DB GB IBS SYS SWITCHING

Connecting and Disconnecting INTERBUS Devices/ Bus Segments

Data Sheet 6508C 10/2002

Bus Control Functions

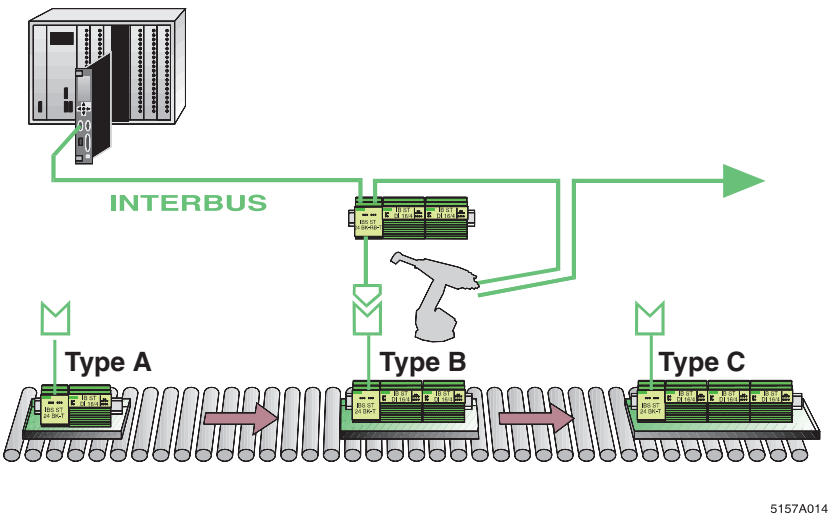


Figure 1 Switching bus parts

The INTERBUS control functions provide services, e.g., for switching INTERBUS devices/segments. These services can be used, for example, for flexible system design (docking and undocking system parts) and to respond to error states (e.g., disconnection of damaged system parts and continuous operation of the remaining bus system). During inspection work the entire bus does not have to be disconnected only the part on which work is to be carried out.

Previously devices/segments could be disconnected by the application and/or the control program (user-defined command) in the event of an error (for additional information, please refer to Page 2). In addition, firmware Version 4.60 or later enables isolated switching of INTERBUS remote bus branch and local bus interfaces. Thus data transmission in the rest of the INTERBUS system is not interrupted. Isolated disconnection is triggered by a bus error.

Switching by the Application

The controlling program connects or disconnects a bus segment.

Process: Switching by the Application

1. The control program deactivates the application.
2. A switching command (please note the warning below) enables the controller board to connect or disconnect the selected bus segment.



During the switching operation the application must be disconnected and operated in a safe state. The preset monitoring time (parameterizable bus shutdown time) is not active. The shutdown time, until the bus stops and is set to a safe state, increases in this case by the switching time of the system.

3. A confirmation by the controller board indicates the end of the switching operation.
4. Depending on the result of the switching operation the application can be reactivated.

Switching in the Event of an Error

In the event of an error, the INTERBUS diagnostics display a precise error message. The affected bus segment, which is installed after the error location, can be disconnected by the control program. The remaining bus configuration, which is not faulty, can be restarted and operated again.

Process: Switching in the Event of an Error

1. The controller board indicates that a bus error has occurred.



At this point the error location is still not known. The outputs remain set for the maximum shutdown time.

2. The control program deactivates the application.
3. The controller board searches for the error and indicates the precise error location, once it has been found.
4. A switching command enables the controller board to disconnect the affected bus segment
5. A confirmation by the controller board indicates the end of the switching operation.
6. Depending on the result of the switching operation the control program can start the bus via a switching command and activate the application.
7. It can be connected after repair work has been carried out (see "Switching by the Application "on page 2).



Please note, the bus error occurs again if the bus segment is connected without the error having been removed.

Isolated Switching

Isolated switching of branching INTERBUS remote bus branch and local bus interfaces does not interrupt data transmission in the rest of the INTERBUS system. With regard to functions, isolated disconnection is similar to the specific disconnection of a branch interface. The difference is that isolated disconnection is not triggered by a user-defined command but by a bus error.



This function must be enabled by defining the function of the branching remote bus branch or local bus interface for the corresponding bus terminal.

Preconditions

– The system used is dedicated SUPI 3 system.
– The bus terminal, which is to be used for isolated switching, contains a SUPI 3 OPC.



Please note that bus terminals in the Inline system are also referred to as bus couplers.

– Only remote bus branches and local bus interfaces can be switched in isolation.
– Error localization is not possible within a bus segment that was disconnected in isolation. In an Inline local bus branch the error is indicated by a diagnostic LED. In the event of an error, the diagnostic LED of the device installed directly after the error flashes.



Additional information on the Inline system can be found in the IB IL SYS PRO UM E User Manual (Order No. 27 43 04 8).



Please note that the simultaneous occurrence of several errors can result in the disconnection of the bus system, regardless of whether isolated disconnection has been activated.



Isolated disconnection can be set in IBS CMD Version 4.60 or later.

Isolated Disconnection in the Event of an Error

If the affected bus segment was parameterized as "can be disconnected in isolation", the controller board automatically disconnects the bus segment. The rest of the INTERBUS system continues to operate.

Process: Isolated Disconnection in the Event of an Error

In the event of an error in a bus segment:

1. The bus terminal automatically disconnects the remote bus branch or local bus interface, which is parameterized for isolated disconnection, and indicates the result to the controller board.



In IBS CMD (Version 4.60 or later), the "Bus System Messages" menu appears with a message that the bus segment was disconnected after a bus error. The bus terminal is displayed in IBS CMD with a superimposed yellow exclamation mark.

