



ibaPDA-Request-S7-DP/PN

Request Data Interface to SIMATIC S7

Manual
Issue 3.10

Measurement Systems for Industry and Energy
www.iba-ag.com

Manufacturer

iba AG
Koenigswarterstr. 44
90762 Fuerth
Germany

Contacts

Main office	+49 911 97282-0
Fax	+49 911 97282-33
Support	+49 911 97282-14
Engineering	+49 911 97282-13
E-mail	iba@iba-ag.com
Web	www.iba-ag.com

Unless explicitly stated to the contrary, it is not permitted to pass on or copy this document, nor to make use of its contents or disclose its contents. Infringements are liable for compensation.

© iba AG 2021, All rights reserved.

The content of this publication has been checked for compliance with the described hardware and software. Nevertheless, discrepancies cannot be ruled out, and we do not provide guarantee for complete conformity. However, the information furnished in this publication is updated regularly. Required corrections are contained in the following regulations or can be downloaded on the Internet.

The current version is available for download on our web site www.iba-ag.com.

Version	Date	Revision - Chapter / Page	Author	Version SW
3.10	09-2021	Note (6.1.2)	st	7.0.0

Windows® is a brand and registered trademark of Microsoft Corporation. Other product and company names mentioned in this manual can be labels or registered trademarks of the corresponding owners.

Content

1	About this manual.....	8
1.1	Target group and previous knowledge	8
1.2	Notations	9
1.3	Used symbols.....	10
2	System requirements	11
3	ibaPDA-Request-S7-DP/PN	13
3.1	General information	13
3.1.1	Overview.....	14
3.1.2	How does the symbolic Request work?	15
3.2	Configuration and engineering SIMATIC S7	16
3.3	Configuration and engineering ibaPDA.....	16
3.3.1	General interface settings.....	16
3.3.2	General module settings.....	16
3.3.3	Signal configuration	18
3.3.3.1	Selection via the absolute address of the operands.....	19
3.3.3.2	Selection via the operand symbols.....	20
3.3.3.3	Measuring the CFC connectors.....	24
3.3.4	Address books	27
4	Request-S7 Variants	32
4.1	Request-S7 for ibaBM-PN	32
4.1.1	General information	32
4.1.2	Configuration and engineering SIMATIC S7-300, S7-400 and WinAC	33
4.1.2.1	Description of the Request blocks	34
4.1.2.2	Hardware configuration.....	37
4.1.2.3	Configuration in STEP 7.....	38
4.1.3	Configuration and engineering SIMATIC S7-1500	40
4.1.3.1	Description of the Request blocks	41
4.1.3.2	Network configuration.....	43
4.1.3.3	Configuration in STEP 7.....	43
4.1.3.4	Device configuration.....	45
4.1.4	Configuration and engineering ibaPDA.....	47
4.1.4.1	General interface settings.....	47

4.1.4.2	General module settings.....	49
4.1.4.3	Connection settings	49
4.1.4.4	Module S7 Request.....	57
4.1.4.5	Module S7 Request Decoder	57
4.1.5	Diagnostics.....	58
4.2	Request-S7 for ibaBM-PN in redundancy mode	59
4.2.1	General information	59
4.2.2	Configuration and engineering SIMATIC S7-300, S7-400 and WinAC	60
4.2.2.1	Description of the Request blocks	60
4.2.2.2	Hardware configuration.....	60
4.2.2.3	Configuration in STEP 7.....	62
4.2.3	Configuration and engineering ibaPDA.....	62
4.3	Request-S7 for ibaBM-DP	64
4.3.1	General information	64
4.3.1.1	ibaCom-L2B compatibility mode	66
4.3.2	Configuration and engineering SIMATIC S7-300, S7-400 and WinAC	66
4.3.2.1	Description of the Request-FC ibaDP_Req (FC122)	67
4.3.2.2	Hardware configuration.....	69
4.3.2.3	Configuration in STEP 7 (STL, LAD, FBD)	71
4.3.2.4	Configuration in STEP 7 (CFC)	72
4.3.3	Configuration and engineering SIMATIC S7-1500.....	74
4.3.3.1	Description of the Request blocks	74
4.3.3.2	Network configuration.....	76
4.3.3.3	Configuration in STEP 7.....	77
4.3.3.4	Device configuration.....	45
4.3.4	Configuration and engineering ibaPDA.....	80
4.3.4.1	General interface settings.....	80
4.3.4.2	General module settings.....	84
4.3.4.3	Connection settings	84
4.3.4.4	Module S7 Request.....	84
4.3.4.5	Modules S7 Request Decoder.....	84
4.3.4.6	Module S7 Request (ibaCom-L2B compatible)	85
4.3.4.7	Module S7 Request Dig512 (ibaCom-L2B compatible)	86
4.3.5	Diagnostics.....	87
4.4	Request-S7 for ibaBM-DP in redundancy mode	88

4.4.1	General information	88
4.4.2	Configuration and engineering SIMATIC S7-300, S7-400 and WinAC	89
4.4.2.1	Description of the Request-FC ibaDP_Req_H (FC123)	89
4.4.2.2	Hardware configuration	92
4.4.2.3	Configuration in STEP 7 (STL, LAD, FBD)	94
4.4.3	Configuration and engineering ibaPDA.....	96
4.5	Request-S7 for ibaBM-DPM-S	98
4.5.1	General information	98
4.5.2	Configuration and engineering SIMATIC S7-300, S7-400 and WinAC	99
4.5.3	Configuration and engineering ibaPDA.....	100
4.5.3.1	General interface settings.....	100
4.5.3.2	General module settings.....	102
4.5.3.3	Connection settings	103
4.5.3.4	Module S7 Request.....	107
4.5.3.5	Modules S7 Request Decoder.....	108
4.5.3.6	Module Dig512 S7 Request	108
4.5.4	Diagnostics.....	108
4.6	Request-S7 for ibaBM-DPM-S in redundancy mode.....	110
4.6.1	General information	110
4.6.2	Configuration and engineering SIMATIC S7-300, S7-400 and WinAC	111
4.6.3	Configuration and engineering ibaPDA.....	111
4.7	Request-S7 for ibaCom-L2B	113
4.7.1	General information	113
4.7.2	Configuration and engineering SIMATIC S7-300, S7-400 and WinAC	114
4.7.2.1	Description of the Request FCs.....	114
4.7.2.2	Hardware configuration.....	121
4.7.2.3	Configuration in STEP 7 (STL, LAD, FBD)	121
4.7.2.4	Configuration in STEP 7 (CFC)	126
4.7.3	Configuration and engineering ibaPDA.....	129
4.7.3.1	General interface settings.....	129
4.7.3.2	General module settings.....	130
4.7.3.3	Module L2B S7 Request.....	130
4.7.3.4	Module L2B S7 Request Dig512.....	130
4.7.4	Diagnostics.....	132

4.7.4.1	ibaCom-L2B card.....	132
4.7.4.2	Module diagnostics	133
5	Diagnostics.....	134
5.1	Checking the license	134
5.2	Log files.....	135
5.3	Connection diagnostics with PING.....	136
5.4	Connection diagnostics via PG/PC interface	137
5.5	Module diagnostics.....	138
6	Appendix	139
6.1	Iba S7 library	139
6.1.1	iba S7 library for SIMATIC Manager	139
6.1.1.1	Integrating the library.....	140
6.1.1.2	Using the blocks.....	142
6.1.2	iba S7 library for SIMATIC TIA portal.....	143
6.1.2.1	Integrating the library.....	145
6.1.2.2	Using the blocks.....	147
6.2	Application examples.....	148
6.3	S7 cycle time measurements	150
6.3.1	ibaCom-L2B.....	150
6.3.1.1	S7-CPU's with external DP interface (CP)	150
6.3.1.2	S7-CPU's with external DP interface (CP)	150
6.3.2	ibaBM-DP.....	151
6.3.3	ibaBM-PN.....	152
6.4	Setting PG/PC interface / defining new access point.....	153
6.5	S7 routing	157
6.5.1	Routing from Ethernet to Ethernet	157
6.5.1.1	Configuration of STEP 7/ NetPro	158
6.5.1.2	ibaPDA configuration	160
6.5.2	Routing from Ethernet to PROFIBUS	161
6.5.2.1	Configuration STEP 7/ NetPro.....	162
6.5.2.2	ibaPDA configuration	164
6.6	Replacing Request-S7 on ibaCom-L2B by ibaBM-DP	166
6.7	Error codes of Request blocks	168

6.8 Use of MPI/DP-TCP-adapters..... 172

7 Support and contact..... 173

1 About this manual

This manual describes the use of the Request data interface to SIMATIC S7.

The product *ibaPDA-Request-S7-DP/PN* is an extension of *ibaPDA* for the direct access to S7 symbols and S7 operands when recording data from SIMATIC S7-CPU's. In this manual only the extensions and deviations are shown. For all other functions and operating functions, please see the *ibaPDA* manual.

Other documentation



This documentation provides supplementary information to the general *ibaPDA* manual.

1.1 Target group and previous knowledge

This documentation addresses qualified professionals, who are familiar with handling electrical and electronic modules as well as communication and measurement technology. A person is regarded as a professional if he/she is capable of assessing the work assigned to him/her and recognizing possible risks on the basis of his/her specialist training, knowledge and experience and knowledge of the standard regulations.

This documentation in particular addresses persons, who are concerned with the configuration, test, commissioning or maintenance of Programmable Logic Controllers of the supported products. For the handling of *ibaPDA-Request-S7-DP/PN* the following basic knowledge is required and/or useful:

- Windows operating system
- Basic knowledge *ibaPDA*
- Basic knowledge network technology
- Knowledge of configuration and operation of SIMATIC S7 PLCs

1.2 Notations

In this manual, the following notations are used:

Action	Notation
Menu command	Menu <i>Logic diagram</i>
Calling the menu command	<i>Step 1 – Step 2 – Step 3 – Step x</i> Example: Select the menu <i>Logic diagram - Add - New function block</i> .
Keys	<Key name> Example: <Alt>; <F1>
Press the keys simultaneously	<Key name> + <Key name> Example: <Alt> + <Ctrl>
Buttons	<Key name> Example: <OK>; <Cancel>
File names, paths	"Filename", "Path" Example: "Test.doc"

1.3 Used symbols

If safety instructions or other notes are used in this manual, they mean:

Danger!



The non-observance of this safety information may result in an imminent risk of death or severe injury:

- Observe the specified measures.
-

Warning!



The non-observance of this safety information may result in a potential risk of death or severe injury!

- Observe the specified measures.
-

Caution!



The non-observance of this safety information may result in a potential risk of injury or material damage!

- Observe the specified measures
-

Note



A note specifies special requirements or actions to be observed.

Tip



Tip or example as a helpful note or insider tip to make the work a little bit easier.

Other documentation



Reference to additional documentation or further reading.

2 System requirements

The following system requirements apply for the use of the data interface *ibaPDA-Request-S7-DP/PN*:

- *ibaPDA* v7.0 or higher
- Basic license for *ibaPDA*
- Additional license for *ibaPDA-Request-S7-DP/PN*
- *ibaBM-PN*, *ibaBM-DP*, *ibaBM-DPM-S* or *ibaCom-L2B*
- *ibaFOB-D* card on the *ibaPDA-PC* when using *ibaBM-PN*, *ibaBM-DP* or *ibaBM-DPM-S* for connecting via FO conductor
- SIMATIC S7 controller S7-300, S7-400, S7-400H, S7-1500, WinAC
- In case PC/CP connections are used:
 - SIMATIC STEP 7 or SIMATIC NET or
 - SIMATIC TIA Portal
- SIMATIC CFC (beginning with V 6.0), if signals are to be chosen by drag & drop

For integrating the Request blocks in the S7 program:

- SIMATIC STEP 7 V5.4 SP5 or higher, or
- SIMATIC STEP 7 (TIA Portal) V14 SP1 or higher

For more requirements on the used PC hardware and the supported operating systems, please see the *ibaPDA* documentation.

System restrictions

- Access to optimized data blocks of S7-1500 controllers is not supported.
- Access to S7-1200 controllers is not supported.
- Connectors of CFC blocks, which have constant values assigned, have no operand address. They are marked as constant in the address book and cannot be selected as signal.
- If FB function blocks are used in CFC, the internal static variables of the FB are also displayed in the address book, because they are treated in exactly the same way as connectors by the compiler. These are to be ignored.
- For data acquisition, *ibaPDA* supports only the following data types:
 - BOOL, BYTE, WORD, DWORD, INT, DINT, REAL, TIME, CHAR

All other data types exist in the address book but cannot be entered in the signal list.

- For functions FC with connections of the data type STRING, POINTER, STRUCT or ANY under CFC, the interpretation of the SCL code does not work, as there are no references to the data types available in the source.

Licenses

Order no.	Name	Description
31.001310	ibaPDA-Request-S7-DP/PN	Extension license for an ibaPDA system to use the Request-S7 function via ibaBM-PN, ibaBM-DP, ibaBM-DPM-S or ibaCom-L2B

Table 1: Available licenses

3 ibaPDA-Request-S7-DP/PN

3.1 General information

ibaPDA-Request-S7-DP/PN is suitable for the measurement data acquisition via PROFIBUS and PROFINET. The measurement data are sent actively from the controller to *ibaPDA* via the PROFIBUS slaves or PROFINET devices - which are realized in the used device (e.g. *ibaBM-DP*). For this purpose, one or more Request blocks (FB/FC+DBs) have to be integrated per slave/device in the program of the S7-CPU. These Request blocks serve to send cyclically the S7 operands which have been selected by the user within *ibaPDA* via PROFIBUS/PROFINET to be recorded in *ibaPDA*. When modifying the signal selection, no modifications in the S7 program are needed.

The signals to be measured can be comfortably selected either by the absolute operand address or by the symbolic name with support of the *ibaPDA* address book browser. This browser allows to access to all defined symbols of the connected STEP 7 project.

When using the SIMATIC CFC editor on the same PC, the signals and connectors that are to be measured can be configured via drag & drop from the control program.

For recording data from SIMATIC S7-CPU with *ibaPDA*, there are different hardware devices of iba AG available.

Via PROFIBUS

- *ibaBM-DP* or the predecessor *ibaBM-DPM-S*
- *ibaCom-L2B* PCI card

Via PROFINET

- *ibaBM-PN*

Request blocks for *ibaPDA-Request-S7-DP/PN* are available for the following system configuration:

SIMATIC STEP 7 V5.x (SIMATIC Manager)				
SIMATIC S7 CPU	ibaBM-DP	ibaBM-DPM-S	ibaCom-L2B	ibaBM-PN
S7-300 integrated DP interface	X	X	X	
S7-300 CP342-5 (PROFIBUS)			X	
S7-400 integrated DP interface and CP443-5	X	X	X	
S7-400 integrated DP interface and CP443-5 (PROFIBUS)	X	X		
S7-300 integrated PN interface				X

SIMATIC STEP 7 V5.x (SIMATIC Manager)				
SIMATIC S7 CPU	ibaBM-DP	ibaBM-DPM-S	ibaCom-L2B	ibaBM-PN
S7-400 integrated PN interface and CP443-1 (PROFINET)				X

Table 2: Available Request blocks SIMATIC Manager

SIMATIC STEP 7 V1x Professional (TIA Portal)			
SIMATIC S7 CPU	ibaBM-DP	ibaBM-DPM-S	ibaBM-PN
S7-300 integrated DP interface	X	X	
S7-300 integrated PN interface			X
S7-400 integrated DP interface and CP443-5	X	X	
S7-400 integrated PN interface and CP443-1 (PROFINET)			X
S7-1500 integrated DP interface and CM1542-5 or CP1542-5 (PROFIBUS)	X	X	
S7-1500 integrated PN interface and CM1542-1 (PROFINET)			X

Table 3: Available Request blocks SIMATIC TIA portal

Note

You find the Request blocks in the iba S7 library (see chapter [↗ Iba S7 library](#), page 139). Always use the current version of this library.

3.1.1 Overview

ibaPDA-Request-S7-DP/PN works with direct access to the S7 operands. The variables that are to be measured can be entered in the signal list either with their operand designations or with their symbolic names that are determined when configuring them in the data block, in the CFC chart or in the symbol table.

This manual is divided into two main parts:

- General part, valid for all communication channels (chapter [↗ ibaPDA-Request-S7-DP/PN](#), page 13)

- Device specific part with special information for each variant (chapter [↗ Request-S7 Variants](#), page 32)

Currently, the following devices are supported as communication channels of *ibaPDA-Request-S7-DP/PN*:

- *ibaCom-L2B-x-8* card, in the following called "L2B-Request"
- *ibaBM-DP*, in the following called "DP-Request", in standard mode as well as in redundancy mode. Here is also a compatibility mode available for replacing systems with *ibaCom-L2B* cards.
- *ibaBM-DPM-S* as predecessor of *ibaBM-DP* in standard mode as well as in redundancy mode.
- *ibaBM-PN*, in the following called „PN-Request“.

Other documentation



You find detailed information about the devices

- *ibaCom-L2B-x-8* card
- *ibaBM-DP*
- *ibaBM-DPM-S*
- *ibaBM-PN*

in the corresponding device manuals.

3.1.2 How does the symbolic Request work?

The S7-CPU generally works with operand addresses. This requires a mapping table between symbols and operands to be created to enable the *ibaPDA* user to select symbolic variables. When selecting a symbol, *ibaPDA* will simultaneously assign the suitable operand addresses.

The mapping table of the symbols to the operand addresses - the address book - is generated using the S7 address book generator that is integrated in *ibaPDA*.

One part of the address book (symbol table and data blocks) is generated by direct evaluation of the STEP 7 project. When using SIMATIC CFC, the SCL code is analyzed that is generated by the CFC Compiler. This SCL code is stored in the STEP 7 project in the "Sources" folder. One look at the SCL code shows that all temporary calculation results at the connectors of the function blocks are stored in data blocks. The address book generator uses this information to create the mapping between block connectors and data blocks.

The browser integrated in the I/O Manager of *ibaPDA* then uses the address book to select the signals.

3.2 Configuration and engineering SIMATIC S7

Independent of the used communication interface, generally the following configuration steps have to be carried out on the SIMATIC side:

- Hardware configuration:
Integration of the devices in the device configuration, HW Config or NetPro.
- Software configuration (STEP 7):
Integration of the Request blocks in the S7 program

For detailed information about this subject, see the respective paragraph in chapter [Request-S7 Variants](#), page 32.

3.3 Configuration and engineering ibaPDA

3.3.1 General interface settings

For the configuration of the device-specific settings, please see chapter [Request-S7 Variants](#), page 32.

3.3.2 General module settings

All modules have the following common setting options:

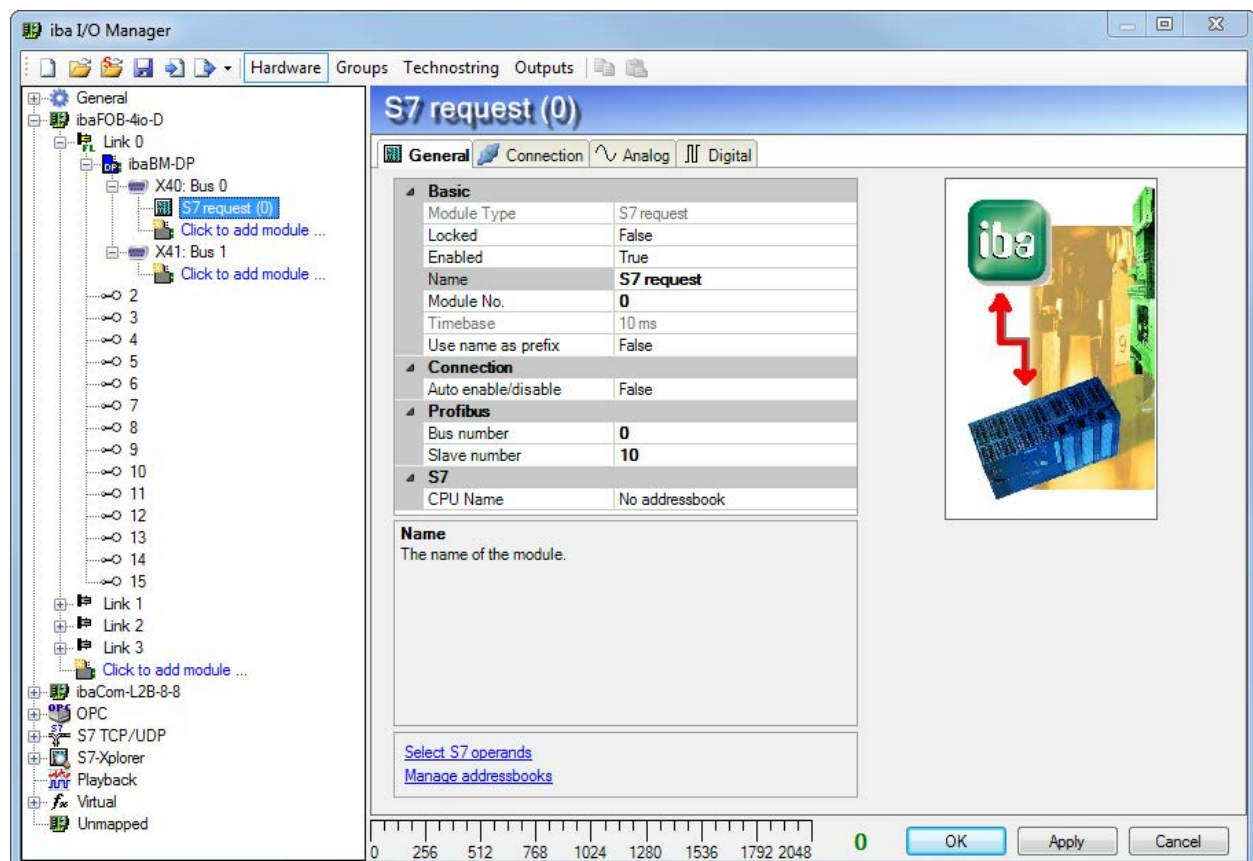


Fig. 1: General module settings

Basic settings

Module Type (information only)

Indicates the type of the current module.

