 0
Device address	95...88	87...80	79...72	71...64	63...56	55...48	47...40	39...32

# Interbus-S

---

Structure of CAN diagnostic data:

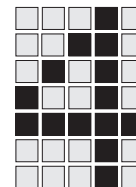
Word	Byte	Contents
0	40 (HB)	Counter for reaching warning limits
	41 (LB)	
1	42 (HB)	Counter for data overrun
	43 (LB)	
2	44 (HB)	Counter for passive error
	45 (LB)	
3	46 (HB)	Counter for reaching warning limits (10 cycles)
	47 (LB)	
4	48 (HB)	Counter for data overrun (10 cycles)
	49 (LB)	
5	50 (HB)	Counter for passive error (10 cycles)
	51 (LB)	
6	52	Maximum bus load in %
	53	Current bus load in %
7	54	Maximum bus load (10 cycles)
	55	Average bus load (10 cycles)

## Transferring operating system objects

Access by the Interbus-S master to the operating system objects does not require support from the user program. The whole process can be performed between the Interbus-S slave and the operating system. This is also the case if the ST section is in a “STOP” condition.

As part of each cycle, the operating system will check whether there is a request for an operating system object and service it accordingly. If the system fails to react to a request from the Interbus-S master (e.g. PSS is switched off), a timeout will be registered on the Interbus-S master after a period of 20 s (see “Error messages on Interbus-S”).

For the transfer of operating system objects to work, a value greater than “0” must be entered in the FS configurator for the ST section minimum block run time.



## User objects

The user program employs user objects in order to provide data to the Interbus-S master via the parameter channel. User objects must be defined and initialised by the user. The Interbus-S master can both read and write to these objects.

Index (hex)	Data type	Length (Bytes)	Access rights	Name
5100	Octet string	max. 130	RaWg	User data 1
5101	Octet string	max. 130	RaWg	User data 2
5102	Octet string	max. 130	RaWg	User data 3
5103	Octet string	max. 130	RaWg	User data 4
5104	Octet string	max. 130	RaWg	User data 5

Ra: Interbus-S master has read access

Wg: Interbus-S master has write access

## Transferring user objects

The transfer of user objects is supported by SB 254 (StBsCall) using functions 60, 61 and 62. It undertakes all the handshake mechanisms via the DPR. A general description of SB 254 can be found in the “ST System Description” in the System Manual. Functions 60, 61 and 62 are described below.

Function code	Function
60	Initialise user objects
61	Check request from Interbus-S
62	Declare object data valid

# Interbus-S

---

User objects must be initialised in the user program before they are transferred. This can be done using function 60 on SB 254.

The user program uses function 61 on SB 254 to check whether there is a request for a user object.

In the case of a write request (Interbus-S master wants to amend a user object), the PSS operating system copies data from the Dual-Port RAM into the user object on the PSS system.

In the case of a read request (Interbus-S master wants to read a user object), the user program will first have to declare the data valid by calling up function 62 from SB 254. The data will then be copied to the RAM and will be available to the Interbus-S master.

Each request is retained until it is processed. However, if it has still not been processed after 20 s, it will be rejected and the Interbus-S will be informed.

## Initialising user objects

Each user object corresponds to a data range of 65 words in a data block on the PSS system.

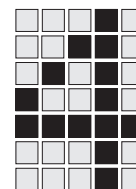
Each user object must be initialised so that the Interbus-S knows in which data block and from which data word a user object is stored.

This is done by calling up function 60 of SB 254.

The initialisation parameters in DB 004 (from DW 200 onwards) must be transferred in the process (see table).

If the initialisation of the user object has been successful, the value "1" will be stored in output parameter *ERG* of SB 254 and the value "0" will be stored in DW 200 of DB 000.

If an error has occurred, output parameter *ERG* of SB 254 will contain the value "16" and DW 200 of DB 000 will contain the error ID (see table).