2. The controller board automatically configures itself so that the remaining INTERBUS devices continue to operate in isolation.
3. The BSA bit is set in the diagnostic status register.
4. An error message is transmitted from the controller board to the application program with the error indication "Device_Fail_Indication" 5340_{hex}. The value $0BD5_{hex}$ is transmitted in the Error_Code parameter. The Device_No parameter contains the device number of the bus terminal, whose branching interface was disconnected in isolation.



For additional information on the structure of error messages, please refer to the IBS SYS FW G4 UM E User Manual (Order No. 27 45 18 5).

Isolated Connection

Isolated connection as a response to isolated disconnection depends on the application. For example, the control program could evaluate an input. As soon as a service engineer has removed the bus error in the field and has acknowledged this, the control program reconnects the bus segment.



Please note, the bus error occurs again if the bus segment is connected without the error having been removed. The rest of the bus system continues to operate. Avoid automatic cyclic connection of the faulty bus segment.

The response from the controller to unsuccessful reconnection of the bus segment contains information about the error location. The error can thus be localized in the disconnected bus segment.

Process: Isolated Connection

Connection (the **error** was **not removed**):

If the faulty remote bus branch or local bus is reconnected the error message appears again in addition to the error location in the faulty segment.

– In IBS CMD, when attempting to reconnect the faulty bus segment, information about the error appears in the "Attention" dialog box. After confirming the "Detail…" button the error window displays the error code and error location (segment/position). The disconnected bus terminal is displayed with a superimposed yellow exclamation mark.
– The LD LED of the faulty bus terminal changes color in the event of an error and thus indicates that the bus segment is faulty.
– BSA is displayed in the controller board display.

Connection (the **error** was **removed**):

Once the error has been detected **and** removed, the disconnected bus segment can be reconnected (see "Switching by the Application "on page 2).

– BSA is no longer displayed in the controller board display.

Cascadability of Bus Terminals to be Switched in Isolation

The following figure gives an overview of the cascadability of bus terminals including their connected local busses or remote bus branches.

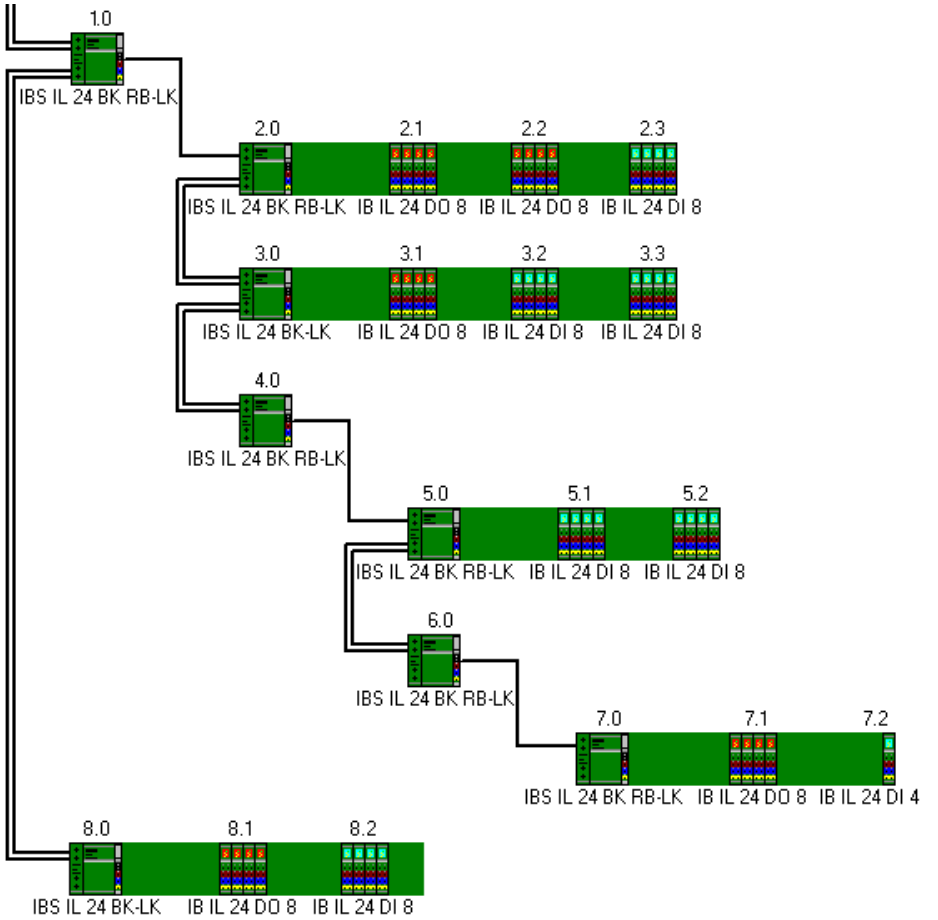


Figure 2 Bus structure with branched remote bus branches and local buses

Please observe the following:

For example, the parameterization of isolated switching for device 1.0 in IBS CMD has an effect on the entire bus configuration that is connected to the branching branch interface of device 1.0. IBS CMD prevents separate parameterization of the devices connected to the branching branch interface of device 1.0. The subsequent device 8.0 in the outgoing remote bus on device 1.0 must be parameterized separately for isolated disconnection.

For parameterization via firmware services please ensure that only one bus terminal is parameterized for isolated disconnection for a bus configuration with several successive devices, which are each connected to the branch interface of the previous device. For a bus configuration such as that illustrated in Figure 2 on page 6, device 1.0 **or** device 4.0 **or** device 6.0 **or** device 7.0 can be parameterized for isolated disconnection.