Locked

A module can be locked to avoid unintentional or unauthorized changing of the module settings.

Enabled

Disabled modules are excluded from signal acquisition.

Name

The plain text name should be entered here as the module designation.

Module No.

Internal reference number of the module. This number determines the order of the modules in the signal tree of *ibaPDA* client and *ibaAnalyzer*.

Timebase

All signals of the module will be sampled on this time base.

Use name as prefix

Puts the module name in front of the signal names.

S7

CPU Name

Select the S7-CPU that is connected to this module. When selecting a S7-CPU, you can select the signals symbolically. Otherwise, the signals are selected via the S7 operand.

This requires that address books have already been generated. Otherwise, the selection list is empty. By clicking on „Create address book...” in the selection list, you get directly to the address book generator (see chapter ↗ *Selection via the operand symbols*, page 20).

Note



General settings that are not described here are module-specific settings. These settings are described in chapter ↗ *Request-S7 Variants*, page 32 under the respective module version.

3.3.3 Signal configuration

The signals to be measured are selected in the I/O Manager, either using the absolute operand address or the symbolic name provided by the *ibaPDA* address book browser.

When using the SIMATIC CFC editor on the same PC, the signals to be measured and the connectors can be configured from the PLC program via drag & drop in *ibaPDA*.

There are 3 options for selecting measurement values:

1. Selection via the absolute address of the S7 operands.
2. Selection via the S7 symbol addresses (symbol table and symbols from data blocks)
3. Selection of the CFC connectors (when programming the CPU with SIMATIC CFC)

SIMATIC CPU	Access via absolute address	Access via symbol	Access via CFC connectors ¹⁾
S7-300	X	X	X
S7-400	X	X	X
WinAC	X	X	X
S7-1500	X	X	

Table 4: Possible ways of access

Supported operand ranges:

Operand ranges	SIMATIC CPUs S7-300/400	SIMATIC CPUs S7-1500
Inputs (I)	X	X
Peripheral inputs (PI)	X	
Outputs (Q)	X	X
Markers (M)	X	X
Data blocks (DB)	X	X ²⁾

Table 5: Supported operand ranges

¹⁾ Prerequisite is the use of the SIMATIC STEP 7 options package S7-CFC

²⁾ The access to optimized data blocks of S7-1500 controllers is not supported.

3.3.3.1 Selection via the absolute address of the operands

There are 2 ways for selecting the measurement values via the operand address.

1. On the one hand, you can click on the *General* tab of the module on the "Select S7 operands" link in order to open the S7 operand editor.

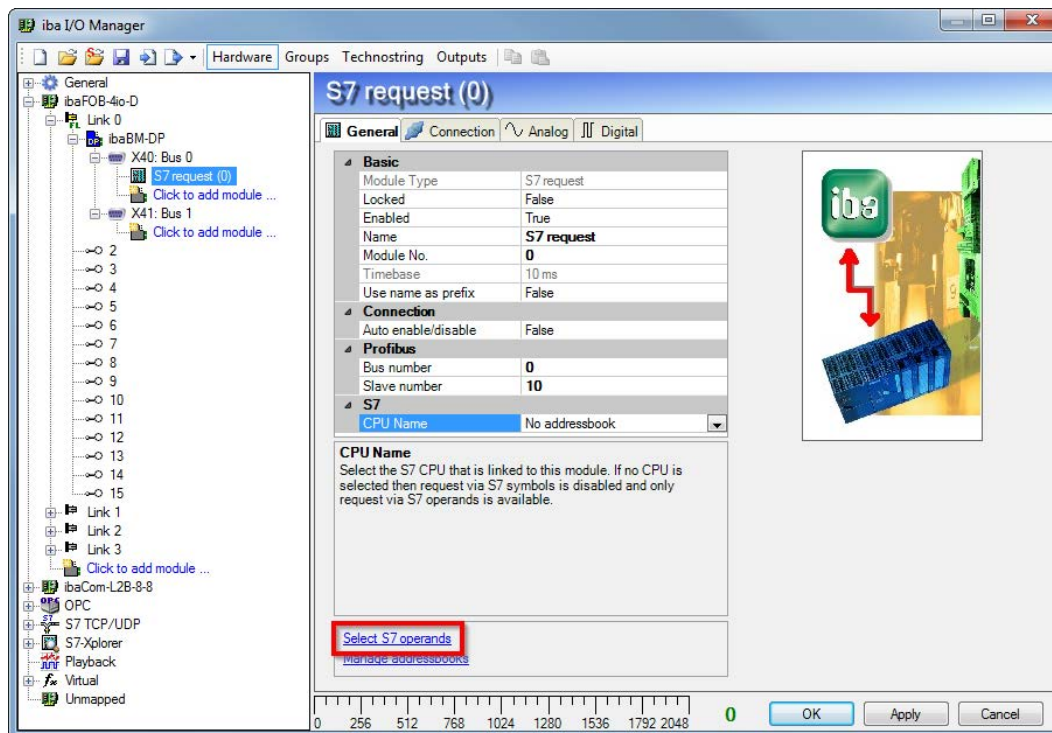


Fig. 2: Selecting S7 operands

2. On the other hand, you can click on the *Analog* or *Digital* tab of the module in one field of the "S7 Operand" column.

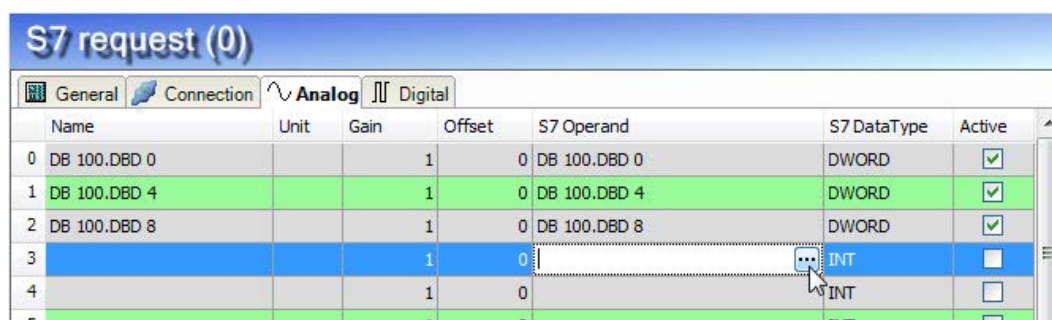



Fig. 3: Selecting S7 operand

In both cases, a dialog box is opened  in which you can select the S7 operand to be measured.

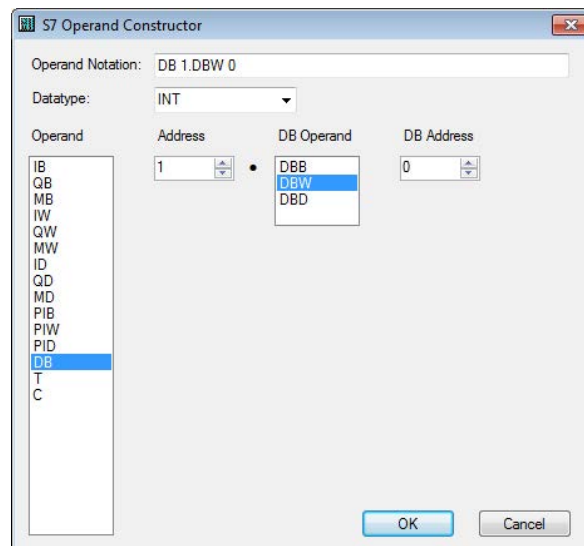


Fig. 4: S7 operand constructor

After you have set the desired operand address, click <OK> to exit the dialog.

Thereafter, you can enter the signal name in the *Name* column.

	Name	Unit	Gain	Offset	S7 Operand	S7 DataType	Active
0	temperature			1	0 DB 100.DBD 0	DWORD	<input checked="" type="checkbox"/>
1	DB 100.DBD 4			1	0 DB 100.DBD 4	DWORD	<input checked="" type="checkbox"/>
2	DB 100.DBD 8			1	0 DB 100.DBD 8	DWORD	<input checked="" type="checkbox"/>
3				1	0	INT	<input type="checkbox"/>
4				1	0	INT	<input type="checkbox"/>

Fig. 5: Operand names

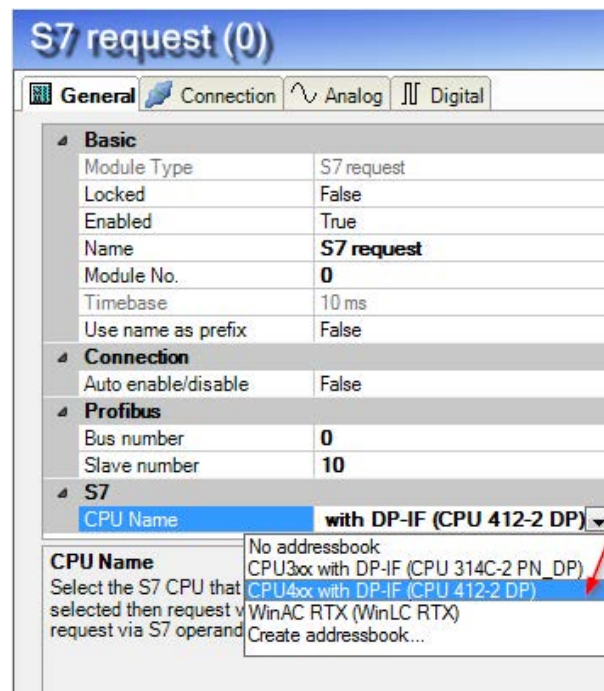
3.3.3.2 Selection via the operand symbols

This method of access requires that the signals to be measured already have an entry in the S7 symbol table, the PLC variable list or in a data block and that an address book has been created (see chapter [➤ Address books](#), page 27).

An advantage of this way of access is that the symbol addresses are applied automatically in *ibaPDA* as signal names.

You can integrate an address book into the module as follows:

1. Select the S7-CPU you want to assign this module to in the drop down menu on the *General* tab of the module under "CPU Name".



2. An additional "S7 symbol" column is displayed on the *Analog* and *Digital* tab.

Name	Unit	Gain	Offset	S7 Symbol	S7 Operand	S7 DataType	Active
0		1	0			INT	<input type="checkbox"/>
1		1	0			INT	<input type="checkbox"/>
2		1	0			INT	<input type="checkbox"/>
3		1	0			INT	<input type="checkbox"/>

Lookup symbols in the signal grid

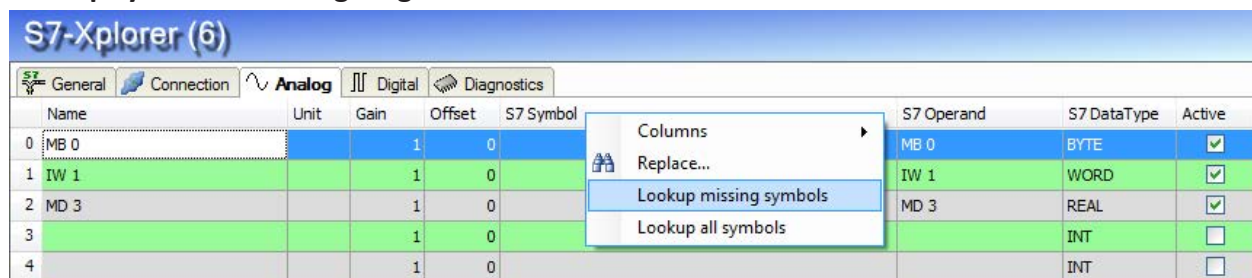


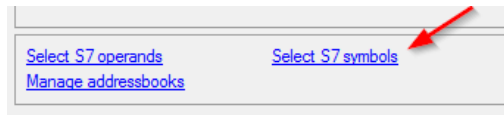
Fig. 6: Lookup symbols

ibaPDA can search for the symbol that corresponds to an operand. Right-click on the header of the signal grid and select either *Lookup missing symbols*, then only the missing symbols are searched for. Or select *Lookup all symbols*, then all symbols will be searched and replaced. The command performs a reverse resolution of the S7 symbols from the S7 operands. *ibaPDA* first searches the symbol table, then CFC, and finally the DBs for the operands.

3.3.3.2.1 Selection via Symbol Browser

Now, you can access the symbol addresses via the S7 CFC and symbol browser (short: symbol browser). The symbol browser can be opened in different ways:

1. Click on the link „Select S7 symbols" on the *General* tab. The selected signals are entered automatically in the right table *Analog* or *Digital*. The symbol browser stays open until it is closed with <OK>. This way, several signals can be added successively.



2. On the *Analog* or *Digital* tabs click in a cell of the "S7 Symbol" column. You can only select the symbols with the data type matching the table. After every selection, the symbol browser is closed.

General		Connection		Analog		Digital				
Name	Unit	Gain	Offset	S7 Symbol	S7 Operand	S7 DataType	Active			
0 ICosine		1	0	SYMBOL\ICosine	MW 44	INT	<input checked="" type="checkbox"/>			
1 ITriangle		1	0	SYMBOL\ITriangle	MW 40	INT	<input checked="" type="checkbox"/>			
2 RTriangle		1	0	SYMBOL\RTriangle	MD 36	REAL	<input checked="" type="checkbox"/>			
3 System clock byte		1	0	DBS\DB_TestData\SystemClockByte	DB 20.DBB 0	BYTE	<input checked="" type="checkbox"/>			
4 Sawtooth signal step width 1		1	0	DBS\DB_TestData\SawTooth_Step1	DB 20.DBW 2	INT	<input checked="" type="checkbox"/>			
5 Sawtooth signal step width 100		1	0	DBS\DB_TestData\SawTooth_Step100	DB 20.DBW 6	INT	<input checked="" type="checkbox"/>			
6 Sinus frequency 2 Hz		1	0	DBS\DB_TestData\Sinus_2Hz	DB 20.DBD 20	REAL	<input checked="" type="checkbox"/>			
7 Cosinus frequency 2 Hz		1	0	DBS\DB_TestData\Cosinus_2Hz	DB 20.DBD 28	REAL	<input checked="" type="checkbox"/>			
8		1	0			INT	<input type="checkbox"/>			

In the symbol browser, you have the following options:

- CFC variables:
On the *CFC* tab, you can select the configured CFC variables – consisting of the configured names of chart, block and connector.
- DB variables:
On the *DB* tab, the individual data blocks and their variables are displayed.
- Symbol table:
On the *Symbols* tab, the entries of the S7 symbol table can be selected.
- Search tab:
You can search the variable via a part of the name.

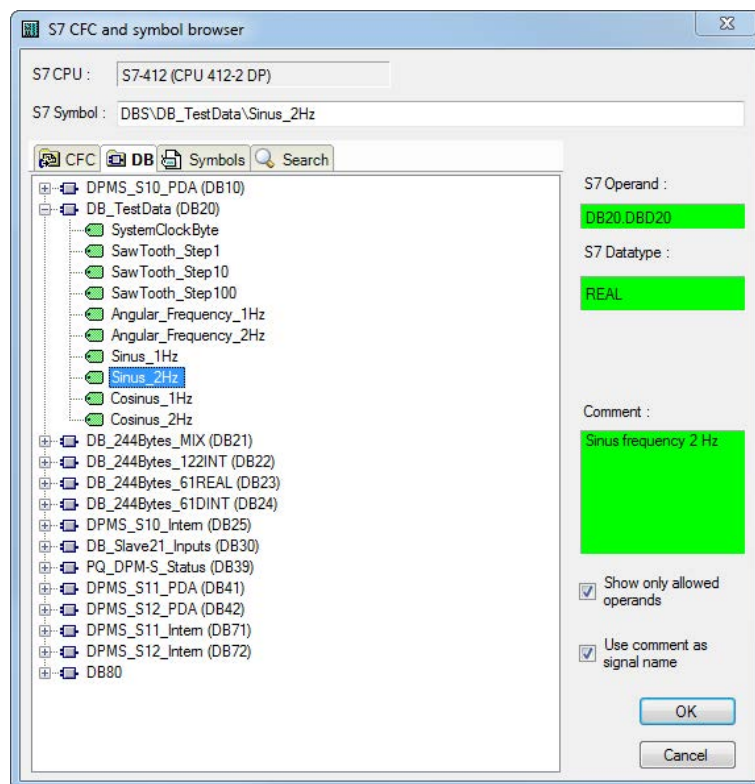


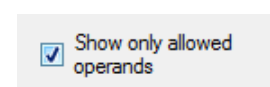
Fig. 7: S7-CFC and symbol browser

After having selected the variable, the assigned operand address, the data type and signal comment are displayed on the right side.

The variables are shown in the following colors:

Green	The operand is valid and can be transferred to the module setting with <OK>.
Yellow	The operand has a data type that does not match the selected column, e. g. in case you have selected a boolean variable as analog value or an integer value as digital value.
Red	The operand has a data type that is not supported by ibaPDA (e.g. STRING), or the operand is a constant.

You can hide all non-valid variables, by clicking on the option "Show only allowed operands".



Usually, the symbolic signal name from STEP 7 is adopted as signal name in the I/O Manager. Optionally, you can adopt the STEP 7 signal comment as signal name in the I/O Manager by selecting the option "Use comment as signal name".



3.3.3.3 Measuring the CFC connectors

For selecting CFC connectors for the measurement process, proceed basically as described in chapter 7 *Selection via the operand symbols*, page 20 until opening the S7-CFC and symbol browser.