Block	Input	Output	Key
SB 254	Funk = 60		Define data blocks in which the user objects for the parameter data channel are stored
		ERG = 1	Function successful
		ERG = 16	Error
DB 004	DW 200		Number of data block containing the object
	DW 201		Data word from which the object starts in the DB
	DW 202		Object index
DB 000		DW 200	Error ID 0: No error 1: Invalid DB number 2: Invalid start address 4: Object data ranges overlap 8: No PCP module connected 16: DB not in R/W mode 32: Object index not permitted 64: Object already exists

Error description:

0: No error

1: Invalid DB number

The stated data block has not been defined in the user program.

2: Invalid start address

Too high a value has been stated for the object's start address in the data block. The user object's 65 words no longer fit in the data block.

4: Object data ranges overlap

In the data block, a user object starts before another user object ends.

8: No PCP module connected

# Interbus-S

---

**16: DB not in R/W mode**

The stated data block was not assigned read/write mode when it was created.

**32: Object index not permitted**

The object index stated for the user object in DB 004 is not within the permitted range.

**64: Object already exists**

A user object with the stated object index has already been defined.

## **Checking whether there is a request from the Interbus-S master**

Once function 61 of SB 254 has been called up, the user program can check whether a request has been made upon a user object. To do this, DW 201 and DW 202 from DB 000 need to be evaluated.

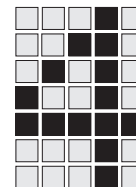
If DW 202 from DB 000 does not equal "0", a write request is present (Interbus-S wants to amend a user object): the PSS operating system will copy data from the Dual-Port RAM into the PSS system's user object. The object index of the amended object will be stored in DW 202 from DB 000.

If DW 201 from DB 000 does not equal "0", a read request is present (Interbus-S master want to read a user object): the object index of the object the Interbus-S master wishes to read is located in DW 201 from DB 000. The user program must read in the object index and make the corresponding data available. The user program confirms that the data is ready by calling up function 62 of SB 254. The data will then be copied to the RAM and will be available to the Interbus-S master.

Each request is retained until it is processed. However, if it has still not been processed after 20 s, it will be rejected and the Interbus-S will be informed.

If there is an error on the Interbus-S slave, no request for the PSS system will be present.





Block	Input	Output	Key
SB 254	Funk = 61		Enquire whether there is a request to read or write to a user object
		ERG = 1	Function successful
		ERG = 16	Error
DB 000		DW 200	Error ID 0: No error 1: No user objects have been initialised in the user program
		DW 201	Object index in the case of a read request, otherwise "0"
		DW 202	Object index in the case of a write request, otherwise "0"

### Declare object data valid

When the Interbus-S master sends a read request to the PSS system, the PSS operating system will not write the user object data into the Dual-Port RAM of the Interbus-S slave until function 62 of SB 254 has been called up.

When function 62 of SB 254 is called up, the object index of the user object must be stated in DW 200 of DB 004 and the length of the object must be stated in DW 201.

The object length can be determined via the user program. Between 1 and 65 words are possible. If the value "0" is stated as the object length, 65 words will be copied.

# Interbus-S

---

Block	Input	Output	Key
SB 254	Funk = 62		User program confirms that data from the requested user object is available in the relevant DB
		ERG = 1	Function successful
		ERG = 16	Error
DB 004	DW 200		Object index
	DW 201		Data length of user object
DB 000		DW 200	Error ID 0: No error 1: Object index is invalid 2: Object index stated does not correspond to the index of requested object 4: No request was present 8: Interbus-S slave is not operational 16: Parameter data cannot be transferred

**Error description:**

0: No error

1: Object index is invalid

The object index stated for the user object in DB 004 is not within the permitted range.

2: Object index stated does not correspond to the index of requested object

The object index transferred was not the object index of the requested user object.

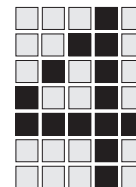
4: No request was present

8: Interbus-S slave is not operational

Supply voltage to the Interbus-S slave is switched off.

16: Parameter data cannot be transferred

The word length set for the parameter data channel on the rotary switch is invalid.



## Program example

Two user objects are initialised in the following program example. The program then checks whether a read or write request from the Interbus-S is present and, if necessary, services this request.