For additional information on the parameterization of isolated switching, please refer to "Parameterization of Isolated Switching "on page 8.

The following section describes the effects on bus segments, which are connected to devices (Dev) that are parameterized for isolated disconnection.



Devices 1.0, 2.0, and 4.0 may not be simultaneously parameterized for isolated disconnection.

Device	Effects After Activating Isolated Disconnection
1.0	As soon as an error occurs on a device in the entire bus segment (Dev 2.0 to Dev 7.2), the entire bus configuration is disconnected from the branch interface of Dev 1.0. This has no effect on device 8.0 connected to the outgoing interface and the bus configuration connected to it.
2.0	As soon as an error occurs on one of the devices (2.1, 2.2, 2.3), the entire bus segment is disconnected from the branch interface of Dev 2.0.
4.0	As soon as a bus error occurs on one of the subsequently connected devices (Dev 5.0 to Dev 7.2), the entire bus configuration, which is connected to the branch interface of Dev 4.0, is disconnected in isolation.
8.0	A bus error at Dev 8.1 or Dev 8.2 results in the disconnection of the bus configuration connected to Dev 8.0.

Parameterization of Isolated Switching

In IBS CMD only isolated disconnection is parameterized. Isolated connection is only carried out as a response to isolated disconnection, as described in "Isolated Connection "on page 5. Only devices that have been disconnected in isolation can be connected in isolation.

Parameterization in IBS CMD (Version 4.60 or Later)

To set isolated disconnection in IBS CMD, please proceed as follows:

– Open the context menu of the bus terminal whose branching interface is to be connected in isolation. Select the "Description…" entry.

– In the "Change Device Description" dialog box, select "Branching Bus Segment" from the drop-down menu under "Isolated Disconnection".

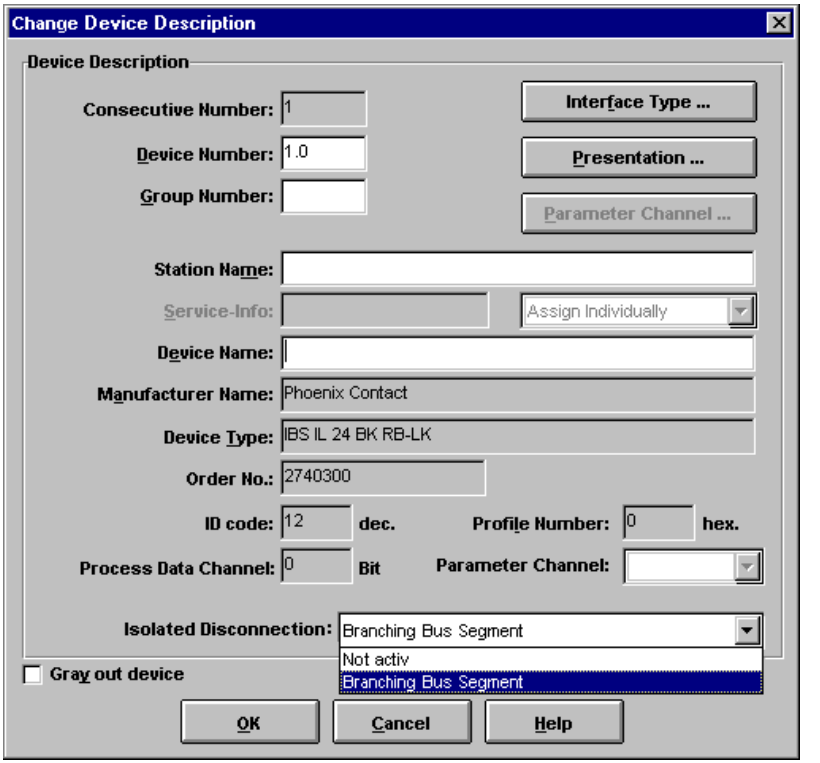


Figure 3 Setting isolated disconnection

Please note that isolated disconnection of a branching bus terminal interface is deactivated if this function is activated **and** deactivated again for a higher-level bus terminal. In this case, you must individually activate isolated disconnection for the corresponding lower-level bus terminals.
Example for device 6.0 in Figure 2 on page 6:
You have activated isolated disconnection for Dev 6.0. Then activate isolated disconnection for Dev 4.0. The same applies to all devices after Dev 4.0. When you deactivate isolated disconnection for Dev 4.0 again, this deactivation applies to all the devices after Dev 4.0. To use this function again for Dev 6.0 and its branching branch interface, isolated disconnection must be reactivated for Dev 6.0.

Parameterization Using Firmware Services

Isolated disconnection is parameterized via a new parameter in the configuration frame. Bit 5 of the **Used_Attributes** parameter is used in the following read/write services:

- Load_Configuration 0307_{hex}
- Read_Configuration 0309_{hex}
- Compare_Configuration 0317_{hex}
- Complete_Load_Configuration $030A_{hex}$
- Complete_Read_Configuration $030B_{hex}$

The **error characteristic** parameter is addressed by setting bit 5 of the Used_Attributes parameter. This is a 16-bit field is used. Setting bit 0 of this parameter means that the outgoing interface of the bus terminal will continue to operate in isolation in the event of an error. Bits 1 to 15 of this parameter are reserved.

The default value of the **error characteristic** of the parameter is "0". Read and write access to the parameter is enabled.

Service Descriptions

"Load_Configuration" Service

Task:

The configuration frame describes each of the specified INTERBUS devices in a separate numbered entry. The order and the numbering of the entries corresponds to the physical bus configuration including all alternatives.

This service transfers the configuration data to the controller board in the form of a list. Use the *Used_ Attributes* parameter to determine which attributes the list should contain.