In the S7-CFC and symbol browser open the CFC tab and select the signals here. The connectors are listed hierarchically by chart name, module name and connector name:

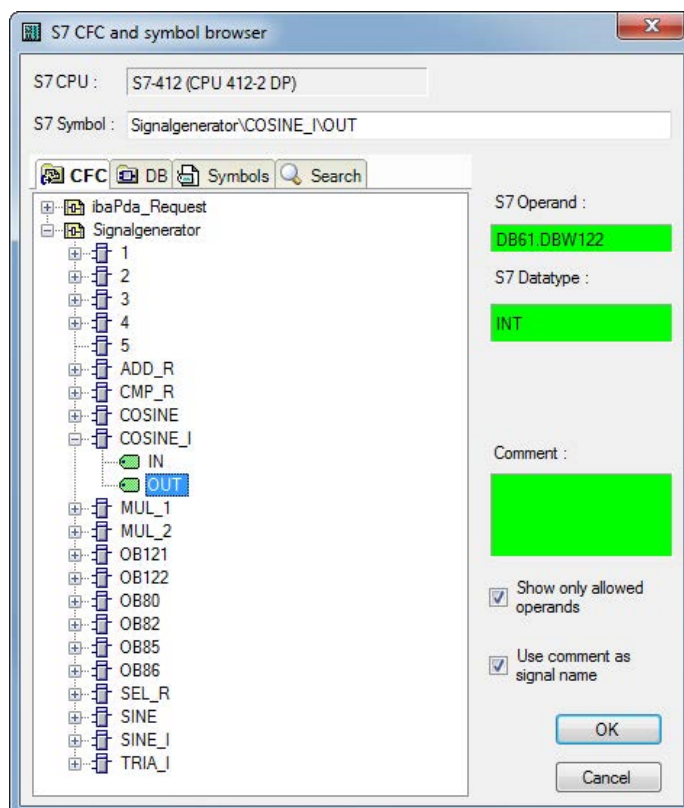
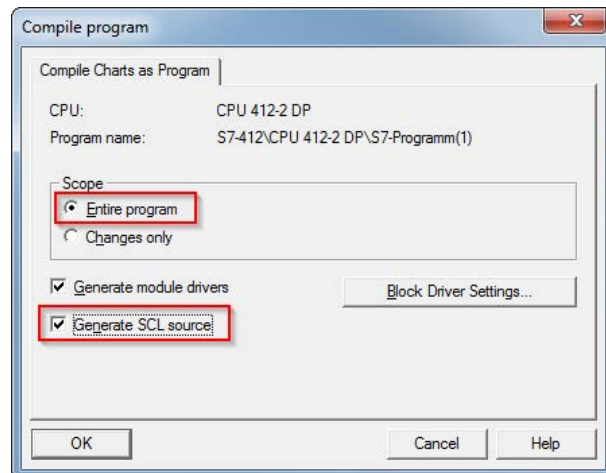


Fig. 8: S7-CFC and symbol browser

Note

In case no connectors are displayed on the CFC tab, then possibly the SCL sources have not been translated in the STEP 7 project.

To solve this problem, make sure that the following options are selected in the dialog box for compiling the program in the SIMATIC software:



Create the address books again.

Note

When compiling a CFC program, generated DB addresses are assigned automatically to the connectors in STEP 7. Depending on the scope of program changes adopted between two compilation runs, connectors may have assigned different DB addresses.

In that case, the address book for *ibaPDA* must be newly generated, too. The symbolically configured signals are checked automatically in the I/O Manager and the associated absolute S7 operands are updated.

Special function Drag & Drop

A convenient way to select signals is to drag & drop them from the CFC chart into the I/O Manager of *ibaPDA*.

For this purpose, display the CFC editor on the same PC in the background as full screen and the *ibaPDA* client in the window mode in the foreground. Then, start the I/O Manager.

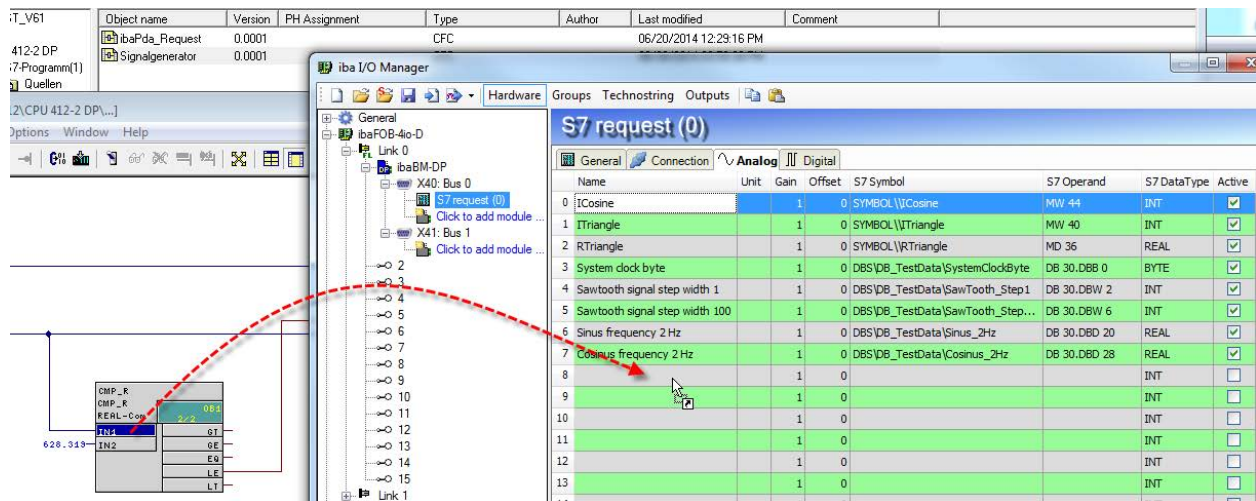


Fig. 9: Selecting signals via drag & drop

Now, click the connector in the CFC chart, hold the mouse key down, drag it onto the I/O Manager and drop it in the desired signal row. The CFC connector is then entered in the signal table of the request module.

Note



For the drag & drop mechanism to work correctly, the address book belonging to the CPU has to be generated before and assigned to the request module.

During the drag & drop procedure, the associated absolute S7 operand is searched based on the selected connector (described by chart, block and connector name) from the address book and inserted into the signal row.

3.3.4 Address books

The address books for SIMATIC S7 controllers are created and managed across modules. One address book can be used in more than one module.

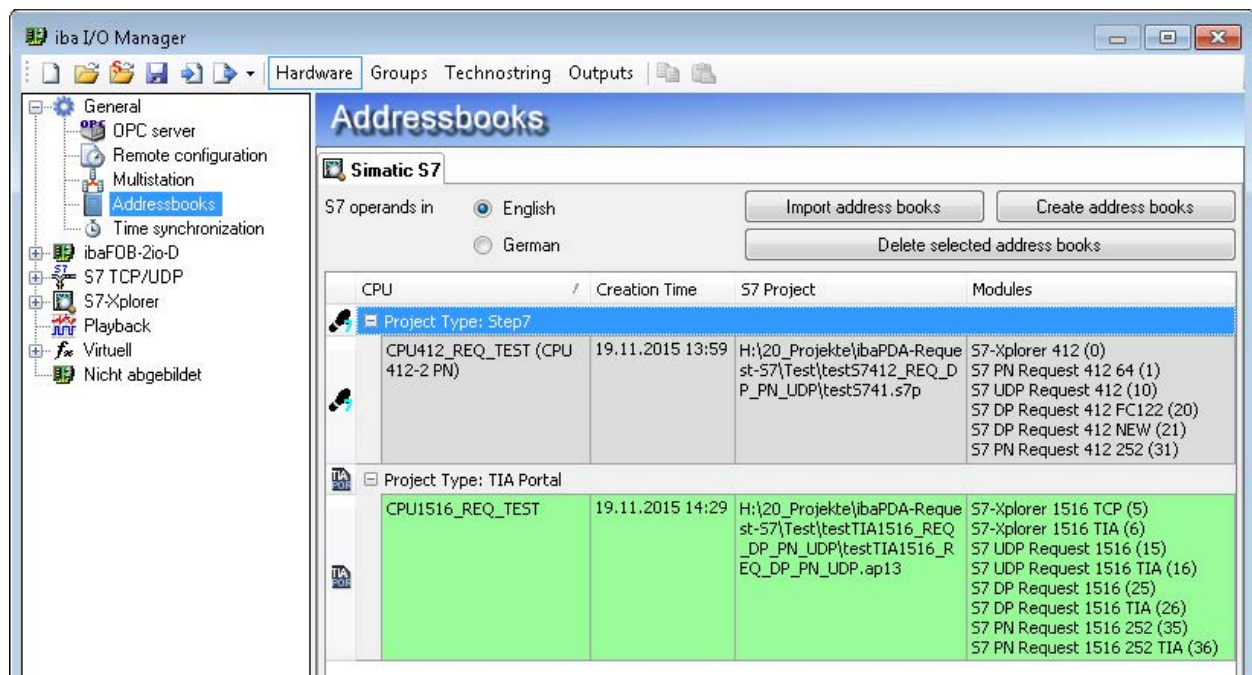


Fig. 10: S7 address book

There are different types of address books for the different S7 project types:

- STEP 7: SIMATIC Manager project
(not for modules with connection mode TCP/IP S7-1x00)
- TIA Portal: TIA Portal Project

S7 operands in English / German

Selection of the signal mnemonics

Import address books

Import of address books which have already been created

Create address books

Creates new address books from STEP 7 projects

Delete selected address books

Deletes address books from a table

Table

List of all address books currently available in the system with name, creation date, storage path of the STEP 7 project or IP address of the CPU for address books read online and location of use of the address book.

Creating an address book

An S7 address book can either be created offline from an S7 project (both STEP 7 CLASSIC and TIA-Portal) or online directly from a CPU (S7-1200 / 1500 only).

Offline from S7 project

For creating an address book, the S7 project has to be available. This is not necessary for subsequent use.

Open the S7 address book generator with the <Create address books> button. Optionally, you can also open it under "S7 - CPU Name" on the *General* tab in the module configuration. Select "Create addressbook..." in the dropdown menu.

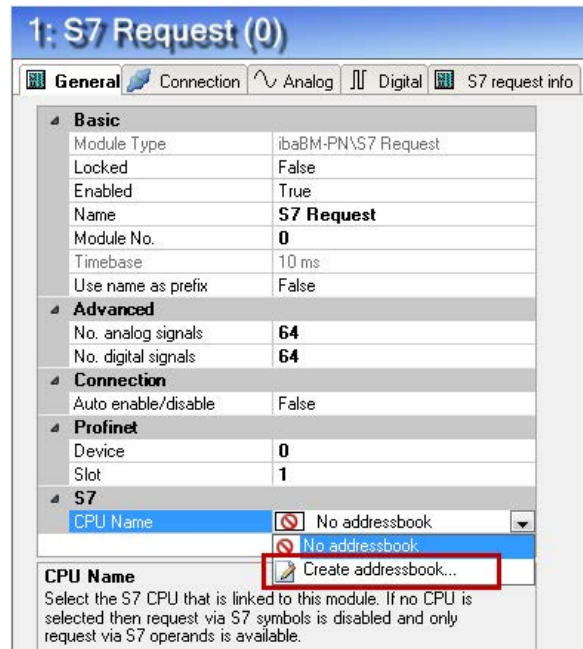
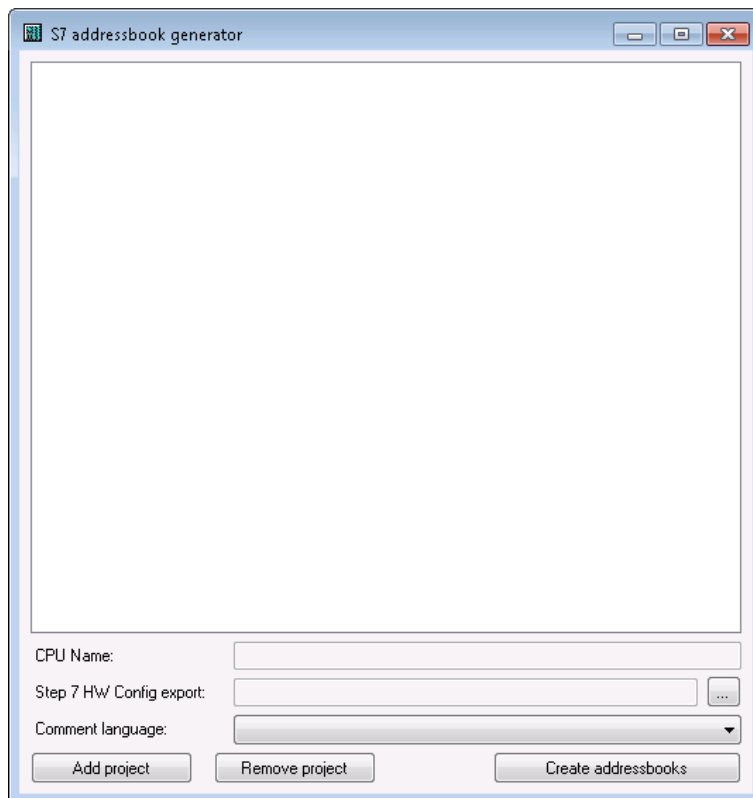


Fig. 11: Create S7 address book from module configuration

Procedure in the S7 address book generator:

1. Click on the button <Add project>

**CPU Name**

CPU designation

Step 7 HW Config export

optional selection of a HW Config export file (useful when using an iba busmonitor in sniffing mode)

Comment language

Imported comment language can be selected (only available in SIMATIC TIA Portal projects)

Add project

Adds a new project to the list

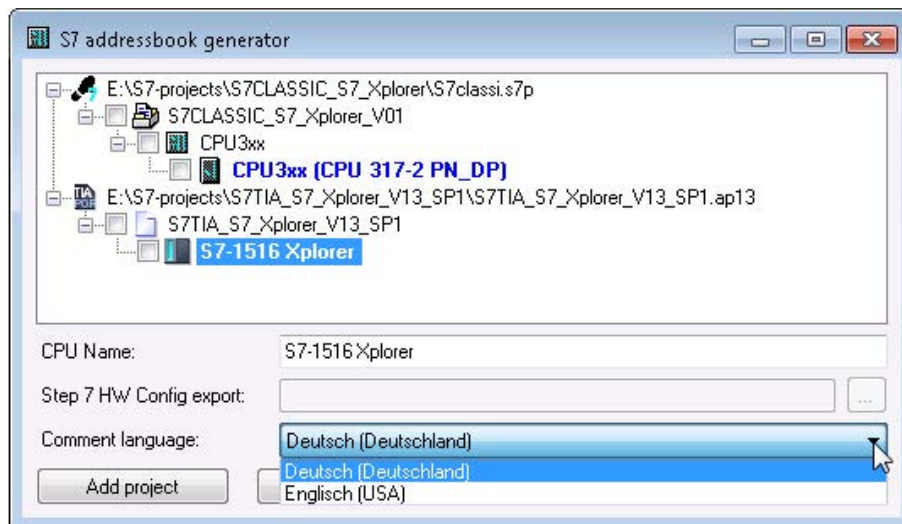
Remove project

Removes the marked project from the list

Create address books

Creates address books from the selected projects

2. Select a project file in the file browser
3. Now, the STEP 7 project with all configured CPUs is displayed. Mark the CPUs you want to create the address books from and click on the button <Create addressbooks>.



4. Now, you can select the created address book in the selection list.

Note



The entry in the "CPU Name" field can be overwritten. Thus, you can assign a unique name for the CPU that differs from that in the STEP 7 project. This is especially interesting, when you use several STEP 7 projects in which the CPUs have the same name.

Note



TIA Portal projects must be compiled, saved and closed, when address books of these projects are generated.

Online from S7-1200 / 1500 CPU

Online address books can be created out of an S7-1200 or S7-1500 CPU if connection mode TCP/IP S7-1x00 has been selected. The address data are read directly from the CPU. Accessing the S7 project is not necessary.

Click on button <Load addressbook from S7> to load the address book.

The CPU name of the address book will be given automatically.

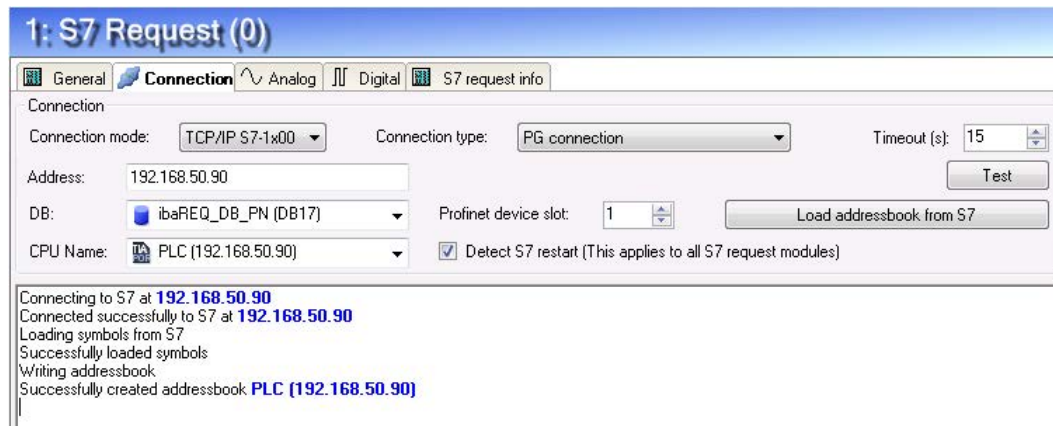


Fig. 12: Load address book online from CPU

Address books which have been created online do also contain operand type address information and hence can be used in connection mode TCP/IP as well. Change the connection mode after creation of the address book.

4 Request-S7 Variants

4.1 Request-S7 for ibaBM-PN

In the following, we describe the Request-S7 version for the PROFINET bus module *ibaBM-PN*.

4.1.1 General information

Request-S7 for *ibaBM-PN* can be configured in the following system configurations:

SIMATIC S7 CPU	SIMATIC STEP 7 V5.x (SIMATIC Manager)	SIMATIC STEP 7 V1x Professional (TIA Portal)
S7-300 integrated PN interface	X	X
S7-400 integrated PN interface and CP443-1	X	X
S7-1500 integrated PN interface and CM1542-1		X

Table 6: Request-S7 for ibaBM-PN system configuration

With the PN Request, the measurement values (Request Handshake) are requested via a separate TCP/IP connection.

Depending on the hardware and software, different access points can be selected for the Request:

- TCP/IP: the connection to the SIMATIC S7 is established over an integrated PN interface of the S7-CPU or the respective CP modules in the PLC and the standard network interface of the PC. No additional Siemens software is required for the connection.
- PC/CP: this is the designation for different SIMATIC specific access points. In contrast to the TCP/IP connection, the SIMATIC communication software (and the corresponding licenses) must be installed on the PC for all connection types within the PC/CP group.
 - MPI, PROFIBUS: The connection to the SIMATIC S7 is established via the MPI or PROFIBUS interface of the PC; e.g. with the PCI card CP5611 or the MPI adapter for USB or serial PC interface.
 - TCP/IP, ISO: here, either the standard network interface of the PC or a suitable interface card is used for the connection to the S7.
- TCP/IP S7-1x00: The connection to a SIMATIC S7-1500 is established via an integrated PN interface of the S7-CPU or respective CP modules in the PLC and the standard network interface of the PC. No additional Siemens software is required for the connection.

System integration with ibaBM-PN

The measured data are transmitted to the *ibaBM-PN* device over PROFINET.

You need the following connections:

- Online connection between *ibaPDA* and S7-CPU (TCP/IP, MPI or DP)
- Fiber optic connection between *ibaPDA/ibaFOB-io-D* and *ibaBM-PN*
- PROFINET connection between *ibaBM-PN* and S7-CPU

The bus monitor offers two separate PROFINET devices. A transfer rate of max. 1440 Byte is possible for each device.

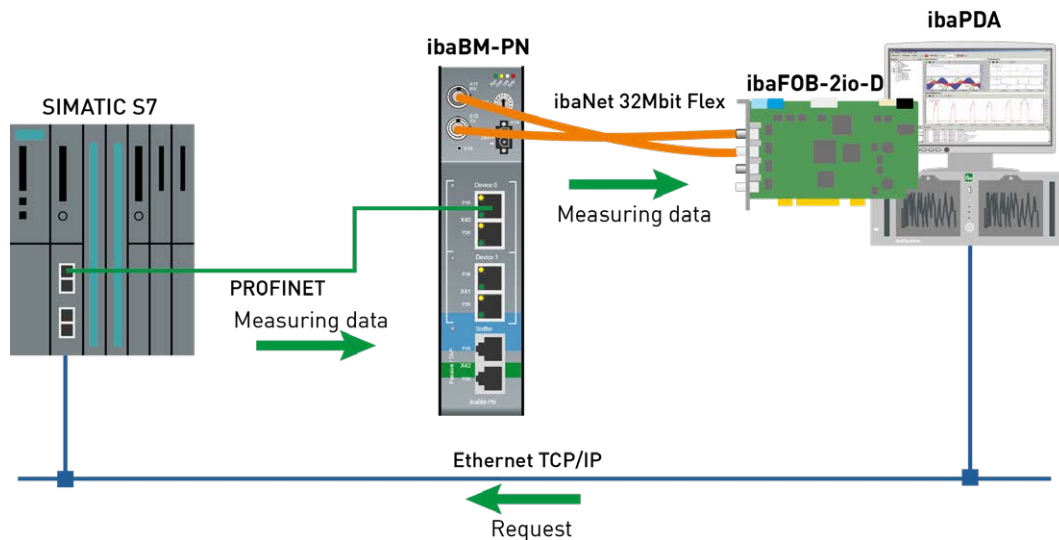


Fig. 13: Request-S7 with ibaBM-PN

Other documentation



For detailed information about *ibaBM-PN*, please see the device manual.