Read data is to be made ready for the Interbus-S master in the user object with object index 5100<sub>hex</sub>. Data from the Interbus-S master is to be stored in the user object with object index 5101<sub>hex</sub>. Flags M 000.00 and M 000.01 are used to ensure the user objects are initialised once only. These flags are set to “1” right at the start, e.g. in OB 020. The status of flag M 000.00 is polled before user object 5100<sub>hex</sub> is initialised. If the status is “1”, initialisation will take place and the flag will be reset to “0”. Flag M 000.01 is used in the same way with user object 5101<sub>hex</sub>.

```

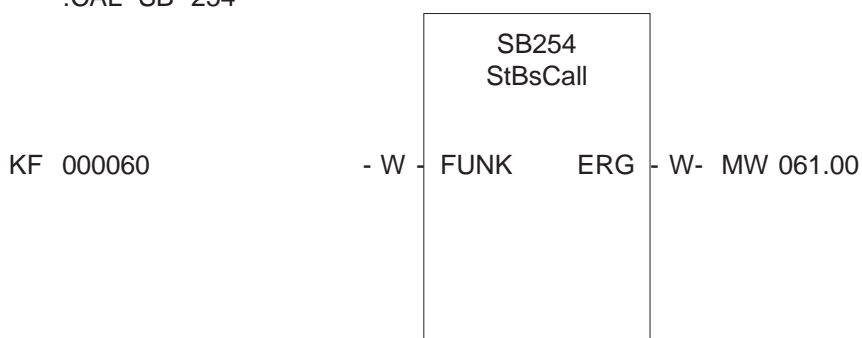
: ...
: ...
: ...
PCP_5100   : Segment 02 // Initialisation of user object 5100hex
                                     // (DB 021 from DW 000 onwards)

:LN  M  000.00 // Poll flag M 000.00
                                     // Initialisation only occurs if M 000.00 = 1

:SPB = PCP_5101
:R   M  000.00 // Reset flag, so that initialisation
                                     // is not repeated.

:A   DB  004 // Select DB 004
:L   KF  000021 // Number of data block containing the object
:T   DW  0200
:L   KF  000000 // Data word from which object starts in DB
:T   DW  0201
:L   KH  5100 // Object index
:T   DW  0202
:CAL SB  254