The "Load_Configuration" service does not check the consistency among the attributes but only whether this data is permitted in principle, e.g., whether it is within the value range.

Prerequisite:

Ensure that the controller board has been prepared for transmission with the following services:

– "Control_Parameterization" ($030E_{hex}$)
– "Initiate_Load_Configuration" (0306_{hex})

Syntax:

Load_Configuration_Request **0307_{hex}**

Word 1	Code	
Word 2	Parameter_Count	
Word 3	Used_Attributes	
Word 4	Start_Entry_No	
Word 5	Entry_Count	
Word 6	Configuration_Entry	1. device
...	...	
	Configuration_Entry	nth device

Bit	15 0

Key:

Code:	0307_{hex} Command code of the service request
Parameter_Count:	Number of subsequent parameter words
	$xxxx_{hex}$ The value depends on the *Entry_Count* parameter and *Used_Atrributes* parameter.
Used_Attributes:	Choice of add-on attributes. The parameter is a 16-bit field in which every bit corresponds to an attribute. Set the corresponding bit to 1 on the attribute that you want to transmit (see the "Configuration_Entry" syntax on Page 12). Settings for the *Used_Attributes* parameter: Bit 0 Device number Bit 1 Device code Bit 2 Reserved (always 0_{bin}) Bit 3 Device level Bit 4 Logical group number Bit 5 Error characteristic Bits 6...15 Reserved (always 0_{bin}) Example: If the entries only consist of the device code and device level, enter the value $000A_{hex}$ for the *Used_Attributes* parameter (bit 1 and bit 3 are set).
Start_Entry_No:	Number of the first device whose attributes you wish to transmit.
Entry_Count:	Number of devices whose attributes you wish to transmit.
Configuration_Entry:	Attribute values of the individual devices to be transmitted according to their order in the physical bus configuration (see syntax on Page 12).



According to the following syntax, enter attributes in the "Configuration_Entry" parameter block that have been enabled with the *Used_ Attributes* parameter (disabled attributes are not entered).



When several entries with several attributes are loaded at the same time, first all the attributes of one entry are loaded, then those of the next entry.

Syntax	"Configuration_Entry"			Attributes:
Word x	Bus_Segment_No		Position	Device number
Word x+1	Length_Code		ID_Code	Device code
Word x+2	Device_Level			Device level
Word x+3	Group		Alternative	Log. group number
Word x+4	Reserved		Bit 0	Error characteristic
Bit	15 8		7 0	

Attributes:

Bus_Segment_No: Number of the bus segment where the device is located.
Value range: 01_{hex} ... FF_{hex} (1 ... 255_{dec})

Position: Physical location in the bus segment.
Value range:
00_{hex} ... 08_{hex} for local bus and ST compact station or
00_{hex} ... $3F_{hex}$ (63_{dec}) for installation local bus
The *Bus_Segment_No* and *Position* parameters together form the device number.

Length_Code: Length code
The length code refers to the address space required by the device in the host.

ID_Code: ID code
The ID code indicates the device type. It is printed as *Module Ident* in decimal notation on the modules (in hexadecimal notation on RT modules).
The *Length_Code* and *ID_Code* parameters together form the device number.

Device_Level: Device level
Value range: 00_{hex} ... $0F_{hex}$, (0 ... 15_{dec})

Group: Group to which the device belongs

Alternative: Alternative bus segment to which the device belongs
The *Group* and *Alternative* parameters together form the logical group number.

Error characteristic:
(Bit 0)
= 1: Isolated switching active
= 0: Isolated switching not active

(Bits 1 to 15) Reserved

Syntax: **Load_Configuration_Confirmation** **8307_{hex}**

Positive message

Word 1	Code
Word 2	Parameter_Count
Word 3	Result

Negative message

Word 1	Code
Word 2	Parameter_Count
Word 3	Result
Word 4	Add_Error_Info

Bit	15 0

Key:

Code: 8307_{hex} Message code of the service confirmation

Parameter_Count: Number of subsequent words

with a positive message:

0001_{hex} Always 1 parameter word

with a negative message:

0002_{hex} Always 2 parameter words

Result: Result of the service processing

0000_{hex} Indicates a positive message.
The controller board executed the service successfully.

$xxxx_{hex}$ Indicates a negative message.
The controller board could not execute the service successfully.
The *Result* parameter indicates why the service could not be executed.

Add_Error_Info: Additional information on the error cause

"Read_Configuration" Service

Task:

This service reads various entries of the configuration directory depending on the *Frame_Reference* and *Start_Entry_No* parameters.

Frame_ Reference	Start_ Entry_No	Entries Read by the Service
0000_{hex}	Not used (0000_{hex})	Header information of the configuration directory (CFG_OD_Header), i.e., the number of all configured configuration frames in ascending order.
> 0000_{hex}	0000_{hex}	Header information of the configuration frame (CFG_Header) selected with the *Frame_Reference* parameter.
> 0000_{hex}	> 0000_{hex}	Entries of the configuration frame (CFG_Entry) selected with the *Frame_Reference* parameter. Either the entire configuration frame or only one part, e.g., a single INTERBUS device description can be read.

Syntax: **Read_Configuration_Request** **0309_{hex}**

Word 1	Code
Word 2	Parameter_Count
Word 3	Frame_Reference
Word 4	Used_Attributes
Word 5	Start_Entry_No
Word 6	Entry_Count

Bit	15 0

Key:

Code:	0309_{hex}	Command code of the service request
Parameter_Count:	Number of subsequent words	
	0004_{hex}	4 parameter words
Frame_Reference:	Number of the configuration frame	
	0000_{hex}	Reads the configuration frame directory.
	$xxxx_{hex}$	Number of the configuration frame to be read.