For information about application examples, see chapter ➤ *Application examples*, page 148.

4.1.2 Configuration and engineering SIMATIC S7-300, S7-400 and WinAC

In the following, we describe the configuration and engineering on the SIMATIC S7 side with SIMATIC Manager (STEP 7 Version ≤ V5).

On the SIMATIC side, generally the following configuration steps have to be carried out:

- Hardware configuration
Integration of the PROFINET device in the hardware configuration
- Software configuration (STEP 7 V5):
Integration of the Request blocks in the S7 program

4.1.2.1 Description of the Request blocks

The communication between the S7 and ibaPDA is initialized and controlled with these blocks.

One set of Request blocks has to be called for each Request module (connection) in *ibaPDA*. The used blocks are part of the iba S7 library (see chapter [Iba S7 library](#), page 139).

Depending on the system configuration, different Request block combinations have to be used:

Request block	CPU with integrated PN interface or WinAC RTX	S7-400 CPU + CP443-1	recommended call up level
ibaREQ_M (FB140)	X	X	OB1
ibaREQ_PN (FB141)	X	X	OB1 or OB3x
ibaREQ_PNdev (FB150)	X	X	OB1 or OB3x
ibaREQ_DB (DB15)	X	X	-

Table 7: Request blocks

The following blocks always have to be used:

- **ibaREQ_M (Management)**
The block realizes the communication with *ibaPDA*. Ideally, the block is called in the OB1. This block has to be called separately in every system configuration for each module in *ibaPDA*.
- **ibaREQ_PN (provides the actual signal values)**
The block provides the actual signal values in the call up cycle and sends the provided values via PROFINET.
- **ibaREQ-PNdev (diagnostics for PROFINET device)**
The block diagnoses cyclically the availability of the used PROFINET device.
- **ibaREQ_DB (interface DB)**
This DB serves as interface to *ibaPDA* and between the different Request blocks.

4.1.2.1.1 ibaREQ_M (FB140)

Description of the formal parameters:

Name	Type	Data Type	Description
REQ_DB	IN	BLOCK_DB	DB of the ibaPDA communication interface ibaREQ_DB
RESET	IN	BOOL	FALSE -> do no perform reset (Standard) TRUE -> perform reset

Name	Type	Data Type	Description
INP_RANGE	IN	INT	Number of input bytes (evaluation only during initialization), 0:automatic detection (recommended)
OUT_RANGE	IN	INT	Number of output bytes (evaluation only during initialization), 0:automatic detection (recommended)
MARKER_RANGE	IN	INT	Number of marker bytes (evaluation only during initialization), 0:automatic detection (recommended)
ERROR_STATUS	OUT	DWORD	Internal error code

The following SFCs are used internally:

- SFC 20 (BLKMOV)
- SFC 21 (FILL)
- SFC 24 (TEST_DB)
- SFC 51 (RDSYSST)

Detailed description:

REQ_DB

The DB is used for data exchange with *ibaPDA*. For all Request blocks that belong together, the identical DB has to be configured.

Length:

5280 Bytes : up to 128 Pointers (min.)

9120 Bytes : up to 512 Pointers

14240 Bytes : up to 1024 Pointers (max.)

Any other length within these limits is permitted. The number of usable pointers will be evaluated according to the length.

RESET

Serves for manually resetting the Request blocks. All Request blocks of a combination are automatically reset together. Usually, the input does not have to be connected

INP_RANGE

Limits the number of input bytes to be measured.

If INP_RANGE = 0, the size of the available processor image of the inputs is determined by the Request FB itself (recommended). Evaluation is done only during the initialization phase of the module.

OUT_RANGE

Limits the number of output bytes to be measured.

With OUT_RANGE = 0, the size of the available processor image of the outputs is determined by the Request FB itself (recommended). Evaluation is done only during the initialization phase of the module.

MARKER_RANGE

Limits the number of marker bytes to be measured.

With MARKER_RANGE = 0, the number of available markers is determined by the Request FB itself (recommended). Evaluation is done only during the initialization phase of the module.

ERROR_STATUS

Internal error of the block. If there is no error, the value 0 will be issued.

For a list of all possible error codes, see chapter [➤ Error codes of Request blocks](#), page 168.

4.1.2.1.2 ibaREQ_PN (FB141)**Description of the formal parameters:**

Name	Type	Data Type	Description
REQ_DB	IN	BLOCK_DB	DB of the ibaPDA communication interface ibaREQ_DB
ADR_SLOT	IN	INT	Start address of the peripheral output range
DEVICE_STATUS	OUT	INT	Status of the PROFINET device
ERROR_STATUS	OUT	INT	Internal error code

The following SFCs are used internally:

- SFC 15 (DPWR_DAT)
- SFC 20 (BLKMOV)
- SFC 21 (FILL)
- SFC 50 (RD_LGADR)
- SFC 71 (LOG_GEO)

Detailed description:**REQ_DB**

The DB is used for data exchange with *ibaPDA*. For all Request blocks that belong together, the identical DB has to be configured.

ADDR_SLOT

Peripheral start address of the used slot in ibaBM-PN in the peripheral output range for sending data.

DEVICE STATUS

Status of the addressed PROFINET device in the ibaBM-PN.
(0: Device not accessible / error, 1: Device accessible)

ERROR_STATUS

Internal error of the block. If there is no error, the value 0 will be issued.

For a list of all possible error codes, see chapter [➤ Error codes of Request blocks](#), page 168.

4.1.2.1.3 ibaREQ_PNdev (FB150)

Description of the formal parameters:

Name	Type	Data Type	Description
ADR_SLOT	IN	INT	Start address of the peripheral output range
DEVICE_STATUS	OUT	INT	Status of the PROFINET device
ERROR_STATUS	OUT	INT	Internal error code

The following SFCs are used internally:

- SFC 51 (RDSYSST)
- SFC 71 (LOG_GEO)

Detailed description:

ADDR_SLOT

Peripheral start address of the used slot in ibaBM-PN in the peripheral output range for sending data.

DEVICE STATUS

Status of the addressed PROFINET device in the ibaBM-PN.
(0: Device not accessible / error, 1: Device accessible)

ERROR_STATUS

Internal error of the block. If there is no error, the value 0 will be issued.

For a list of all possible error codes, see chapter [➤ Error codes of Request blocks](#), page 168.

4.1.2.2 Hardware configuration

For each PROFINET device, a separate PROFINET device has to be configured.

The GSDML file "GSDML-Vx.yy-ibaBM-PN-yyyymmdd.xml" has to be used.

Note



You find the latest version of the GSDML file on the "iba Software & Manuals" DVD in the following directory:
 \02_iba_Hardware\ibaBM-PN\01_GSD_Files\

S7-CPU's provide consistent slots with a max. of 252 Bytes. You need one slot for each Request block. You can also use slots with a smaller size.

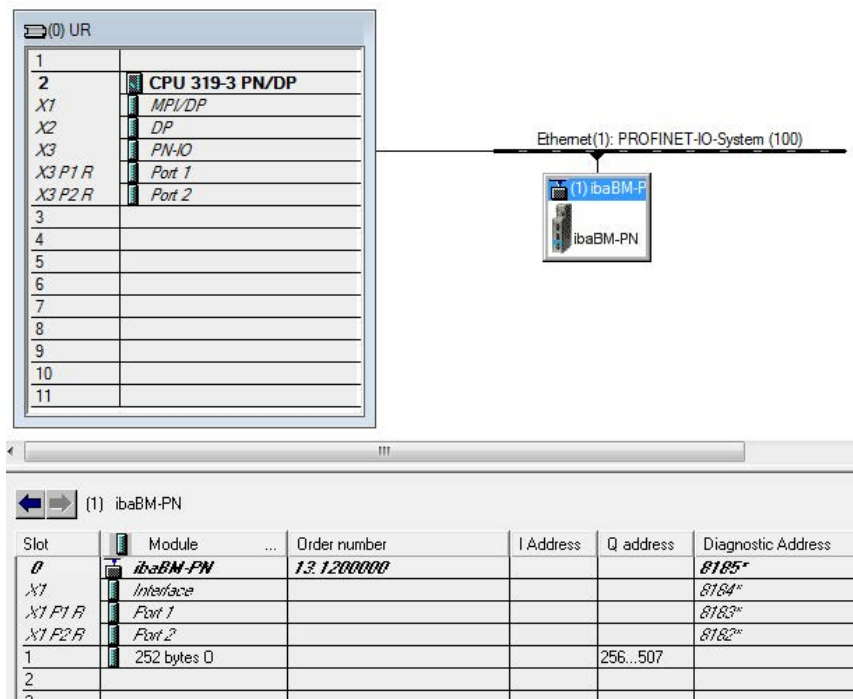


Fig. 14: Hardware configuration

4.1.2.3 Configuration in STEP 7

In the following, we describe how to configure the Request blocks in STEP 7 V5.

Request-S7 is intended for the use with CPUs S7-300/400 with integrated PN interface or CPUs S7-400 with external interface CP 443-1 (PROFINET controller).

Note



The use of the external PN interface CP343-1 of a CPU S7-300 is not supported!

Note



The request blocks do not support multi-instance calls.

Copy the required blocks from the iba S7 library (see chapter [Iba S7 library](#), page 139) to the blocks folder of your STEP 7 project:

Note

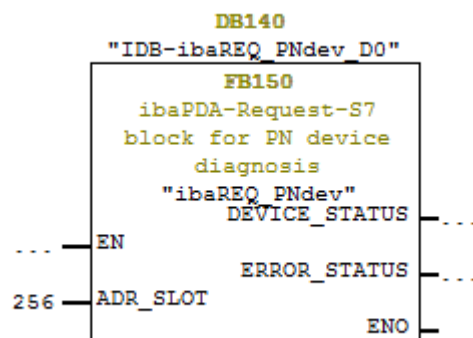
If the block numbers in your project are already occupied, assign new numbers to the blocks from the iba S7 library when copying.

The following blocks are required:

- ibaREQ_M (FB140)
- ibaREQ_PN (FB141)
- ibaREQ_PNdev (FB150)
- ibaREQ_DB (DB15)

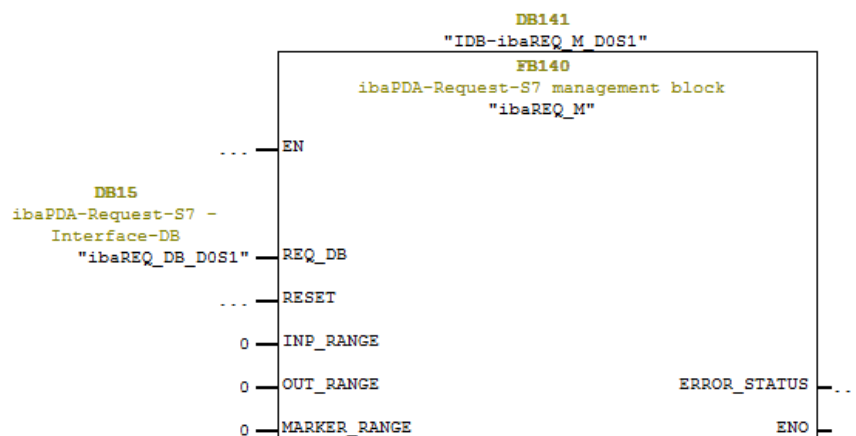
For each used PROFINET device of an ibaBM-PN, the following steps have to be carried out:

- ➡ Call the ibaREQ_PNdev (FB150) preferably within the OB1 or a cyclic interrupt (OB3x).

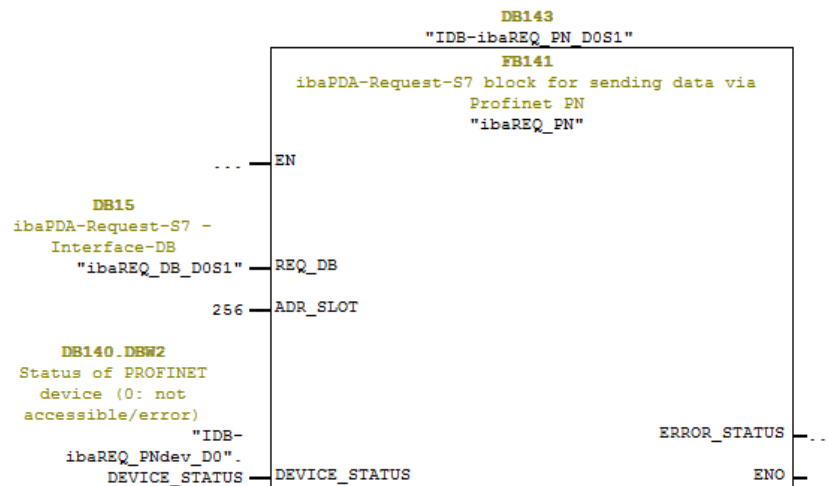


For each Request module, the following steps have to be carried out:

1. Call the ibaREQ_M (FB140) preferably within the OB1



2. Enable "Retain" for the complete instance block, you have just created.
3. Call the ibaREQ_PN (FB141), preferably within the OB1 or a cyclic interrupt OB (OB3x).



For each additional Request module:

- In the blocks folder, an ibaREQ_DB (DB15) has to be available for each Request module. Copy the data block and assign a new unique DB number.
- Within the OB1, the ibaREQ_M (FB140) has to be called once more for each Request module with a new DB number (input REQ_DB).
- Within the OB1 or a cyclic interrupt OB (OB3x), the ibaREQ_PN (FB141) has to be called once more for each Request module with the new DB number (Input REQ_DB).
- Please consider that all instance data blocks have to be unique and that the values for the ADR_SLOT are assigned uniquely.
- It is sufficient to call the ibaREQ_PNdev just once per used PROFINET device and not for each Request module.

Finish:

- Load all blocks into the S7-CPU and restart.

4.1.3 Configuration and engineering SIMATIC S7-1500

In the following, we describe the configuration and engineering on the SIMATIC S7 side with the SIMATIC TIA Portal.

Basically, the following configuration steps have to be carried out on the SIMATIC TIA Portal side:

- Network configuration
Integration of the PROFINET devices in the device configuration
- Configuration Software:
Integration of the Request blocks in the S7 program
- Device configuration:
Setting the CPU protection properties

4.1.3.1 Description of the Request blocks

The communication between the S7 and ibaPDA is initialized and controlled with these blocks. One set of Request blocks has to be called for each Request module (connection).

The used blocks are part of the iba S7 library (see chapter [Iba S7 library](#), page 139).

Request block	S7-1500 CPU with integrated PN interface	S7-1500 CPU with CM1542-1	recommended call up level
ibaREQ_M (FB1400)	X	X	OB1
ibaREQ_PN (FB1401)	X	X	OB1 or OB3x
ibaREQ_DB (DB15)	X	X	-
ibaREQ_DB-Interface	X	X	-

Table 8: Request blocks

- **ibaREQ_M (Management)**
The block realizes the communication with *ibaPDA*. Ideally, the block is called in the OB1.
- **ibaREQ_PN (provides and sends the actual signal values)**
The block provides the actual signal values in the send cycle. Ideally, the block is called in a cyclic interrupt OB.
- **ibaREQ_DB (interface DB)**
This DB serves as interface to *ibaPDA* and between the different Request blocks.

4.1.3.1.1 ibaREQ_M (FB1400)

Description of the formal parameters:

Name	Type	Data Type	Description
REQ_DB	IN	DB_ANY	DB of the ibaPDA communication interface ibaREQ_DB
RESET	IN	BOOL	TRUE -> perform reset
CPU_HW_ID	IN	HW_IO	Hardware ID of local CPU
ERROR_STATUS	OUT	WORD	Error code

The following SIMATIC standard blocks are used internally:

GET_IM_DATA (FB801)**Detailed description:****REQ_DB**

The DB is used for data exchange with *ibaPDA*. For all Request blocks that belong together, the identical DB has to be configured.

The length of the data block is fixed.

RESET

Serves for manually resetting the Request blocks. All Request blocks of a combination are automatically reset together. Usually, the parameter does not have to be connected

CPU_HW_ID

TIA Portal system constant, which refers to the corresponding CPU.

ERROR_STATUS

Internal error of the block. If there is no error, the value 0 will be issued.

For a list of all possible error codes, see chapter [➤ Error codes of Request blocks](#), page 168.

4.1.3.1.2 ibaREQ_PN (FB1401)**Description of the formal parameters:**

Name	Type	Data Type	Description
ADR_SLOT	IN	VARIANT	Start address of the output range
RESET_CON	IN	BOOL	TRUE -> reset of the communication connection
REQ_DB	INOUT	UDT	DB of the ibaPDA communication interface ibaREQ_DB
ERROR_STATUS	OUT	WORD	Internal error code

Detailed description:**ADR_SLOT**

Start address of the used slot in *ibaBM-PN* in the output range of the process image.

REQ_DB

The DB is used for data exchange with *ibaPDA*. For all Request blocks that belong together, the identical DB has to be configured.

ERROR_STATUS

Internal error of the block. If there is no error, the value 0 will be issued.

For a list of all possible error codes, see chapter [➤ Error codes of Request blocks](#), page 168.

4.1.3.2 Network configuration

For each PROFINET device, a separate PROFINET device has to be configured.

The GSDML file "GSDML-Vx.yy-ibaBM-PN-yyyyymmdd.xml" has to be used.

Note



You find the latest version of the GSDML file on the "iba Software & Manuals" DVD in the following directory:

\02_iba_Hardware\ibaBM-PN\01_GSD_Files\

S7-CPUs allow consistent slots with max. 252 Bytes. You need one slot for each Request block. You can also use slots with a smaller size.

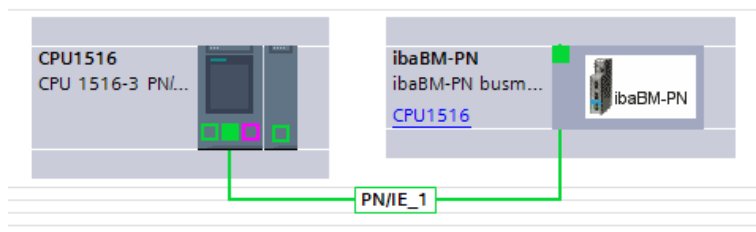


Fig. 15: Net view

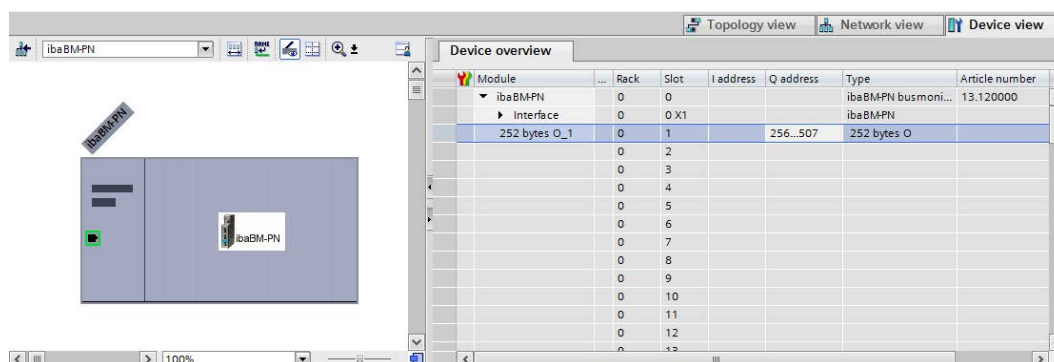


Fig. 16: Device view

4.1.3.3 Configuration in STEP 7

In the following, we describe how to configure the Request blocks in TIA Portal STEP 7.

Copy the required blocks from the iba S7 library (see chapter [Iba S7 library](#), page 139) to the blocks folder of your STEP 7 project:

Note

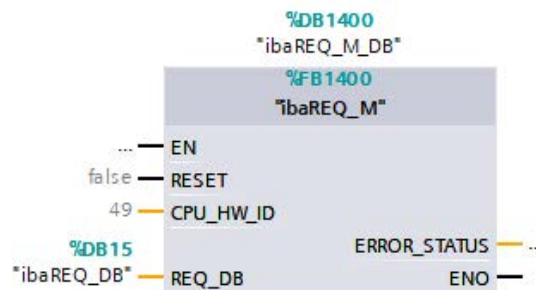
The request blocks do not support multi-instance calls.