```



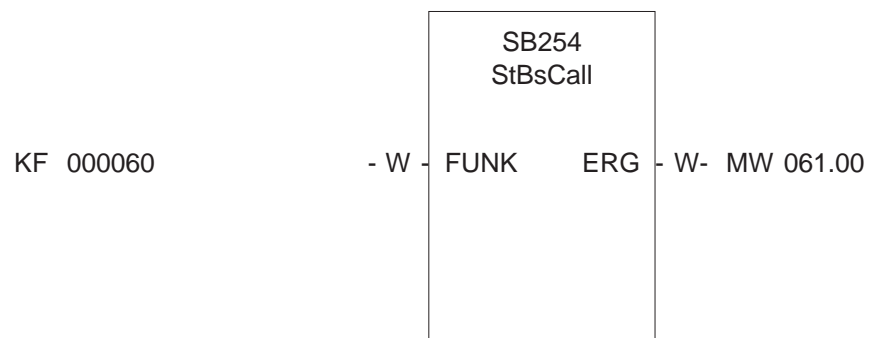
# Interbus-S

---

```

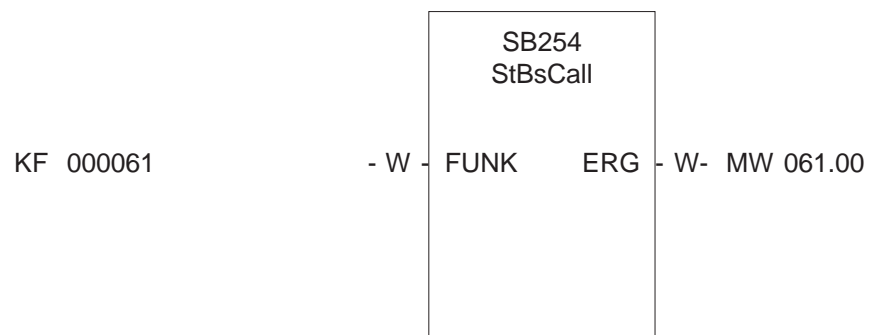
PCP_5101   : Segment 03 // Initialisation of user object 5101hex
                                     // (DB 021 from DW 100 onwards)

:LN  M  000.01 // Poll flag M 000.01
                                     // Initialisation only occurs if M 000.01 = 1
:SPB =  PCP_Read
:R   M  000.01 // Reset flag, so that initialisation
                                     // is not repeated.
:A   DB  004   // Select DB 004
:L   KF  000021 // Number of data block containing the object
:T   DW  0200
:L   KF  000100 // Data word from which object starts in DB
:T   DW  0201
:L   KH  5101   // Object index
:T   DW  0202
:CAL SB  254
  
```



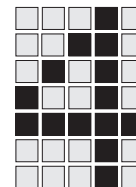
```

PCP_Read: Segment 04 // Check whether there is a read or write request from
                                     // the Interbus-S master.
:CAL SB  254
  
```



```

:L   MW  061.00 // Poll result of function
:!=  KF  000001 // if MW 061.00 = 1, function was successful
:SPB =  PCP-ReadReq
:SPA =  End_Example
  
```



**PCP\_ReadReq: Segment 05**

// Once it is established that function 61 was  
 // successfully carried out, a check will be made to  
 // see whether there is a read request upon user  
 // object 5100<sub>hex</sub>.

```
:A DB 000 // Select DB 000
:L DW 201 // If a read request is present, DW 201 will contain the
           // index of the object that is to be read.

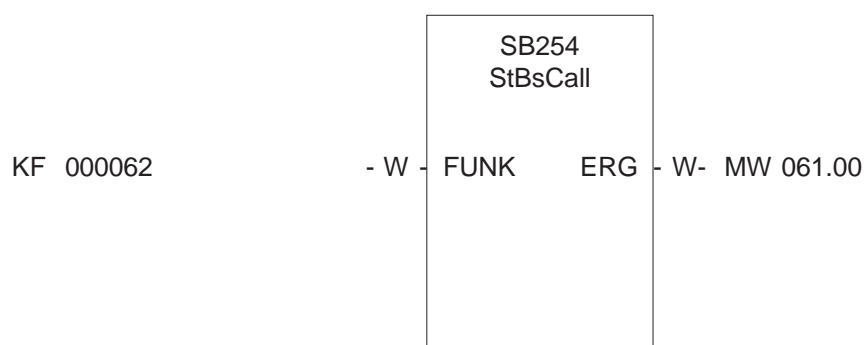
:= KH 5100
:SPB = PCP-Read5100 // If DW 201 = 5100hex, there is a read request upon this
                   // object.
:SPA = PCP-WriteReq // If DW 201 ≠ 5100hex, there may be a read request for a
                   // non-existent object or a write request.
```

**PCP\_Read5100: Segment 06**

// Processing the read request

```
: ... // The user object data that is to be declared
: ... // valid using function 62 of DB 254 should be entered
: ... // here (in DB 021 from DW 000 onwards). The data
       // length of the user object should be 10 data words.

:A DB 004 // Select DB 004
:L KH 5100 // Object index
:T DW 0200
:L KF 000010 // Data length of user object
:T DW 0201
:CAL SB 254
```



```
:SPA = End_Beispiel // Read request processed in full
```

# Interbus-S

---

```
PCP_WriteReq: Segment 07 // Check whether there is a write request upon user //
// object 5101hex.

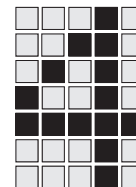
:L   DW 202                // If a write request is present, DW 202 will contain the
// index of the object into which the Interbus-S master
// wishes to write data.

:=   KH 5101
:SPB = PCP-Write5101 // If DW 202 = 5101hex, there is a request to write to
// this object.
:SPA = End_Example // If DW202 ≠ 5101hex, no request is present.
```

## **PCP\_Write5101: Segment 08**

```
: ... // Data from the user object (DB 021 from DW 100
: ... // onwards) can be read and evaluated at this point.
```

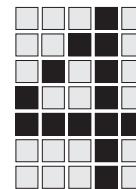
End\_Example: Segment 09



## Error messages on Interbus-S

If an error occurs during the transfer of parameter data, the Interbus-S master will receive an error message from the PSS.