Only used if *Frame_Reference* > 0000_{hex}

Used_Attributes:	Attributes to be read The parameter is a 16-bit field in which every bit corresponds to an attribute. Set the corresponding bit to 1 on the attributes to be read. Settings for the *Used_Attributes* parameter:	
	Bit 0	Device number
	Bit 1	Device code
	Bit 2	Reserved (always 0_{bin})
	Bit 3	Device level
	Bit 4	Logical group number
	Bit 5	Error characteristic
	Bits 6...7	Reserved (always 0_{bin})
	Bit 8	global_bus_error
	Bit 9	separate_bus_error
	Bit 10	Device status
	Bit 11	Additional module status information
	Bit 12	Visual diagnostic information
	Bits 13...15	Reserved (always 0_{bin})
Start_Entry_No:	Position of the first entry	
	0000_{hex}	Only reads the header information of the configuration frame.
	$xxxx_{hex}$	Reads the entries from the configuration directory from this number onwards.
Entry_Count:	Number of entries to be read	

The positive message transmits the desired entries from the configuration directory. Depending on the *Frame_Reference* and *Start_Entry_No* parameters in the service request, it has one of the following structures.

Syntax **Read_Configuration_Confirmation** **8309_{hex}**

1. structure

Positive message for service request with:
– *Frame_Reference* = 0000_{hex},
– *Start_Entry_No* not relevant (= 0000_{hex})

Word 1	Code	
Word 2	Parameter_Count	
Word 3	Result	
Word 4	More_Follows	
Word 5	Frame_Reference	= 0000_{hex}
Word 6	Current_Configuration	
Word 7	Configuration_Count	
Word 8	Frame_Reference 1	
...	...	
	Frame_Reference n	

2. structure

Positive message for service request with:
– *Frame_Reference* > 0000_{hex},
– *Start_Entry_No* = 0000_{hex}

Word 1	Code	
Word 2	Parameter_Count	
Word 3	Result	
Word 4	More_Follows	
Word 5	Frame_Reference	> 0000_{hex}
Word 6	Used_Attributes	Not relevant
Word 7	Start_Entry_No	= 0000_{hex}
Word 8	Frame_Device_Count	
Word 9	Active_Device_Count	
Word 10	Frame_IO_Bit_Count	
Word 11	Active_IO_Bit_Count	
Word 12	Frame_PCP_Device_Count	

Word 13	Active_PCP_Device_Count	
Word 14	Frame_PCP_Word_Count	
Word 15	Active_PCP_Word_Count	

Bit	15 0

3. structure

Positive message for service request with:

– Frame_Reference > 0000_{hex},

– Start_Entry_No > 0000_{hex}

Word 1	Code	
Word 2	Parameter_Count	
Word 3	Result	
Word 4	More_Follows	
Word 5	Frame_Reference	
Word 6	Used_Attributes	
Word 7	Start_Entry_No	
Word 8	Entry_Count	
Word 9	Configuration_Entry	1. device
...	...	
	Configuration_Entry	nth device

Negative message

Word 1	Code
Word 2	Parameter_Count
Word 3	Result
Word 4	Add_Error_Info

Bit	15 0

Key:

Code:	8309_{hex} Message code of the service confirmation
Parameter_Count:	Number of subsequent words
	with a positive message and if *Frame_Reference* = 0000_{hex}:
	$xxxx_{hex}$ = 5 + *Configuration_Count*
	with a positive message and if *Frame_Reference* > 0000_{hex} and *Start_Entry_No* = 0000_{hex}:
	$000D_{hex}$ 12 parameter words
	with a positive message and if *Frame_Reference* > 0000_{hex} and *Start_Entry_No* > 0000_{hex}:
	$xxxx_{hex}$ The value depends on the number of devices in the configuration frame and the number of enabled attributes.
	with a negative message:
	0002_{hex} 2 parameter words
Result:	Result of the service processing
	0000_{hex} Indicates a positive message. The service request has been executed successfully. The data is available in the following parameters.
	$xxxx_{hex}$ Indicates a negative message. The controller board could not execute the service successfully. The *Result* parameter indicates why the service could not be executed (see also *Add_Error_Info*).
Add_Error_Info:	Additional information on the error cause
More_Follows:	0000_{hex} Indicates that all requested entries are contained in the service confirmation.
	0001_{hex} Indicates that the service confirmation does not contain all requested entries, as the amount of data is larger than the mailbox (MXI) that is available for the services. Call the service again to read the remaining data.
Frame_Reference:	Number of the read configuration frame. The parameter contains the value that was entered with the service request.
Current_Configuration:	Number of the currently activated configuration frame
Configuration_Count:	Number of configured configuration frames
Frame_Reference x:	Number of all configured configuration frames in ascending order

Frame_Device_Count:	Number of configured INTERBUS devices in the selected configuration frame
Active_Device_Count:	Number of active INTERBUS devices in the selected configuration frame
Frame_IO_Bit_Count:	Number of configured I/O bits in the selected configuration frame
Active_IO_Bit_Count:	Number of active I/O bits in the selected configuration frame
Frame_PCP_Device_ Count:	Number of configured PCP devices in the selected configuration frame
Active_PCP_Device_ Count:	Number of active PCP devices in the selected configuration frame
Frame_PCP_Word_ Count:	Number of configured PCP words in the selected configuration frame
Active_PCP_Word_ Count:	Number of active PCP words in the selected configuration frame
Used_Attributes:	Read attributes The parameter contains the value that was entered with the service request.
Start_Entry_No:	Position of the first entry or 0000_{hex} if only the header information was read.
Entry_Count:	Number of entries that are transmitted by the service confirmation. The *More_Follows* parameter indicates if there are further entries.
Configuration_Entry:	Selected entries in the order of the physical bus configuration. The attributes contained in every entry are enabled in the service request by the *Used_Attributes* parameter (see the "Configuration_Entry" syntax on Page 20).