The following blocks are required:

- ibaREQ_M (FB1400)
- ibaREQ_PN (FB1401)
- ibaREQ_DB (DB15)
- ibaREQ_DB-Interface (PLC data type)

For each Request module, the following steps have to be carried out:

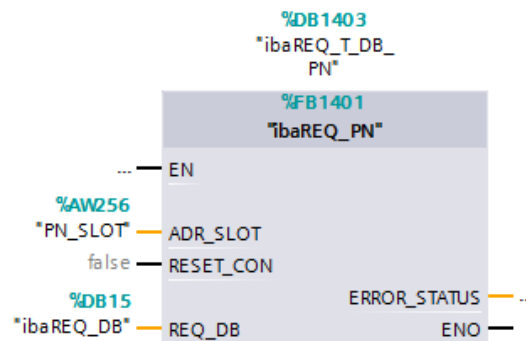
1. Call the ibaREQ_M (FB1400) preferably within the OB1



2. Enable "Retain" for the complete instance block, you have just created.

Name	Data type	Offset	Start value	Retain	Accessible f...	Visible in ...	Setpoint
1. Input							
2. RESET	Bool	0.0	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3. Output							
4. ERROR_STATUS	Word	2.0	16#0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5. InOut							
6. REQ_DB	*ibaREQ_DB-interfa...	4.0		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7. Static							
8. EXPERT	Struct	10.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9. sIDBinitialized	Bool	12.0	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
10. sOperandsInvalid	Bool	12.1	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
11. sIMDataValid	Bool	12.2	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
12. sOk	Bool	12.3	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
13. sGET_IM_DATA	Get_IM_Data			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14. sIM_DATA	Array[0..53] of Byte	14.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
15. sAdrOPList	Dint	68.0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
16. cBAddr_PDA257	Int	72.0	64	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
17. cBAddr_S72PDA	Int	74.0	1440	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
18. cBAddr_Xchange	Int	76.0	2464	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
19. cBAddr_Ops	Int	78.0	2528	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
20. cBAddr_actVal	Int	80.0	12768	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
21. cMaxDataLength	UInt	82.0	1472	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
22. sREQ_AnzOperanden...	Int	84.0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
23. sREQ_AnzOperanden...	Int	86.0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

3. Call the ibaREQ_PN (FB1401), preferably within the OB1 or a cyclic interrupt OB (OB3x).



For each additional Request module:

- In the blocks folder, an ibaREQ_DB (DB15) has to be available for each Request module. Copy the data block and assign a new unique DB number.
- Within the OB1, the ibaREQ_M (FB1400) has to be called once more for each Request module with the new DB number.
- Within the OB1 or within a cyclic interrupt (OB3x), the ibaREQ_PN (FB1401) has to be called for each Request module with the new DB number once more.
- Please consider that all instance data blocks have to be unique and that the values for the ADR_SLOT are assigned uniquely.

Finish:

- Load all blocks into the S7-CPU and restart.

4.1.3.4 Device configuration

The following settings have to be done in the device configuration of the CPU:

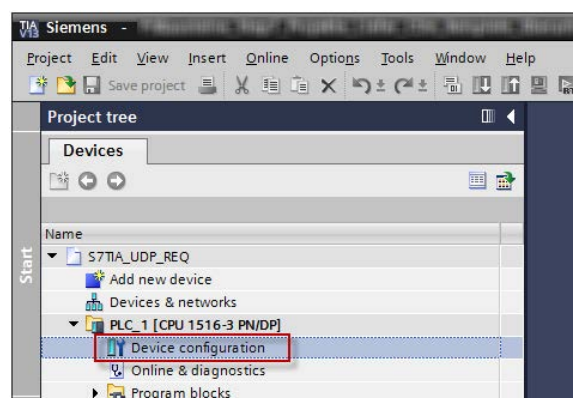
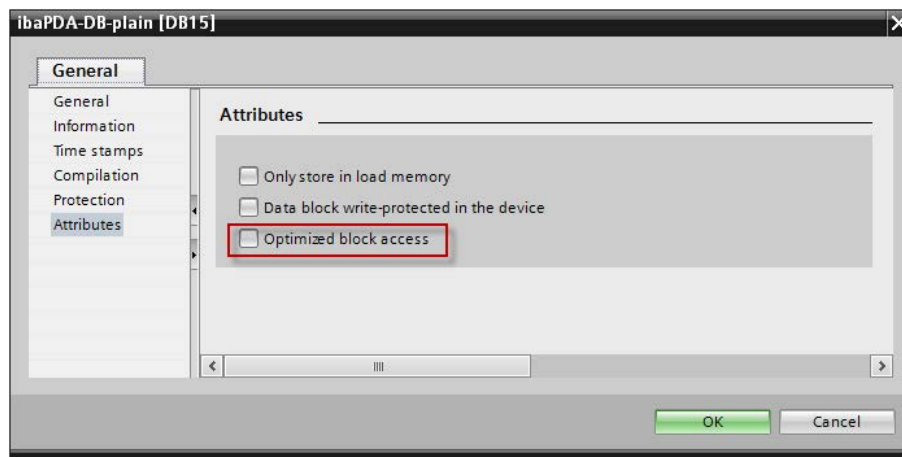


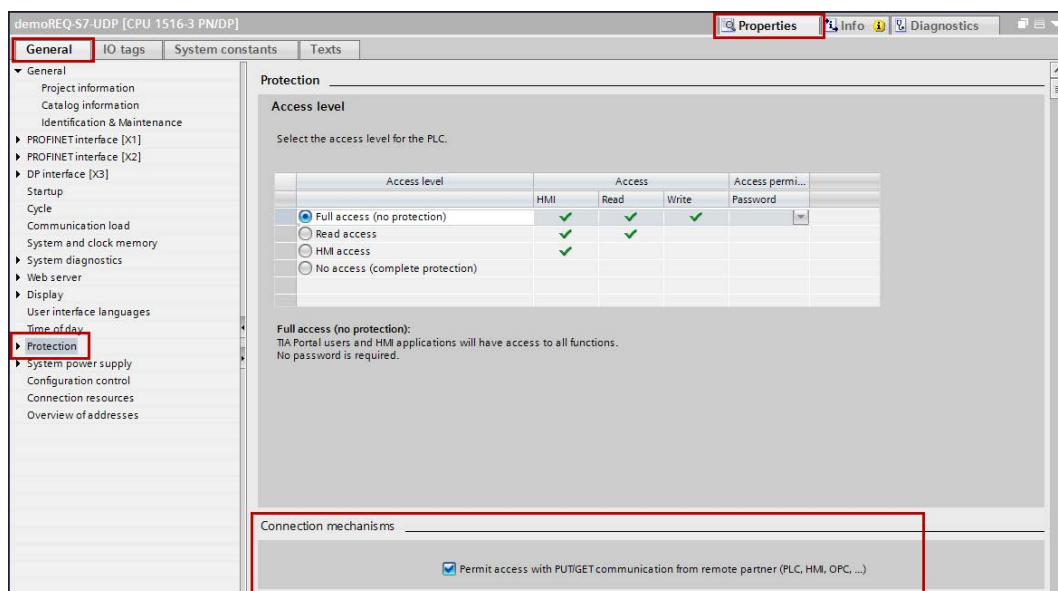
Fig. 17: Device configuration

1. Disable ibaREQ_DB (DB15) Properties – Attributes – Optimized block access



When using the connection mode TCP/IP (not TCP/IP S7-1x00) with S7-1500 CPUs, the following settings at the device configuration have to be made in addition:

2. Properties – General – Protection – Connection mechanisms: Permit access with PUT/GET communication



Access protection S7-1500

An access protection can be enabled for an S7-1500 CPU. The following dependencies on *ibaPDA* apply:

Access level	CPU access	ibaPDA reads symbolic from CPU	S7-Request access
Full access (no protection)	HMI, read, write	OK	OK
Read access	HMI, read	OK	OK
HMI access	HMI	No	OK
No access (complete protection)		No	No

Table 9: Access protection S7-1500

4.1.4 Configuration and engineering ibaPDA

4.1.4.1 General interface settings

The interface *ibaPDA-Request-S7-DP/PN* is configured in the *ibaPDA* "I/O Manager". Prerequisite is the installation of an *ibaFOB-D* card in the *ibaPDA* computer. When the *ibaFOB* card has been installed successfully, it is shown in the interface tree.

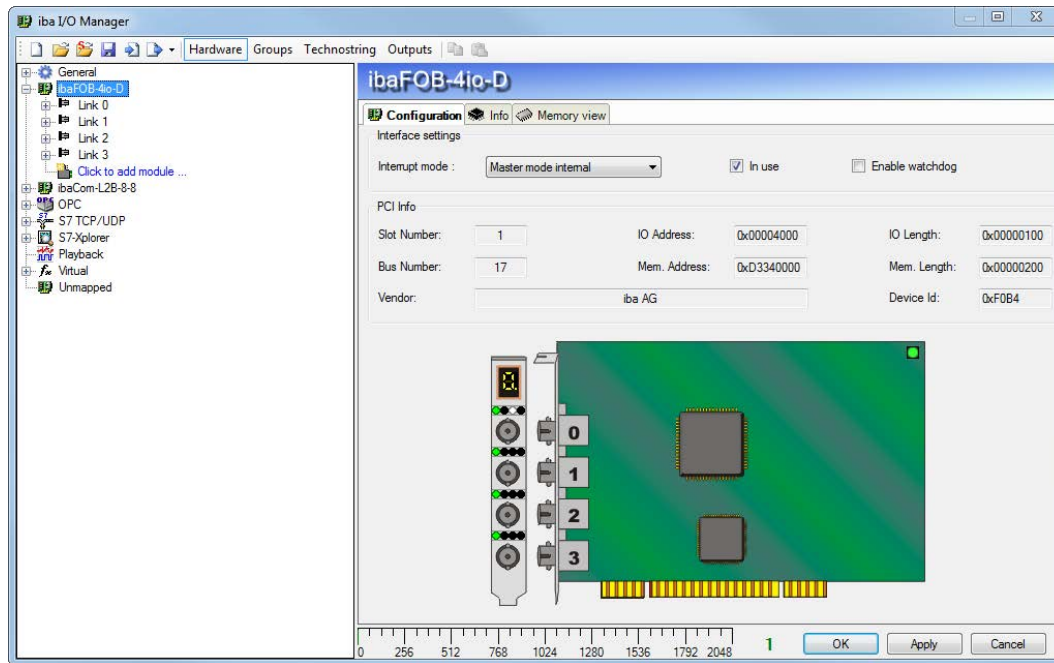




Fig. 18: I/O Manager, display ibaFOB-D card

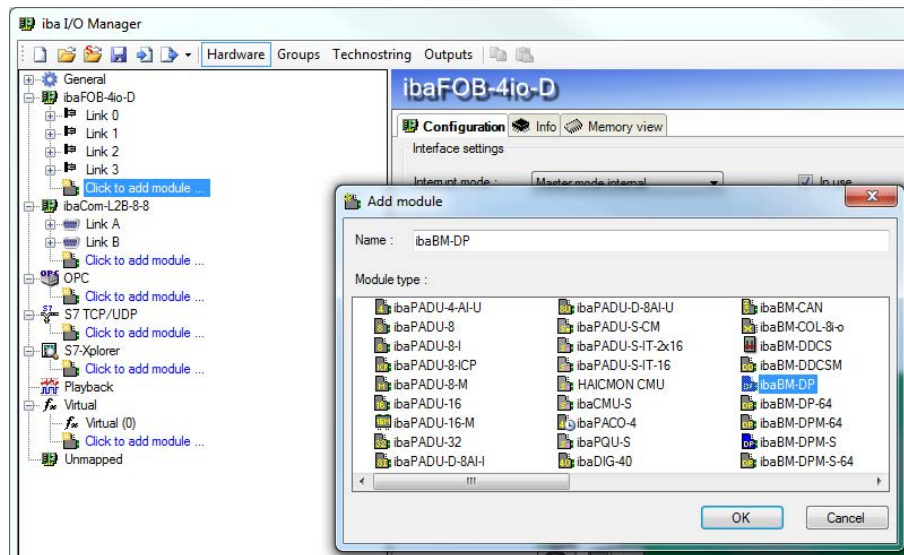
Other documentation



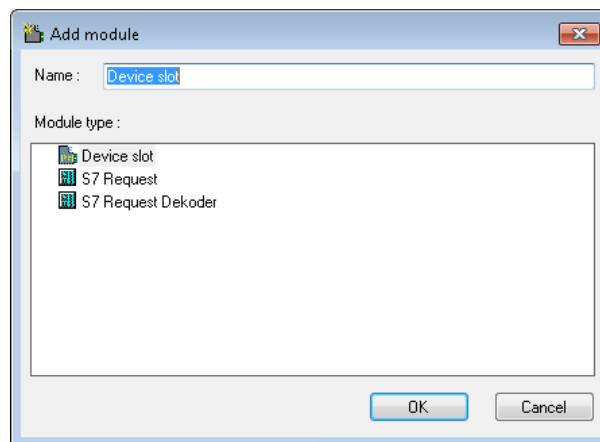
You can find more information about the *ibaFOB-D* card in the respective manual.

Configuring ibaBM-PN

1. Start the *ibaPDA* client  and open the I/O Manager .
2. Select the link of the *ibaFOB* card to which *ibaBM-PN* is connected in the interface tree (left). Click on the link "Click to add module..." and select the *ibaBM-PN* module.



3. Add a Request module (or several modules, in case you need more connections to one or to different S7-CPU's) to the *ibaBM-PN* module on the respective PROFINET connection Device 0 or Device 1. Available modules are:
 - S7 Request
(for acquiring analog and digital signals)
 - S7 Request Decoder
(for acquiring up to 1024 digital signals)



4. Configure the required module settings and signals as described in the following chapters. The *General* and *Connection* tabs are identical for all Request modules. The Request modules only differ in the *Analog* and *Digital* tabs.
5. After you have finished the configuration, click <Apply> or <OK> to transfer the new configuration to the device and start data acquisition with *ibaPDA*.

4.1.4.2 General module settings

You find the description of all settings that are identical for all Request-S7 modules in chapter [↗ General interface settings](#), page 16.

The *ibaBM-PN* modules have the following common specific setting options:

Profinet

Device

Number of the PROFINET device, to which the module is assigned

Slot

Number of the slot, to which the module is assigned

4.1.4.3 Connection settings

On the *Connection* tab, the connection to the controller is configured.

The following controllers, connection modes and selection methods are supported:

Controller	Connection mode		
	TCP/IP	PC/CP	TCP/IP S7-1x00
S7-300	X	X	
S7-400	X	X	
S7-1500	X		X

Depending on the connection mode different settings have to be made.

4.1.4.3.1 Connection mode TCP/IP

This mode activates a connection via the standard network interface of the PC.

Fig. 19: Connection configuration TCP/IP

CPU Name

Selection of the linked address book

Connection mode

Selection of the TCP/IP connection mode

Connection type

Selection of the connection type PG, OP or other connections (determines which type of connection resource is occupied on the CPU).

Timeout

Timeout of the connection

Address

IP address of the controller

Rack

Rack number of the controller (default: 0)

Slot

Slot number of the controller in the rack
(for: S7-1500 CPUs: 0)

Test

Connection test to the CPU and available diagnostic data are issued

The screenshot shows the '1: S7 Request (0)' configuration window. The 'Connection' tab is active, displaying various settings for a TCP/IP connection. Below the settings, a section titled 'Connection established' provides detailed diagnostic data for the connected PLC.

Connection Settings:

- Connection mode: TCP/IP
- Connection type: PG connection
- Timeout (s): 15
- Address: 192.168.50.95
- Rack: 0
- Slot: 0
- Test button
- Activate S7 routing: ☐
- DB: 15
- Profinet device slot: 1
- CPU Name: No addressbook
- Detect S7 restart (This applies to all S7 request modules): ☒

Connection established:

- MLFBNr of PLC is: **6ES7 412-2EK06-0AB0**
- PLC status: **RUN**
- Cycle times: Actual **1 ms** Min **1 ms** Max **2 ms**
- Reading **DB15**
- DB id: **ibaREQ-S7-M**
- DB version: **1.0.0.0**
- FB version: **1.0.0.0**
- DB length: **5280**
- Max. pointers: **128**
- Max. data bytes: **252**
- Hw/ version: **0**
- Total memory size: **1072432**
- DB memory size: **528384**
- DB used size: **11762**
- Code memory size: **544048**
- Code used size: **26580**
- No. inputs: **128**
- No. outputs: **128**
- No. markers: **4096**
- No. timers: **2048**
- No. counters: **2048**
- I/O space: **4096**
- Local datasize: **4096**

Device slot	Length	Status
1	252	available

Fig. 20: Connection test / diagnostic data

Tip

Error message "DB is not a valid request DB ..."

Please check if...

- the Request block has been loaded into the CPU
 - the right DB number has been configured on the Request block
 - the Request block is called in the program
 - possibly, the DB is written from another source.
-

Activate S7 routing

(see also ↗ *S7 routing*, page 157)

Activate this option, if the S7-CPU and the *ibaPDA*-PC are not in the same network, but only communicate over a gateway that supports S7 routing. Such a gateway can be e. g. an IE/PB link, over which a S7-CPU can be reached without an Ethernet connection.

Two additional input fields appear:

- Address of the device acting as gateway: Enter IP address of the gateway
- S7 Subnet ID of target net: Enter subnet ID from STEP 7 NetPro

Detect S7 restart:

The current request configuration is stored in a data block on the CPU. In case the "Detect S7 restart" option is enabled, *ibaPDA* can detect if this data block has been deleted or overwritten, e. g. as a result of loading the offline program or due to a cold restart and will restart the data acquisition. The configuration data are transferred again. This does not affect a warm restart of the CPU.

4.1.4.3.2 Connection mode PC/CP

This mode activates a connection over the interface card of the PC, which is configured using SIMATIC Net.

The interfaces configured in SIMATIC Net can be used, e. g.:

- MPI adapter (COM)
- MPI adapter (USB)
- PROFIBUS (CP5611)
- TCPIP (RFC1005)
-

Note

If you want to use this connection type, the Siemens software SIMATIC Net (e.g. SIMATIC Manager or Softnet) has to be installed. When using the modules CP55..., CP56... and the MPI adapter, the installation of the device drivers will be sufficient.

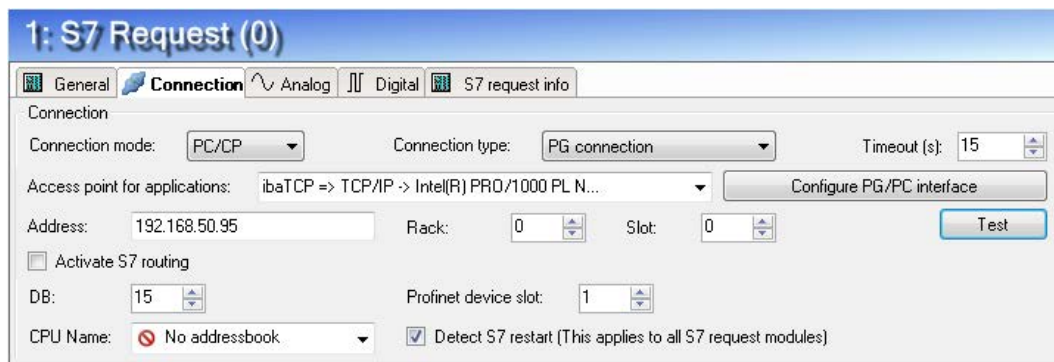


Fig. 21: Connection configuration PC/CP

CPU Name

Selection of the linked address book

Connection mode

Selection of the PC/CP connection mode

Connection type

Selection of the connection type PG, OP or other connections (determines which type of connection resource is occupied on the CPU).