Error class	Error code	Additional code	Key
8	0	1	PSS is switched off
8	0	2	PSS fails to react to request Reasons: 1. Error in user program 2. PSS has a major error
8	0	3	Unable to access user object. Reason: ST user program is in STOP
8	0	4	User not employing these objects
8	0	16	Access to SafetyBUS p error stack without entering device address
8	0	17	Stated device address not permitted
8	0	18	SafetyBUS p error stack has already been transferred in full. Requested sub-index can no longer be serviced.
8	0	19	Sub-index is not permitted
8	0	20	Timeout on transfer of SafetyBUS p error stack
8	0	30	Error on SafetyBUS p
8	0	31	SafetyBUS p is inactive

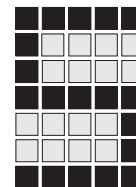


# Interbus-S

---

**Notes**





# Operation

---

## Commissioning

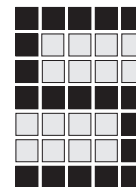
- Install the module as described in the “PSS 3000/3100 Installation Manual”.
- Set the required transmission rate and process/parameter data length for Interbus-S using the rotary switches on the front of the module. If you wish to set the PSS(1) IBS-S PCP module as a replacement for its predecessor the PSS(1) IBS-S, set the upper rotary switch to the required process data length and the lower switch to “30x 150”.



### NOTICE

The larger the data width, the longer the Interbus-S scan time will be!

- Connect bus subscriber: Connect the Interbus-S master or remote bus subscriber using the male 9-pin D-Sub connector “IBS-IN”; connect additional remote bus subscribers using the female 9-pin D-Sub connector “IBS-OUT”.
- If required, connect the supply voltage for the Interbus-S slave (“IBS PWR”) and switch on. The supply voltage connection is described in Chapter 3.
- Power up the PSS.
- Some settings for the PSS(1) IBS-S PCP need to be made in the IBS CMD software. Select “Configuration” and “Read In”. Then select the PSS(1) IBS-S PCP module using the right-hand mouse button. The “Description” dialogue box will open. Select “Parameter Channel” and set the following values:
  - Message Lengths Transmit: 240 Byte
  - Message Lengths Receive: 240 Byte
  - Only select “Client Services of Controller Board”, “Read” and “Write”.



# Operation

---

## Faults

If a fault occurs on the safety system or there is a wiring error, the PSS will switch to a STOP condition and output a message to the display. The error stack display in the programming device can be used to locate the error (see “PSS-Range System Description”).

If you have connected the Interbus-S slave to a separate external supply “IBS PWR” and this supply fails, the module will automatically switch to the PSS supply. The Interbus connection will be interrupted briefly as the module switches over. As soon as the supply is returned to the Interbus-S slave it will automatically be reinitialised and the Interbus-S can be operated once more. The Interbus-S master will receive an error message.

## Display elements

### LED “US”

Lights when a separate external supply voltage is present for the Interbus-S slave (“IBS PWR”).

### LED “RC”

Goes out when the cable connection for remote bus IN is defective or when the Interbus-S master is in reset. When both LEDs “US” and “RC” are lit, the Interbus-S slave is ready for operation.

### LED “RD”

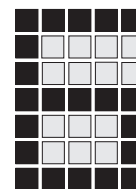
Lights when the remote bus and everything connected to “IBS OUT” is switched off (remote bus disable).

### LED “BA”

Lights when data is being transferred to Interbus-S.

### LED “TR”

Lights when parameter data is being transferred to Interbus-S.