A configuration entry for a device does not have to contain all attributes. If an attribute is not enabled in the service request by the *Used_Attributes* parameter, the configuration entry is reduced by the relevant data words.

In the following, the structure of a configuration entry is shown where **all** attributes are enabled.

Syntax	"Configuration_Entry"			Attribute:
Word x	Bus_Segment_No		Position	Device number
Word x+1	Length_Code		ID_Code	Device code
Word x+2	Device_Level			Device level
Word x+3	Group		Alternative	Logical group number
Word x+4	Reserved		Bit 0	Error characteristic
Word x+5	SUPI_Type		Add_Info	"global_bus_ error"
Word x+6	Transmission_Error			
Word x+7	Device_Error			
Word x+8	Transmission_Fail_DO			"separate_ bus_error"
Word x+9	Transmission_Fail_DI			
Word x+10	MAU_FAIL_DO			
Word x+11	MAU_FAIL_DI			
Word x+12	MAU_Warning_DO		MAU_Warning_DI	
Word x+13	Out1_FAIL			
Word x+14	Out2_FAIL			
Word x+15	Reconfigure_Request			
Word x+16	Peripheral_State			
Word x+17	Microprocessor_Watchdog			
Word x+18	Device_State			Device status
Word x+19	Channel 16...31			Additional module status information
Word x+20	Channel 0...15			
Word x+21	Initiator		Power	
Word x+22	Length		Optical Info Forward Line	Visual diagnostic information
Word x+23	Optical Info Return Line		- -	

Bit	15 8	7 0

Key:

Attribute: Device number

Bus_Segment_No:	Number of the bus segment where the INTERBUS device is located. Value range: 00_{hex} to FF_{hex} (0_{dec} to 255_{dec})
Position:	Physical location in the bus segment. Value range: 00_{hex} to 08_{hex} (0_{dec} to 8_{dec}) for local bus and ST compact station or 00_{hex} to 40_{hex} (0_{dec} to 64_{dec}) for installation local bus

Attribute: Device code

Length_Code:	Length code The length code refers to the address space required by the INTERBUS device in the host.
ID_Code:	ID code The ID code describes the INTERBUS device function. It is printed as *Module Ident* in decimal notation on the modules (in hexadecimal notation on RT modules).

Attribute: Device level

Device_Level:	Device level The device level describes the branch level at which the device is located. Value range: 00_{hex} to $0F_{hex}$ (0 to 15_{dec})

Attribute: Logical group number

Group:	Group to which the device belongs
Alternative:	Alternative bus segment to which the device belongs

Attribute: Error characteristic

Bit 0	= 1:	Isolated switching active
	= 0:	Isolated switching not active
Bits 1 to 15	Reserved	

Attribute: "global_bus_error"

SUPI_Type:	SUPI type	
	xx00	Old SUPIs and SUPI 2
	xx01	SUPI 1
	xx03	SUPI 3 and later
Add_Info:	SUPI chip identification	
	00_{hex}	Earlier SUPI version than SUPI 3
	$A0_{hex}$	IB8052

	$A2_{hex}$	LPC 2
	$A3_{hex}$	LPC 1
	$A4_{hex}$	SUPI 3 DPC
	$A5_{hex}$	SUPI 3
	$D0_{hex}$	SUPI 3 OPC
	FF_{hex}	Reserved

Transmission_Error: Transmission error

Device_Error: I/O error

Attribute: "separate_bus_error"

Transmission_Fail_DO: Counter for transmission errors in the data forward path

Transmission_Fail_DI: Counter for transmission errors in the data return path

MAU_FAIL_DO: Counter for cable interrupts in the data forward path

MAU_FAIL_DI: Counter for cable interrupts in the data return path

MAU_Warning_DO: Counter for deterioration of the transmission quality in the forward path (for optical fiber transmission), overloaded internal power source for chip LPC 2

MAU_Warning_DI: Counter for deterioration of the transmission quality in the return path (for optical fiber transmission), temperature of the IBS protocol chip too high for chip LPC 2

Out1_FAIL: Counter for errors on the RBST signal (jumper in the connector) on the outgoing interface or error on the unused OUT1 interface.

Out2_FAIL: Counter for errors on the LBST signal (jumper in the connector) on the outgoing interface or error on the unused OUT2 interface.

Reconfigure_Request: Counter for reconfiguration requests, voltage too low for the initiators for chip LPC 2.

Peripheral_Fault: Counter for peripheral faults (I/O errors)

Microprocessor_Watchdog: Counter for resets of the connected microprocessor, for chip LPC 2: exceeding the permitted output current of the power drive.

Attribute: Device state

Device_State: Status of the device

Bit 0, 1	Not used	
Bit 2	1_{bin}	Device active
	0_{bin}	Device not active
Bit 3, 4	Not used	

	Bit 5	1_{bin} Device jumpered 0_{bin} Device not jumpered
	Bit 6...15	Reserved

Attribute: Additional module status information

Channel 16...31:	Bits 0...15	A set bit indicates a peripheral fault at one of the module channels 16 to 31.
Channel 0...15:	Bits 0...15	A set bit indicates a peripheral fault at one of the module channels 0 to 15.
Initiator:	Bits 8...15	A set bit indicates the failure of the initiator supply at one of the module groups 1 to 8.
Power:	Bits 0...3	A set bit indicates the failure of one of the supply voltages U1 to U4.