Timeout

Timeout of the connection

Access point for applications

Selection of the access point that is to be used

For notes on creating and adapting an access point, see [➤ Setting PG/PC interface / defining new access point](#), page 153

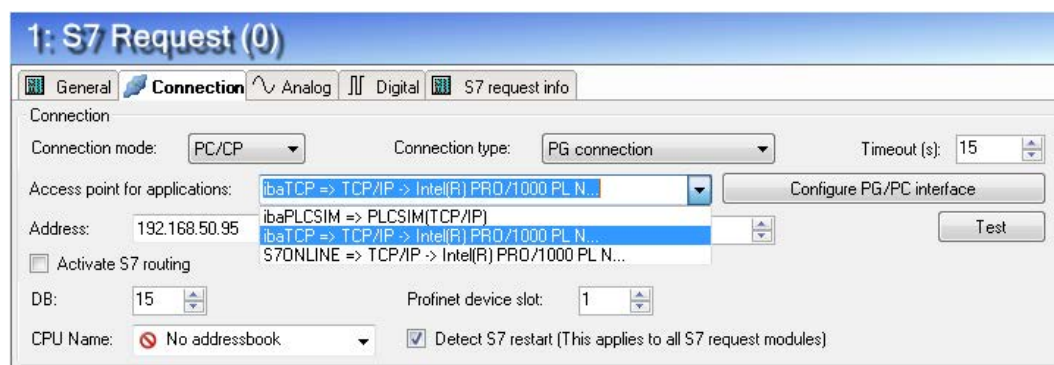


Fig. 22: Select access point

Note

The access points that are available under SIMATIC Net are configured using the Siemens tool "PG/PC interface settings".

In case *ibaPDA-Request-S7-DP/PN* and SIMATIC Manager run on the same computer, iba recommends configuring a special access point for *ibaPDA* for the connection between *ibaPDA-Request-S7-DP/PN* and SIMATIC S7 via PC/CP connections. A separate access point for *ibaPDA-Request-S7-DP/PN* prevents the risk that access for *ibaPDA* is impaired when changing the standard access point in the SIMATIC Manager.

Configure PG/PC interface

Opens the dialog box for setting the PG/PC interface of SIMATIC STEP 7

Address

Address of the controller (MPI, PROFIBUS or IP address depending on the configured access point)

Rack

Rack number of the controller (default: 0)

Slot

Slot number of the controller in the rack
(for: S7-1500 CPUs: 0)

Test

Connection test to the CPU and available diagnostic data are issued.

1: S7 Request (0)

General | **Connection** | Analog | Digital | S7 request info

Connection mode: PC/CP | Connection type: PG connection | Timeout (s): 15

Access point for applications: ibaTCP => TCP/IP -> Intel(R) PRO/1000 PL N... | Configure PG/PC interface

Address: 192.168.50.95 | Rack: 0 | Slot: 0 | Test

☐ Activate S7 routing

DB: 15 | Profinet device slot: 1

CPU Name: No addressbook | ☒ Detect S7 restart (This applies to all S7 request modules)

Connection established
 MLFBNr of PLC is: **6ES7 412-2EK06-0AB0**
 PLC status: **RUN**
 Cycle times: Actual **1 ms** Min **1 ms** Max **2 ms**
 Reading **DB15**
 DB id: **ibaREQ-S7-M**
 DB version: **1.0.0.0**
 FB version: **1.0.0.0**
 DB length: **5280**
 Max. pointers: **128**
 Max. data bytes: **252**
 Hw/ version: **0**
 Total memory size: **1072432**
 DB memory size: **528384**
 DB used size: **11762**
 Code memory size: **544048**
 Code used size: **26580**
 No. inputs: **128**
 No. outputs: **128**
 No. markers: **4096**
 No. timers: **2048**
 No. counters: **2048**
 I/O space: **4096**
 Local datasize: **4096**

Device slot	Length	Status
1	252	available

Fig. 23: Connection test / diagnostic data

Tip

Error message "DB is not a valid request DB ..."

Please check if...

- the Request block has been loaded into the CPU
- the right DB number has been configured on the Request block
- the Request block is called in the program
- possibly, the DB is written from another source.

Activate S7 routing

(see also [S7 routing](#), page 157)

Activate this option, if the S7-CPU and the *ibaPDA*-PC are not in the same network, but only communicate over a gateway that supports S7 routing. Such a gateway can be e. g. an IE/PB link, over which a S7-CPU can be reached without an Ethernet connection.

Two additional input fields appear:

- Address of the device acting as gateway: Enter IP address of the gateway
- S7 Subnet ID of target net: Enter subnet ID from STEP 7 NetPro

Detect S7 restart:

The current request configuration is stored in a data block on the CPU. In case the "Detect S7 restart" option is enabled, *ibaPDA* can detect if this data block has been deleted or overwritten, e. g. as a result of loading the offline program or due to a cold restart and will restart the data acquisition. The configuration data are transferred again. This does not affect a warm restart of the CPU.

4.1.4.3.3 Connection mode TCP/IP S7-1x00

This mode activates a connection over the standard network interface of the PC and can only be used in combination with S7-1500 CPUs.



Fig. 24: Connection configuration TCP/IP S7-1x00

CPU Name

Selection of the linked address book; only address books of the "TIA Portal" type can be selected.

Connection mode

Selection of the TCP/IP S7-1x00 connection mode

Connection type

Selection of the connection type PG, OP or other connections (determines which type of connection resource is occupied on the CPU).

Timeout

Timeout of the connection

Address

IP address of the controller

Test

Connection test to the CPU and available diagnostic data are issued.

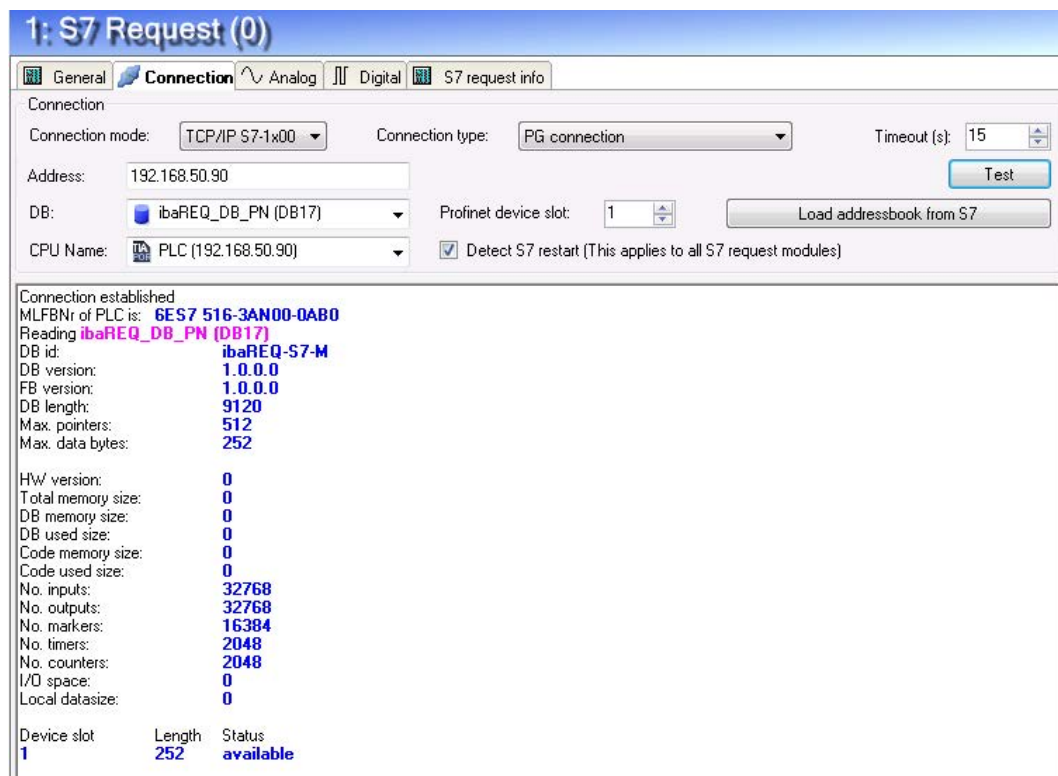
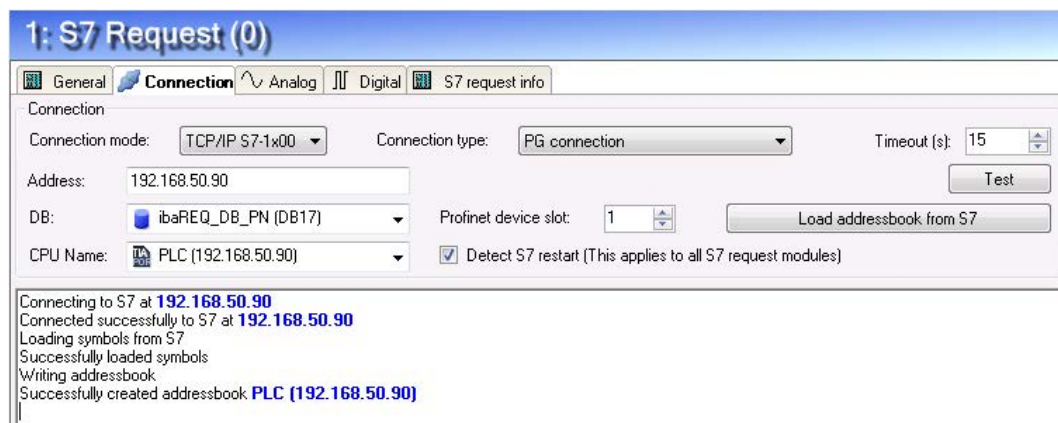


Fig. 25: Connection test / diagnostic data

Load addressbook from S7

By clicking on this button *ibaPDA* will read the list of symbols directly from the PLC and store it in an addressbook for further use in the symbol browser.



Tip

Error message "DB is not a valid request DB ..."

Please check if...

- the Request block has been loaded into the CPU
 - the right DB number has been configured on the Request block
 - the Request block is called in the program
 - possibly, the DB is written from another source.
-

In this context, also note the access protection of an S7-1500 CPU.

For more information, see ➤ *Device configuration Device configuration*, page 45

Detect S7 restart:

The current request configuration is stored in a data block on the CPU. In case the "Detect S7 restart" option is enabled, *ibaPDA* can detect if this data block has been deleted or overwritten, e. g. as a result of loading the offline program or due to a cold restart and will restart the data acquisition. The configuration data are transferred again. This does not affect a warm restart of the CPU.

4.1.4.4 Module S7 Request

With the "S7 Request" module it is possible to acquire data according to the size of the PROFINET slots used. A maximum of up to 254 bytes is possible.

A separate PROFINET slot and Request block call has to be configured for each module.

For a description of the module settings, see chapter ➤ *General module settings*, page 16 and ➤ *General module settings*, page 49.

4.1.4.5 Module S7 Request Decoder

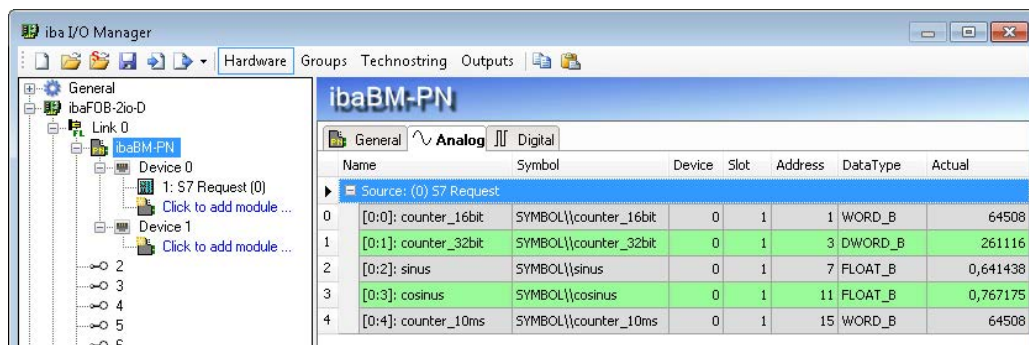
The "S7 Request Decoder" module allows the acquisition of up to 16384 digital signals sent as a maximum of 128 words (16 Bit). This module type is suited for applications where large amounts of digital signals have to be acquired and for which the max. 1024 directly addressable digital values of the *ibaBM-PN* are not sufficient.

A separate PROFINET slot and Request block call has to be configured for each module.

For a description of the module settings, see chapter ➤ *General module settings*, page 16 and ➤ *General module settings*, page 49.

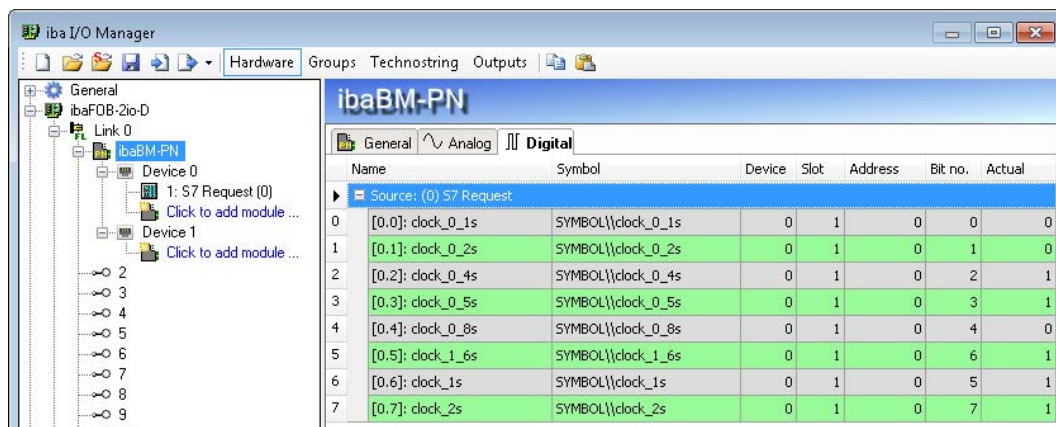
4.1.5 Diagnostics

When you select the bus module node in the signal tree and choose the *Analog* or *Digital* tab, you get a list of all operands that have been acquired in the bus module with data type and actual value.



	Name	Symbol	Device	Slot	Address	Data Type	Actual
Source: (0) S7 Request							
0	[0:0]: counter_16bit	SYMBOL\counter_16bit	0	1	1	WORD_B	64508
1	[0:1]: counter_32bit	SYMBOL\counter_32bit	0	1	3	DWORD_B	261116
2	[0:2]: sinus	SYMBOL\sinus	0	1	7	FLOAT_B	0,641438
3	[0:3]: cosinus	SYMBOL\cosinus	0	1	11	FLOAT_B	0,767175
4	[0:4]: counter_10ms	SYMBOL\counter_10ms	0	1	15	WORD_B	64508

Fig. 26: List of the acquired operands on the Analog tab



	Name	Symbol	Device	Slot	Address	Bit no.	Actual
Source: (0) S7 Request							
0	[0:0]: clock_0_1s	SYMBOL\clock_0_1s	0	1	0	0	0
1	[0:1]: clock_0_2s	SYMBOL\clock_0_2s	0	1	0	1	0
2	[0:2]: clock_0_4s	SYMBOL\clock_0_4s	0	1	0	2	1
3	[0:3]: clock_0_5s	SYMBOL\clock_0_5s	0	1	0	3	1
4	[0:4]: clock_0_8s	SYMBOL\clock_0_8s	0	1	0	4	0
5	[0:5]: clock_1_6s	SYMBOL\clock_1_6s	0	1	0	6	1
6	[0:6]: clock_1s	SYMBOL\clock_1s	0	1	0	5	1
7	[0:7]: clock_2s	SYMBOL\clock_2s	0	1	0	7	1

Fig. 27: List of the acquired operands on the Digital tab

Other documentation



You find a detailed description of the device-specific diagnostic options of *ibaBM-PN* in the device manual.

4.2 Request-S7 for ibaBM-PN in redundancy mode

In the following, the Request-S7 version for the PROFINET bus module *ibaBM-PN* in S2 redundancy mode is described.

4.2.1 General information

With the S2 redundancy mode of *ibaBM-PN*, the device can be operated on redundant PROFINET systems (S2 system redundancy) in combination with SIMATIC S7-R/H controllers, whose measurement data is to be acquired. You need an additional license for using the S2 redundancy mode of the *ibaBM-PN*. Please contact the iba AG support team.

The following figure shows an exemplary integration of an *ibaBM-PN* in S2 redundancy mode:

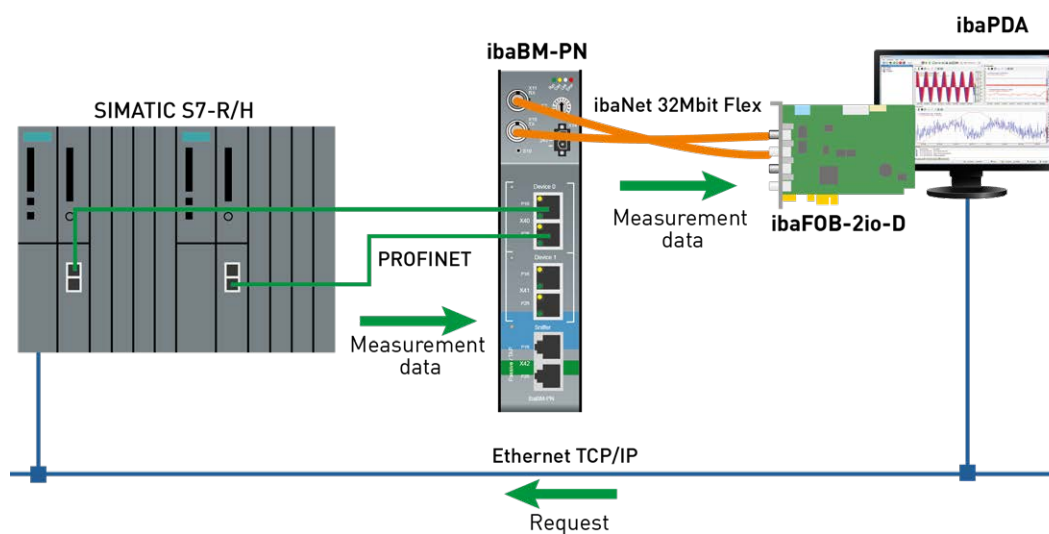


Fig. 28: Request-S7 for ibaBM-PN in S2 redundancy mode

Other documentation



For detailed information about the S2 redundancy mode of the *ibaBM-PN*, please see the device manual.

The functionality of the Request-S7 for *ibaBM-PN* in redundancy mode mostly corresponds to the functionality in standard mode (see chapter [Request-S7 for ibaBM-PN](#), page 32). The differences and extensions are described in the following.

4.2.2 Configuration and engineering SIMATIC S7-300, S7-400 and WinAC

4.2.2.1 Description of the Request blocks

For use in S2 redundancy mode, the same request blocks are used as in normal operation (see chapter [➤ Description of the Request blocks](#), page 34).

4.2.2.2 Hardware configuration

For each PROFINET device used, a separate PROFINET device has to be configured in the hardware configuration.

The same GSDML file is to be used as for normal operation (from version V2.35-20200101).

Note



You find the latest version of the GSDML file on the "iba Software & Manuals" DVD in the following directory:

\02_iba_Hardware\ibaBM-PN\01_GSD_Files\

When inserting an *ibaBM-PN* in a redundant hardware configuration, this is automatically connected with both PROFINET IO systems.

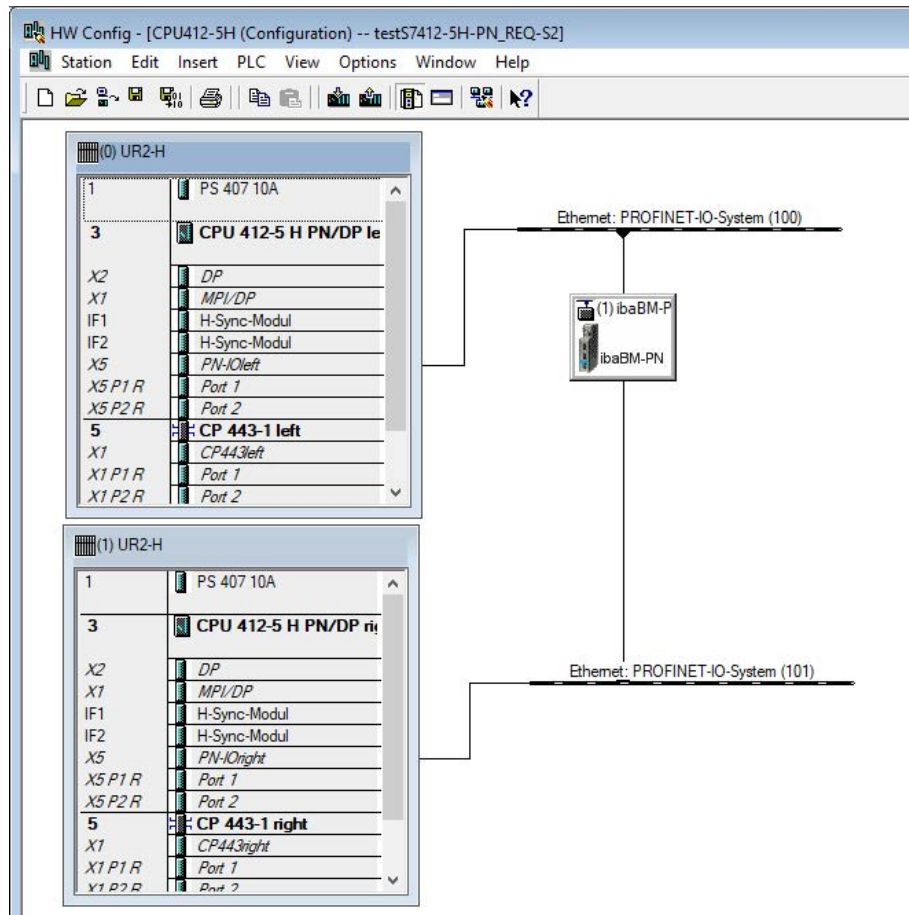


Fig. 29: HW config redundant system

This can be adapted in the *Redundancy* tab of Properties.