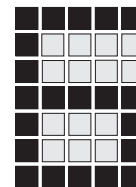
## Technical Details

### Electrical data

Supply voltage "IBS PWR"	24 VDC
Tolerance range "IBS PWR"	20 ... 30 VDC including residual ripple of max. $\pm 1.2$ V
Power consumption "IBS PWR"	7.5 W (2 kV galvanically isolated from PSS supply voltage)
Connection type "IBS PWR"	Screw connector
Current consumption	<p><i>Incl. "IBS-PWR" and excl. IBS-OPTOSUB connector:</i>            typ. 130 mA from PSS supply and            typ. 190 mA from "IBS-PWR"</p> <p><i>Incl. "IBS-PWR" and incl. IBS-OPTOSUB connector:</i>            typ. 130 mA from PSS supply and            typ. 280 mA from "IBS-PWR"</p> <p><i>Excl. "IBS-PWR" and excl. IBS-OPTOSUB connector:</i>            typ. 800 mA from PSS supply</p> <p><i>Excl. "IBS-PWR" and incl. IBS-OPTOSUB connector:</i>            typ. 1.4 A from PSS supply</p>

### Interbus-S

Device type	Slave with PCP channel
Application range	Non-safety-related applications
Status indicator	LEDs
Data width	0 ... 32 words, selectable (incl. PCP)
PCP channel	0, 1, 2 or 4 words, selectable
Transmission rate	500 kBaud or 2 MBaud, selectable
Connection	
IBS IN	Male 9-pin D-Sub connector (2 kV galvanically isolated from "IBS OUT" and "IBS PWR"; ability to connect an IBS-OPTOSUB connector with OPC)
IBS OUT	Female 9-pin D-Sub connector (2 kV galvanically isolated from "IBS IN" and "IBS PWR", ability to connect an IBS-OPTOSUB connector with OPC)



## Technical Details

Environmental data	
Protection type (EN 60529, 10/91)	IP 20, installed on module rack
Protection class (DIN VDE 0106, Part 1/A1, 04/90)	1
Mounting position	Vertical
Ambient temperature (DIN IEC 68-2-14, 06/87)	0 ... 60 °C
Storage temp. (EN 60068-2-1/-2, 03/93)	-25 ... +70 °C
Climatic suitability (DIN IEC 68-2-30, 09/86)	Max. 95 % r.h.
Vibration (EN 60068-2-6, 04/95)	Frequency range: 10 ... 100 Hz Amplitude: 0.1 mm, max. 5g
Vibration resistance (DIN IEC 68-2-29)	30g, 11 ms/10g, 16 ms
EMC	EN 50082-2, 03/95 EN 55011 A, 08/96
Mechanical data	
Space requirement	1 slot
Weight	PSS IBS-S PCP: 940 g PSS1 IBS-S PCP: 350 g

The names of products, goods and technologies used are trademarks of the companies concerned.

## Module code

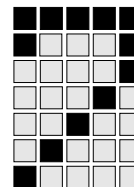
P9/P10: B528<sub>hex</sub>  
PSS 3000/3100 series: B520<sub>hex</sub>

If the rotary switch for setting the parameter data length is set to "30x 150", the module code corresponds to that of the PSS(1) IBS-S module:

P9/P10: B127<sub>hex</sub>  
PSS 3000/3100 series: B120<sub>hex</sub>

## Identification code

Parameter data length	Identification code
0	3 (03 <sub>hex</sub> )
1	243 (F3 <sub>hex</sub> )
2	240 (F0 <sub>hex</sub> )
4	241 (F1 <sub>hex</sub> )



## Appendix

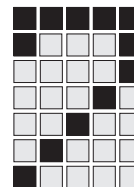
---

### **Address of Safety Network International e.V.**

Safety Network International e.V.  
Robert-Bosch-Straße 30  
D-73 760 Ostfildern  
Tel: (07 11) 34 09-1 18  
Fax: (07 11) 34 09-4 49  
E-Mail: [info@safetybus.com](mailto:info@safetybus.com)  
Internet: [www.safetybus.com](http://www.safetybus.com)

### **Address of Interbus-S Club**

INTERBUS-S Club  
User Group United Kingdom  
Unit 620 Wharfedale Road  
Wokingham  
Berkshire  
RG41 5TP  
Internet: [www.interbusclub.com](http://www.interbusclub.com)



# Appendix

---

**Notes**