Attribute: Visual diagnostic information

Length:	Bits 8...13	Length of the incoming path
	Bit 14	Optical medium
	Bit 15	Enable status
Optical Info Forward Line:	Bits 0...3	Power level
	Bit 4	Reserved
	Bit 5	Distortion
	Bit 6, 7	Reserved
Optical Info Return Line:	Bits 8...11	Power level
	Bit 12	Reserved
	Bit 13	Distortion
	Bit 14, 15	Reserved

"Compare_Configuration" Service

Task:

This service transfers the configuration data to the controller board in the form of a list of entries. Use the *Used_ Attributes* parameter to determine which attributes the list should contain.

The controller board compares this list with the configuration frame specified with the *Frame_Reference* parameter. If the list and the configuration frame are not identical, a negative confirmation is generated and the controller board transmits information about the error in the form of error codes.

Prerequisite:

Ensure that the configuration frame to be specified with the *Frame_Reference* parameter is available on the controller board.



The configuration frame describes **each** of the specified INTERBUS devices in a separate numbered entry. The order and the numbering of the entries corresponds to the physical bus configuration including all alternatives.

Syntax: **Compare_Configuration_Request** **0317_{hex}**

Word 1	Code	
Word 2	Parameter_Count	
Word 3	Frame_Reference	
Word 4	Used_Attributes	
Word 5	Start_Entry_No	
Word 6	Entry_Count	
Word 7	Configuration_Entry	1. device
...	...	
	Configuration_Entry	nth device

Bit	15 0

Key:

Code:	0317_{hex} Command code of the service request
Parameter_Count:	Number of subsequent words $xxxx_{hex}$ The value depends on the number of entries and the number and type of attributes that you want to compare.
Frame_Reference:	Number of the configuration frame with which the controller board is to compare the transmitted list.
Used_Attributes:	Attributes of the configuration frame to be read The parameter is a 16-bit field in which every bit corresponds to an attribute. Set the corresponding bits to 1 on the attribute that you want to read. Settings for the *Used_Attributes* parameter: Bit 0 Device number Bit 1 Device code Bit 2 Reserved (always 0_{bin}) Bit 3 Device level Bit 4 Logical group number Bit 5 Error characteristic Bits 6...7 Reserved (always 0_{bin}) Bit 8 global_bus_error Bit 9 separate_bus_error Bit 10 Device status Bits 11...15 Reserved (always 0_{bin})
Start_Entry_No:	Number of the first device in the configuration frame whose attributes are to be compared with the transmitted list.
Entry_Count:	Number of entries you wish to transmit to the controller board.
Configuration_Entry:	Attribute values of the individual devices Only enter the attribute values that you selected with the *Used_Attributes* parameter and enter the entries in the order of the physical bus configuration. The *Configuration_Entry* parameters are described in the "Read_Configuration" service on Page 14.

Syntax: **Compare_Configuration_Confirmation** **8317_{hex}**

Positive message

Word 1	Code
Word 2	Parameter_Count
Word 3	Result

Negative message

Word 1	Code
Word 2	Parameter_Count
Word 3	Result
Word 4	Add_Error_Info

Bit	15 0

Key:

Code: 8317_{hex} Message code of the service confirmation

Parameter_Count: Number of subsequent words

with a positive message:

0001_{hex} 1 parameter word

with a negative message:

0002_{hex} 2 parameter words

Result: Result of the service processing

0000_{hex} Indicates a positive message.
The controller board executed the service successfully.

$xxxx_{hex}$ Indicates a negative message.
The controller board could not execute the service successfully.
The *Result* parameter indicates why the service could not be executed.

Add_Error_Info: Additional information on the error cause

"Complete_Load_Configuration" Service

Task:

This service transmits configuration data to the INTERBUS master in the form of attributes selected with the *Used_Attributes* parameter. It is specially adapted to the PLC programming requirements.

Prerequisite:

Ensure that the controller board has been prepared for transmission with the following services:

– "Control_Parameterization" ($030E_{hex}$)
– "Initiate_Load_Configuration" (0306_{hex})

Remark:

This service can be understood as a meta service for the "Load_Configuration" service (0307_{hex}, see Page 10). This service transmits all entries of the configuration frame *(Start_Entry_No* = 1).

Syntax: **Complete_Load_Configuration_Request** **$030A_{hex}$**

Word 1	Code	
Word 2	Parameter_Count	
Word 3	Used_Attributes	
Word 4	Entry_Count	
Word 5	Configuration_Entry	1. device
...	...	
	Configuration_Entry	nth device

Bit	15 0

Key:

Code:	$030A_{hex}$ Command code of the service request
Parameter_Count:	Number of subsequent words
	$xxxx_{hex}$ The value depends on the number of entries and the number and type of attributes that you want to transmit.
Used_Attributes:	Attributes to be transmitted The parameter is a 16-bit field in which every bit corresponds to an attribute. Set the corresponding bits to 1 on the attribute that you want to transmit. Settings for the *Used_Attributes* parameter:
	Bit 0 Device number
	Bit 1 Device code
	Bit 2 Reserved (always 0_{bin})
	Bit 3 Device level
	Bit 4 Logical group number
	Bit 5 Error characteristic
	Bits 6...15 Reserved (always 0_{bin})
Entry_Count:	Number of entries to be transmitted
Configuration_Entry:	Attribute values of the individual devices Only enter the attribute values that you selected with the *Used_Attributes* parameter and enter the entries in the order of the physical bus configuration. For the description of the *Configuration_Entry* parameters see "Read_Configuration" service (0309_{hex}) on Page 14.