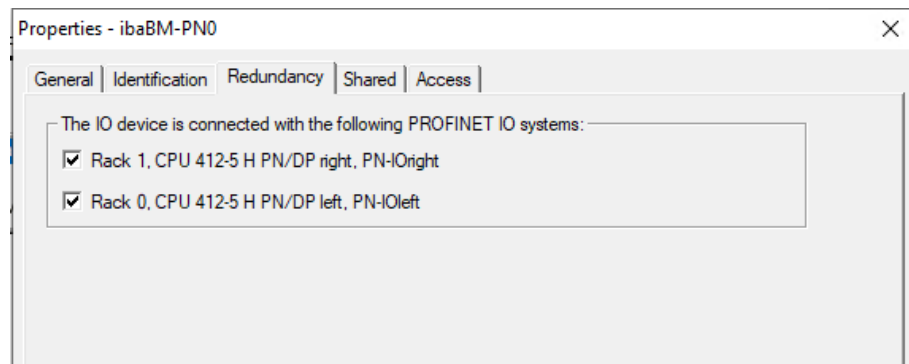


Fig. 30: Redundancy settings

ibaBM-PN is operated in the S2 redundancy mode by applying this hardware configuration.

4.2.2.3 Configuration in STEP 7

The configuration of the request blocks corresponds to the configuration in normal operation (see chapter [↗ Configuration in STEP 7](#), page 38).

4.2.3 Configuration and engineering ibaPDA

First connect *ibaBM-PN* to a free link of an ibaFOB-D card. In the I/O Manager, add a device module *ibaBM-PN* to the corresponding link.

Then add a module "S7 Request."

In the *General* tab, set the parameter "S2 Redundancy" to "True." The redundancy mode switched on in this way is indicated by an orange colored icon of the device module.

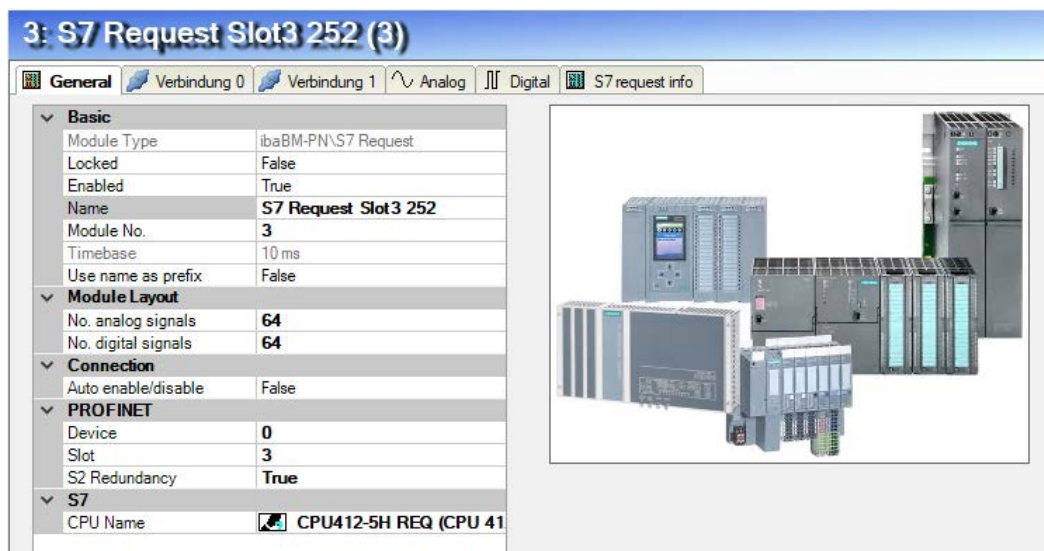


Fig. 31: Setting redundancy mode in the I/O Manager

The configuration of the modules otherwise corresponds to the configuration in standard mode. You can find information about this in chapter [↗ General module settings](#), page 49

Connection settings

For transmitting the operand data to both CPUs of the SIMATIC S7-R/H, two different connections are established. These connections are switched on and off depending on availability and requirements.

Both connections are configured on the tabs *Connection 0* and *Connection 1*. The name of the tabs can be assigned in the "Connection name" fields.

Fig. 32: Connection settings for connection 0

Fig. 33: Connection settings for connection 1

The other settings are identical to the settings for operation without redundancy mode (see chapter [➤ Connection settings](#), page 49).

The following special features have to be considered:

- The rack numbers for an H system are 0 or 1 for the both redundant CPUs.
- The DB number, the PROFINET device slots and the CPU name for assigning an address book is set only once in the *Connection 0* tab.

4.3 Request-S7 for ibaBM-DP

In the following, we describe the Request-S7 version for the PROFIBUS bus module *ibaBM-DP*.

4.3.1 General information

Request-S7 for *ibaBM-DP* can be configured in the following system configurations:

SIMATIC S7 CPU	SIMATIC STEP 7 V5.x (SIMATIC Manager)	SIMATIC STEP 7 V1x Professional (TIA Portal)
S7-300 integrated DP interface	X	X
S7-400 integrated DP interface and CP443-5	X	X
S7-400 integrated DP interface and CP443-5	X	H-CPU's are not supported by the TIA portal, yet.
S7-1500 integrated DP interface and CM1542-5 or CP1542-5		X

For DP Request, the measurement values (Request Handshake) are not requested via the PROFIBUS, but via a separate connection.

Depending on the hardware and software, different access points can be selected for the Request:

- TCP/IP: the connection to the SIMATIC S7 is established over an integrated PN interface of the S7-CPU or the respective CP modules in the PLC and the standard network interface of the PC. No additional Siemens software is required for the connection.
- PC/CP: this is the designation for different SIMATIC specific access points. In contrast to the TCP/IP connection, the SIMATIC communication software (and the corresponding licenses) must be installed on the PC for all connection types within the PC/CP group.
 - MPI, PROFIBUS: The connection to the SIMATIC S7 is established via the MPI or PROFIBUS interface of the PC; e.g. with the PCI card CP5611 or the MPI adapter for USB or serial PC interface.
 - TCP/IP, ISO: here, either the standard network interface of the PC or a suitable interface card is used for the connection to the S7.
- TCP/IP S7-1x00: The connection to a SIMATIC S7-1500 is established via an integrated PN interface of the S7-CPU or respective CP modules in the PLC and the standard network interface of the PC. No additional Siemens software is required for the connection.

System integration with ibaBM-DP

The measured data are transmitted to the *ibaBM-DP* device over PROFIBUS DP.

You need the following connections:

- Online connection between *ibaPDA* and S7-CPU (TCP/IP, MPI or DP)
- Fiber optic connection between *ibaPDA/ibaFOB-io-D* and *ibaBM-DP*
- PROFIBUS connection between *ibaBM-DP* and S7 PROFIBUS master
- A connection from *ibaBM-DP* to the network (TCP/IP over Ethernet) is only needed if the device is operated in compatibility mode, i.e. not with 32Mbit Flex (short: Flex). In this case the system integration is like with *ibaBM-DPM-S* (see chapter [Request-S7 for ibaBM-DPM-S](#), page 98).

In the standard version a maximum of eight connections, i.e. eight PROFIBUS slaves, can be configured per device. Up to 244 Byte data can be transmitted per slave.

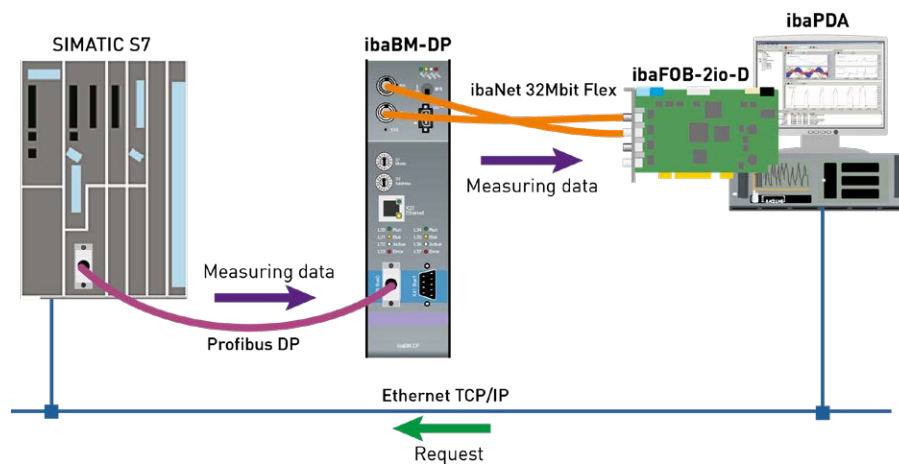


Fig. 34: Request-S7 with ibaBM-DP

Note



The following description refers to the Request blocks V4.0 or higher. Should you require information regarding older versions, please contact our support.

Other documentation



For detailed information about *ibaBM-DP*, please see the device manual.

For information and application examples, see chapter [Application examples](#), page 148.

4.3.1.1 ibaCom-L2B compatibility mode

ibaBM-DP can be used for the Request-S7 functionality of the *ibaCom-L2B-PROFIBUS* card as successor with full functional compatibility.

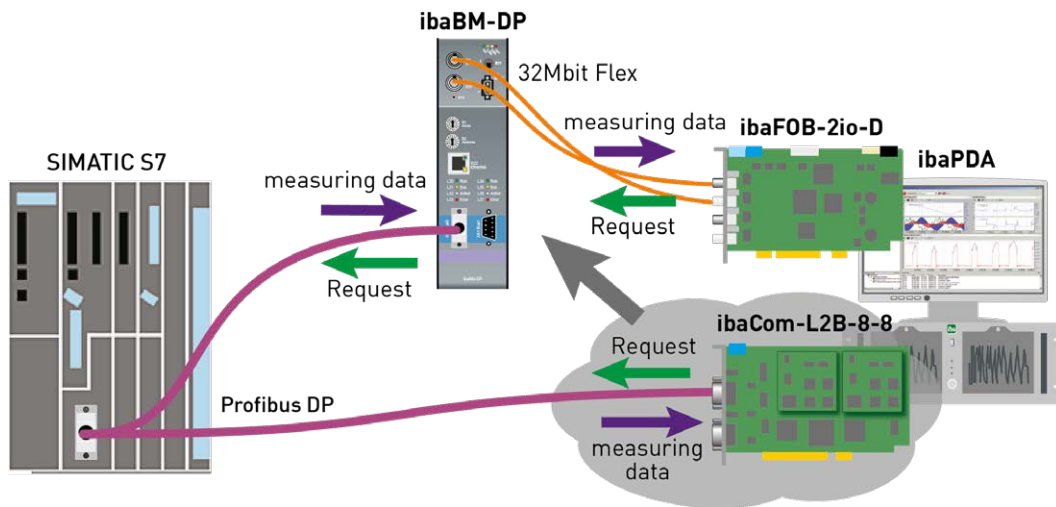


Fig. 35: Request-S7 with ibaBM-DP, replaces ibaCom-L2B

For this purpose, the following modules can be used:

- S7 Request (ibaCom-L2B compatible)
- S7 Request Dig512 (ibaCom-L2B compatible)

Compared to the original configuration on the basis of the *ibaCom-L2B* card, modifications in the S7 configuration are not necessary (Hardware or S7 program).

For detailed information about this subject, please see chapter ↗ *Request-S7 for ibaCom-L2B*, page 113 and chapter ↗ *Replacing Request-S7 on ibaCom-L2B by ibaBM-DP*, page 166.

Note



The *ibaCom-L2B* compatibility mode can only be used in the Flex-mode of *ibaBM-DP*.

4.3.2 Configuration and engineering SIMATIC S7-300, S7-400 and WinAC

In the following, we describe the configuration and engineering on the SIMATIC S7 side with SIMATIC Manager (STEP 7 Version ≤ V5) when using the modules "S7 Request" and "S7 Request Decoder".

The configuration and engineering when using the compatibility modules "S7 Request (ibaCom-L2B compatible)" and "S7 Request Dig512 (ibaCom-L2B compatible)" corresponds to using an *ibaCom-L2B* card. You find a description in chapter ↗ *Configuration and engineering SIMATIC S7-300, S7-400 and WinAC*, page 114.

4.3.2.1 Description of the Request-FC ibaDP_Req (FC122)

With this function, the communication between the S7, *ibaPDA* and the *ibaBM-DP* device is initialized and controlled.

The ibaDP_Req has to be called once for each slave in the cyclic program.

The used blocks are part of the iba S7 library (see chapter [Iba S7 library](#), page 139).

Description of the formal parameters of the ibaDP_Req (FC122):

Name	Type	Data Type	Description
DB_PDA	IN	BLOCK_DB	DB of the baPDA communication interface ibaDP_DB_PDA (2064 Byte)
DB_INTERN	IN	BLOCK_DB	DB of the internal data interface ibaDP_DB_work (1900 Byte)
OUTPUT_ADR_SLAVE	IN	INT	Start address of the peripheral output range, continuous and 244 Bytes long
INIT_FC	IN	BOOL	TRUE -> Initialize
INP_RANGE	IN	INT	Number of output bytes (evaluation only during initialization), 0:automatic detection (recommended)
OUT_RANGE	IN	INT	Number of output bytes (evaluation only during initialization), 0:automatic detection (recommended)
MARKER_RANGE	IN	INT	Number of marker bytes (evaluation only during initialization), 0:automatic detection (recommended)
ERROR_STATUS_INIT	OUT	BYTE	Initialization error
ERROR_STATUS_COM	OUT	BYTE	Communication error

The following SFCs are used internally:

- SFC 13 (DPNRM_DG)
- SFC 15 (DPRD_DAT)
- SFC 20 (BLKMOV)
- SFC 21 (FILL)
- SFC 24 (TEST_DB)
- SFC 49 (LGC_GADR)
- SFC 50 (RD_LGADR)
- SFC 51 (RDSYSST)

Detailed description:**DB_PDA**

DB in the range of 1 to n (see technical data of the CPU). The DB is used for data exchange with *ibaPDA*. Length at least 2064 Bytes.

DB_INTERN

DB in the range of 1 to n (see technical data of the CPU).
Length at least 1900 Bytes.

The following data are stored in this DB:

- Technical data of CPU which are determined during initialization
- The S7 pointers requested by *ibaPDA*
- The binary and analog transmission data

OUTPUT_ADR_SLAVE

DP start address of the *ibaBM-DP* in the peripheral output range for sending data. Length 244 Bytes, continuous (without gaps!)

INIT_FC

Used for initializing the Request block. The Request block is initialized automatically internally. Additionally, the initialization can be performed manually by an external logic via the input `INIT_FC = TRUE`.

INP_RANGE

Limits the number of input bytes to be measured.

If `INP_RANGE = 0`, the size of the available process image of the inputs will be evaluated by the Request block (recommended). Evaluation is done only during the initialization phase of the module.

OUT_RANGE

Limits the number of output bytes to be measured.

With `OUT_RANGE = 0` the size of the available process image will be evaluated by the Request block (recommended). Evaluation is done only during the initialization phase of the module.

MARKER_RANGE

Limits the number of marker bytes to be measured.

With `MARKER_RANGE = 0`, the number of the available markers will be evaluated by the Request block (recommended). Evaluation is done only during the initialization phase of the module.

ERROR_STATUS_INIT

The following error codes can be generated:

- 1: DB_PDA is read-only
- 2: DB_PDA-No = 0 or higher than the max. permissible DB number of this CPU
- 3: DB_PDA with the specified number is not available
- 5: DB_PDA is too short
- 11: DB_INTERN is read-only
- 12: DB_INTERN-No = 0 or higher than the max. permissible DB number of this CPU

13: DB_INTERN with the specified number is not available
15: DB_INTERN is too short
16: Error when reading the identification data of the CPU
19: Initialization not finished
21: Not enough memory space for the data set
22: SZL_ID is wrong or not known in this CPU
23: The index of SZL is wrong or not permitted
30: OUT_ADR_SLAVE is no peripheral output range
31: OUT_ADR_SLAVE has no ROFIBUS DP assigned

ERROR_STATUS_COM

The following error codes can be generated:

100: Bit number not 0
101: Bit number not 0-7
103: The operand range is not defined.
104: The data type is not defined
105: DB0 has been requested as data source
106: DB number is higher than the max. permitted number of DBs for this CPU
107: DB with the specified number is not available
109: DB is too short for the accessed operand
110 Specified data address does not exist
111: Initialization aborted with error
(error cause can be seen in ERROR_STATUS_INIT)
112 Initialization not carried out
(error cause can be seen in ERROR_STATUS_INIT)
150: Order fragmentation is not supported
151: Order Request: Total number of values does not match values in the order
152: Order Request: Up to 64 binary values permitted
153: Order Request: Up to 64 analog values permitted
200: DP station is not available

4.3.2.2 Hardware configuration

For each module, an iba-PROFIBUS slave has to be defined.

The GSD file "ibaDPMSi.gsd" has to be used (version V2.2 and higher).

Note



You find the GSD-file "ibaDPMSi.gsd" on the DVD "iba Software & Manuals" in the following directory:

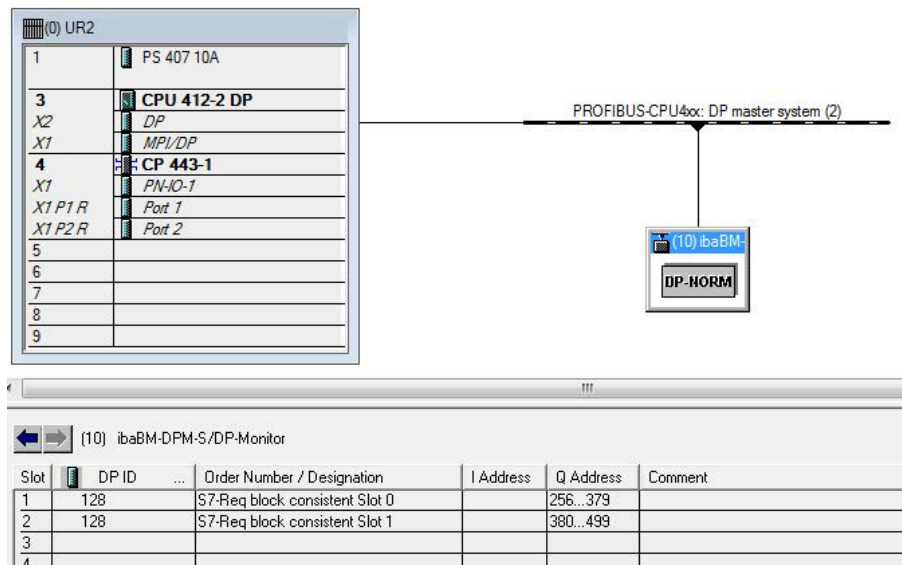
\02_iba_Hardware\ibaBM-DP\02_GSD_Files\01_General\

³⁾ More recent S7 CPU models are generally: S7-400 with firmware version 3.0 or higher and S7-300 with firmware version 2.0 or higher

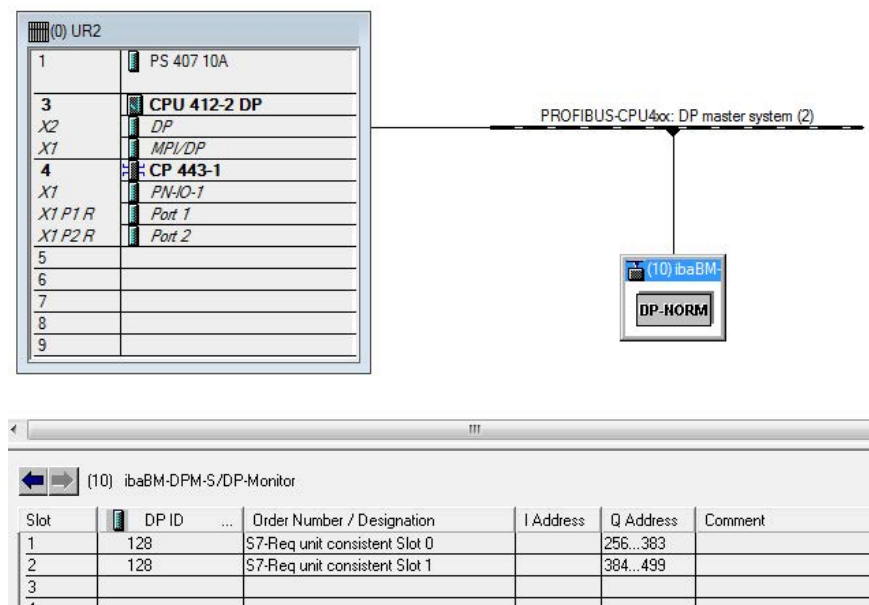
Note

The slots have to be created without gaps and with consecutive address range.