Syntax: **Complete_Load_Configuration_Confirmation** **$830A_{hex}$**

Positive message

Word 1	Code
Word 2	Parameter_Count
Word 3	Result

Negative message

Word 1	Code
Word 2	Parameter_Count
Word 3	Result
Word 4	Add_Error_Info

Bit	15 0

Key:

Code: $830A_{hex}$ Message code of the service confirmation

Parameter_Count: Number of subsequent words

with a positive message:

0001_{hex} 1 parameter word

with a negative message:

0002_{hex} 2 parameter words

Result: Result of the service processing

0000_{hex} Indicates a positive message.
The controller board executed the service successfully.

$xxxx_{hex}$ Indicates a negative message.
The controller board could not execute the service successfully.
The *Result* parameter indicates why the service could not be executed.

Add_Error_Info: Additional information on the error cause

"Complete_Read_Configuration" Service

Task:

This service reads entries in the configuration directory in the form of one or more columns, which have been selected with the *Used_Attributes* parameter. It is specially adapted to the PLC programming requirements.

Remark:
This service can be understood as a meta service for the "Read_Configuration" service (0309_{hex}). The *Start_Entry_No* parameter does not need to be specified, since this service reads all entries of the configuration frame *(Start_Entry_No* = 1).

Syntax: **Complete_Read_Configuration_Request** **$030B_{hex}$**

Word 1	Code
Word 2	Parameter_Count
Word 3	Used_Attributes

Bit	15 0

Key:

Code: $030B_{hex}$ Command code of the service request

Parameter_Count: Number of subsequent words

0001_{hex} Always 1 parameter word

Used_Attributes: The parameter is a 16-bit field in which every bit corresponds to an attribute. Set the corresponding bits to 1 on the attribute that you want to read. Settings for the *Used_Attributes* parameter:

Bit 0	Device number
Bit 1	Device code
Bit 2	Reserved (always 0_{bin})
Bit 3	Device level
Bit 4	Logical group number
Bit 5	Error characteristic
Bits 6...10	Reserved (always 0_{bin})
Bit 11	Additional module status information
Bit 12	Visual diagnostic information
Bits 13...15	Reserved (always 0_{bin})

Syntax: **Complete_Read_Configuration_Confirmation** **$830B_{hex}$**

Positive message

Word 1	Code	
Word 2	Parameter_Count	
Word 3	Result	
Word 4	More_Follows	
Word 5	Frame_Reference	
Word 6	Used_Attributes	
Word 7	Start_Entry_No	0001_{hex}
Word 8	Entry_Count	
Word 9	Configuration_Entry	1. device
...	...	
	Configuration_Entry	nth device

Negative message

Word 1	Code	
Word 2	Parameter_Count	
Word 3	Result	
Word 4	Add_Error_Info	

Bit	15 0	

Key:

Code:	$830B_{hex}$ Message code of the service confirmation
Parameter_Count:	Number of subsequent words with a positive message: $xxxx_{hex}$ The value depends on the number of entries and the number and type of attributes that have been read. with a negative message: 0002_{hex} 2 parameter words
Result:	Result of the service processing 0000_{hex} Indicates a positive message. The controller board executed the service successfully. $xxxx_{hex}$ Indicates a negative message. The controller board could not execute the service successfully. The *Result* parameter indicates why the service could not be executed.
Add_Error_Info:	Additional information on the error cause
More_Follows:	0000_{hex} Indicates that all requested entries are contained in the service confirmation. 0001_{hex} Indicates that the service confirmation does not contain all requested entries, as the amount of data is larger than the mailbox (MXI) that is available for the services. Call the "Read_Configuration" service (0309_{hex}) to read the remaining data.
Frame_Reference:	Number of the active configuration frame
Used_Attributes:	Read attributes The parameter contains the value that was entered with the service request.
Start_Entry_No:	Number of the first entry 0001hex With this service all entries are read, starting with the first entry.
Entry_Count:	Number of entries that are transmitted by the service confirmation.
Configuration_Entry:	Entries in the order of the physical bus configuration The attributes contained in every entry are enabled in the service request by the *Used_Attributes* parameter. For the description of the *Configuration_Entry* parameters see "Read_Configuration" service (0309_{hex}) on Page 14.

Firmware Error Messages

$0A5B_{hex}$

The configuration data transmitted to the controller board using the "Compare_Configuration" service (0317_{hex}) does not match the configuration frame data specified with the *Frame_Reference* parameter.

The device specified by the *Add_Error_Info* parameter, which is to be switched in isolation, has been entered differently.

Remedy: Compare the lists of the transmitted configuration data with the specified configuration frame.

Add_Error_Info: Line number of the configuration frame, which does not match the transmitted configuration data.

$0CA0_{hex}$

Meaning: The controller board could not start up the bus configuration.

Cause: An attempt was made to parameterize a bus for isolated disconnection, which does not contain only SUPI 3 devices.

Remedy: Only use devices with SUPI 3.



Only set isolated switching in a dedicated SUPI 3 system.

$0A80_{hex}$

Meaning: The configuration was rejected, the bus cannot be started up.

Cause: You parameterized a bus, which consists of SUPI 3 devices parameterized for isolated disconnection. The bus terminal, whose branching branch interface was parameterized for isolated disconnected, does not have an OPC chip.

Remedy: Only use devices with SUPI 3 OPC.

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

+ 49 - (0) 52 35 - 3-00

+ 49 - (0) 52 35 - 3-4 12 00

www.phoenixcontact.com

Worldwide Locations:
www.phoenixcontact.com/salesnetwork

© Phoenix Contact 10/2002 Technical modifications reserved TNR 90 09 74 1