- **S7-CPU of newer type³⁾** provide consistent slots with a max. of 128 Bytes.
Use the elements "S7-Req **block consistent** Slot 0 / Slot 1".



- for **S7-CPUs and for CP443-5 of older types** no long consistent slots can be used.
Use the elements "S7-Req **unit consistent** Slot 0 / Slot 1"



4.3.2.3 Configuration in STEP 7 (STL, LAD, FBD)

Request-S7 is intended for the use with CPUs S7-300/400 with integrated DP interface or CPUs S7-400 with external interface CP 443-5 (PROFIBUS master).

Note



The use of the external DP interface CP342-5 of a CPU S7-300 is not supported!

For Request-S7 with one PROFIBUS slave:

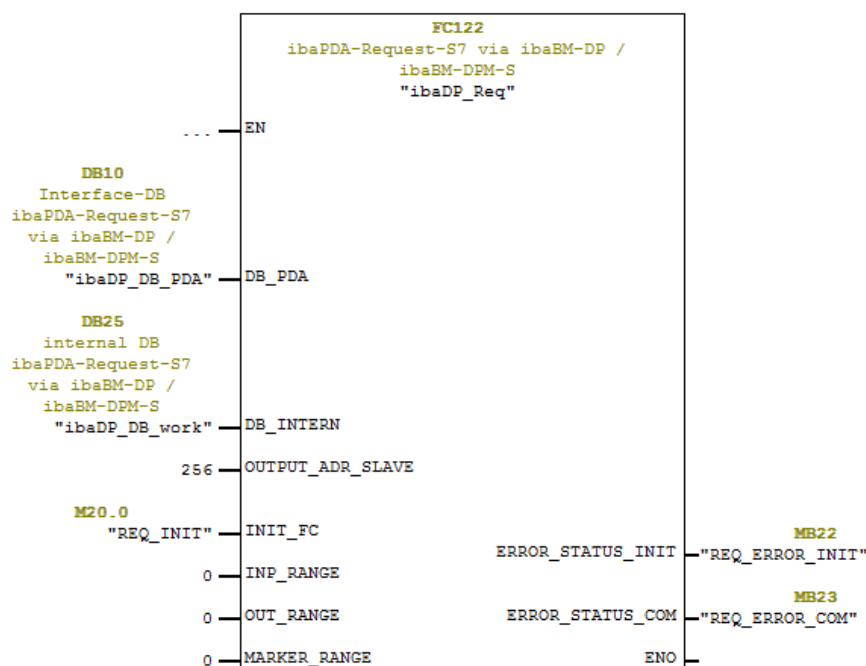
- Copy the following blocks from the iba S7 library (see chapter [Iba S7 library](#), page 139) to the blocks folder of your STEP 7 project:
 - FC122 (ibaDP_Req)
 - DB10 (ibaDP_DB_PDA)
 - DB25 (ibaDP_DB_work).

Note



If the numbers FC122, DB10 and DB25 are already used in your project, assign new numbers to the blocks from the iba S7 library when copying.

- Create the error OBs (OB82, OB85, OB86, OB87, OB122) in order to prevent CPU stops in case of an error.
- Call and parameterize the ibaDP_Req (FC122) in the cyclic program.



For Request-S7 with multiple PROFIBUS slaves:

- In the blocks folder, a data block ibaDP_DB_PDA (DB10) has to be available for each slave. Copy the DB10 in a DB with a new DB number.
- In the blocks folder, a data block ibaDP_DB_work (DB25) has to be available for each slave. Copy the DB25 in a DB with a new DB number.
- In the cyclic OB, the ibaDP_Req (FC122) must be called again for each slave with the new DB numbers and the peripheral addresses of the new PROFIBUS slave.
- The input INIT_FC and the outputs ERROR_STATUS_INIT and ERROR_STATUS_COM should have unique markers (or DB elements) for each slave.

Finish:

- Load all blocks into the S7-CPU and restart.

4.3.2.4 Configuration in STEP 7 (CFC)**For Request-S7 with one PROFIBUS slave:**

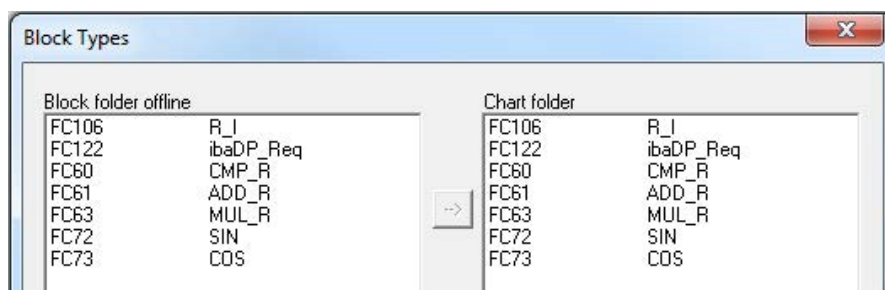
- Copy the following blocks from the iba S7 library (see chapter [Iba S7 library](#), page 139) to the blocks folder of your STEP 7 project:
 - FC122 (ibaDP_Req)
 - DB10 (ibaDP_DB_PDA)
 - DB25 (ibaDP_DB_work).

Note

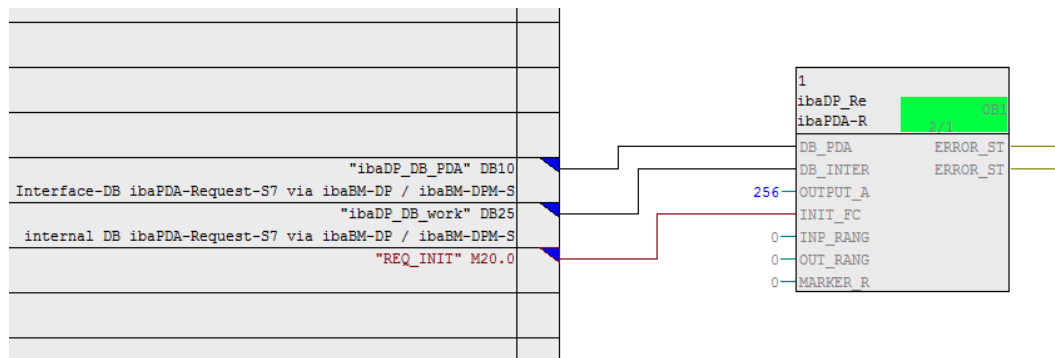
If the numbers FC122, DB10 and DB25 are already used in your project, assign new numbers to the blocks from the iba S7 library when copying.

The chosen block numbers have to match the reserved number ranges for DBs and FCs under the CFC settings for "Translating/Loading".

- Start the CFC editor and import the ibaDP_Req (FC122) (under menu "Options - Block Types"). The FC is stored in the group "ibaPDA" in the function block library.



- Drag the ibaDP_Req (FC122) to your CFC chart. Pay attention to the run sequence. The function block must be called in a cyclic OB (cyclic interrupt OB or OB1).



- Make sure that the error tasks OB82, OB85, OB86, OB87 and OB122 are created in order to prevent the CPU from running into STOP in case of access failure.
To do so, you have to define an empty runtime group within the sequence of these tasks if they are empty. Alternatively, you may place any dummy block into the error OB. This block must not be deleted, too.

Note



Do not choose the option "Delete empty runtime groups" for compilation otherwise the error OBs will be removed! Some older CFC versions have this option in the compilation dialog.

For Request-S7 with multiple PROFIBUS slaves:

- In the blocks folder, a data block ibaDP_DB_PDA (DB10) has to be available for each slave. Copy the DB10 in a DB with a new DB number.
- In the blocks folder, a data block ibaDP_DB_work (DB25) has to be available for each slave. Copy the DB25 in a DB with a new DB number.
- In the CFC chart, you have to create another instance of the ibaDP_Req (FC122) for each slave. Configure the connectors with the new DB numbers and the peripheral addresses of the new PROFIBUS slave.

After each program modification:

- Compile the complete program:
Activate the option "Entire program", activate the option "Generate SCL source" (for CFC version 6.1 or higher),
deactivate the option "Delete empty runtime groups" if available.
- Load the program and restart S7-CPU.

4.3.3 Configuration and engineering SIMATIC S7-1500

In the following, we describe the configuration and engineering on the SIMATIC S7 side with the SIMATIC TIA Portal.

Basically, the following configuration steps have to be carried out on the SIMATIC TIA Portal side:

- Network configuration
Integration of the PROFIBUS slaves in the device configuration
- Configuration Software:
Integration of the Request blocks in the S7 program
- Device configuration:
Setting the CPU protection properties

You cannot use the "S7 Request (ibaCom-L2B compatible)" and "S7 Request Dig512 (ibaCom-L2B compatible)" compatibility modules in combination with a CPU S7-1500.

4.3.3.1 Description of the Request blocks

The communication between the S7 and *ibaPDA* is initialized and controlled with these blocks. One set of Request blocks has to be called for each Request module (connection).

The used blocks are part of the iba S7 library (see chapter [↗ Iba S7 library](#), page 139).

Request block	S7-1500 CPU with integrated DP interface	S7-1500 CPU with CM1542-5 or CP1542-5	recommended call up level
ibaREQ_M (FB1400)	X	X	OB1
ibaREQ_DP (FB1402)	X	X	OB1 or OB3x
ibaREQ_DB (DB15)	X	X	-
ibaREQ_DB-Interface	X	X	-

Table 10: Combination of Request blocks

- ibaREQ_M (Management)
The block realizes the communication with *ibaPDA*. Ideally, the block is called in the OB1.
- ibaREQ_DP (provides and sends the actual signal values)
The block provides the actual signal values in the send cycle. Ideally, the block is called in a cyclic interrupt OB.
- ibaREQ_DB (interface DB)
This DB serves as interface to *ibaPDA* and between the different Request blocks.

4.3.3.1.1 ibaREQ_M (FB1400)

Description of the formal parameters:

Name	Type	Data Type	Description
REQ_DB	IN	DB_ANY	DB of the ibaPDA communication interface ibaREQ_DB
RESET	IN	BOOL	TRUE -> perform reset
CPU_HW_ID	IN	HW_IO	Hardware ID of local CPU
ERROR_STATUS	OUT	WORD	Error code

The following SIMATIC standard blocks are used internally:

GET_IM_DATA (FB801)

Detailed description:

REQ_DB

The DB is used for data exchange with *ibaPDA*. For all Request blocks that belong together, the identical DB has to be configured.

The length of the data block is fixed.

RESET

Serves for manually resetting the Request blocks. All Request blocks of a combination are automatically reset together. Usually, the parameter does not have to be connected

CPU_HW_ID

TIA Portal system constant, which refers to the corresponding CPU.

ERROR_STATUS

Internal error of the block. If there is no error, the value 0 will be issued.

For a list of all possible error codes, see chapter [➤ Error codes of Request blocks](#), page 168.

4.3.3.1.2 ibaREQ_DP (FB1402)

Description of the formal parameters:

Name	Type	Data Type	Description
ADR_SLOT_0	IN	VARIANT	Start address of the output range of slot 0
ADR_SLOT_1	IN	VARIANT	Start address of the output range of slot 1
RESET_CON	IN	BOOL	TRUE -> reset of the communication connection
REQ_DB	INOUT	UDT	DB of the ibaPDA communication interface ibaREQ_DB
ERROR_STATUS	OUT	WORD	Internal error code

Detailed description:**ADR_SLOT_0**

Start address of the used slot 0 in *ibaBM-DP* in the output range of the process image.

ADR_SLOT_1

Start address of the used slot 1 in *ibaBM-DP* in the output range of the process image.

REQ_DB

The DB is used for data exchange with *ibaPDA*. For all Request blocks that belong together, the identical DB has to be configured.

ERROR_STATUS

Internal error of the block. If there is no error, the value 0 will be issued.

For a list of all possible error codes, see chapter [➔ Error codes of Request blocks](#), page 168.

4.3.3.2 Network configuration

For each module, an iba-PROFIBUS slave has to be defined.

The GSD file "ibaDPMSi.gsd" has to be used (version V2.2 and higher).

Note

You find the GSD-file "ibaDPMSi.gsd" on the DVD "iba Software & Manuals" in the following directory:

\02_iba_Hardware\ibaBM-DP\02_GSD_Files\01_General\

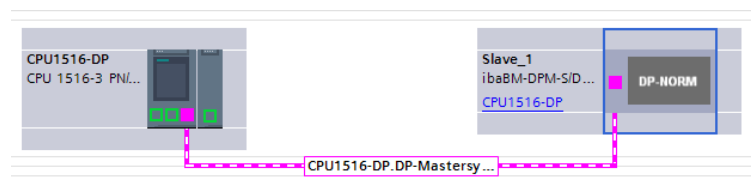


Fig. 36: Net view

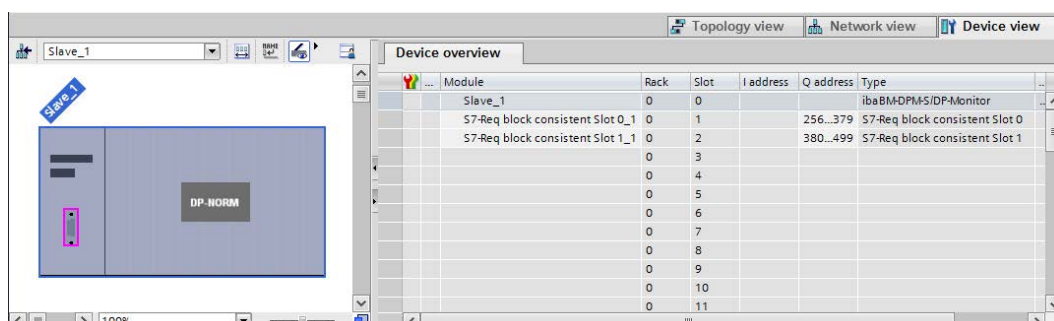


Fig. 37: Device view

4.3.3.3 Configuration in STEP 7

In the following, we describe how to configure the Request blocks in TIA Portal STEP 7.

Copy the required blocks from the iba S7 library (see chapter [Iba S7 library](#), page 139) to the blocks folder of your STEP 7 project:

Note



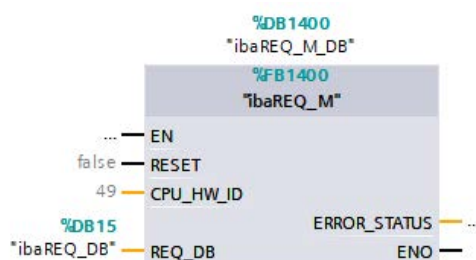
The request blocks do not support multi-instance calls.

The following blocks are required:

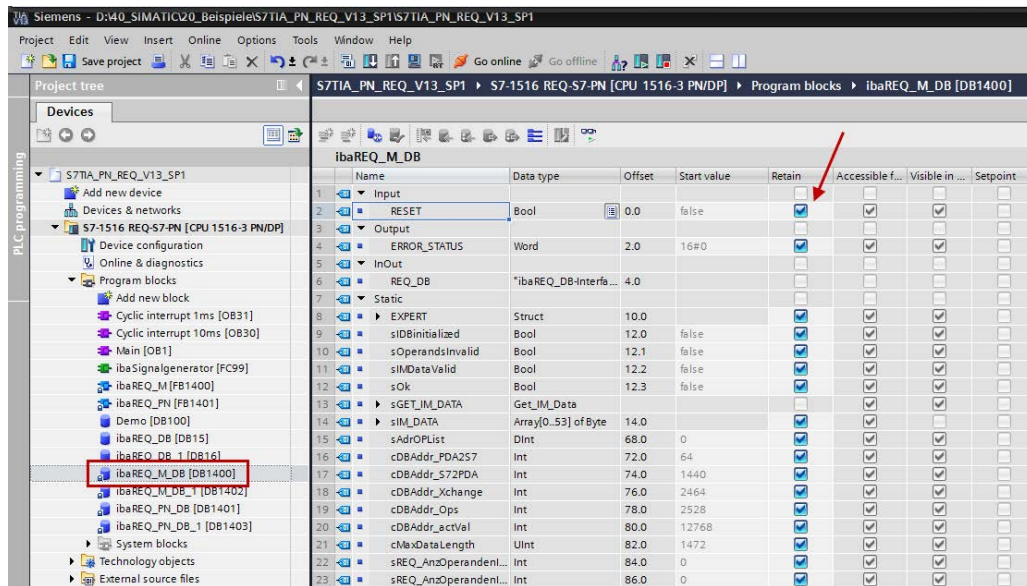
- ibaREQ_M (FB1400)
- ibaREQ_DP (FB1402)
- ibaREQ_DB (DB15)
- ibaREQ_DB-Interface (PLC data type)

For each Request module, the following steps have to be carried out:

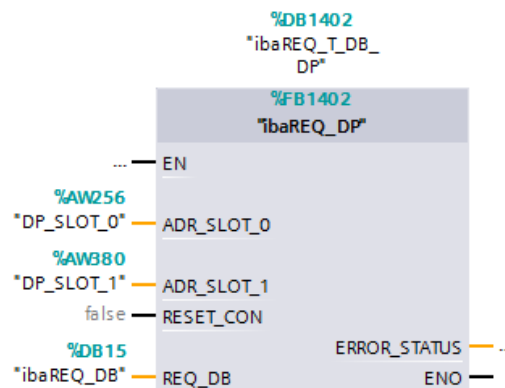
1. Call the ibaREQ_M (FB1400) preferably within the OB1



2. Enable "Retain" for the complete instance block, you have just created.



3. Call the ibaREQ_DP (FB1402), preferably within the OB1 or an cyclic interrupt (OB3x).



For each additional Request module:

- In the blocks folder, an ibaREQ_DB (DB15) has to be available for each Request module. Copy the data block and assign a new unique DB number.
- Within the OB1, the ibaREQ_M (FB1400) has to be called once more for each Request module with the new DB number.
- Within the OB1 or a cyclic interrupt (OB3x), the ibaREQ_DP (FB1402) has to be called for each Request module once more with the new DB numbers.
- Please consider that all instance data blocks have to be unique and that the values for the ADR_SLOT_0 and ADR_SLOT_1 are assigned uniquely.

Finish:

- Load all blocks into the S7-CPU and restart.

4.3.3.4 Device configuration

The following settings have to be done in the device configuration of the CPU:

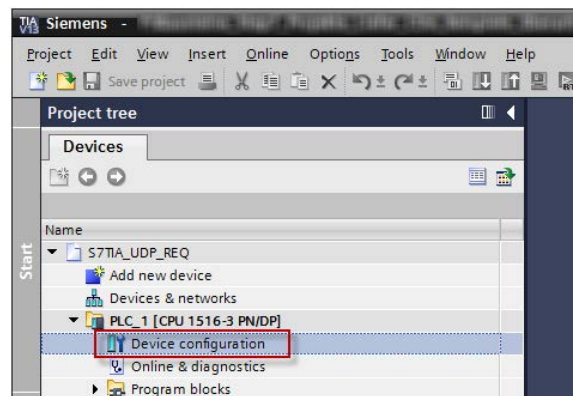
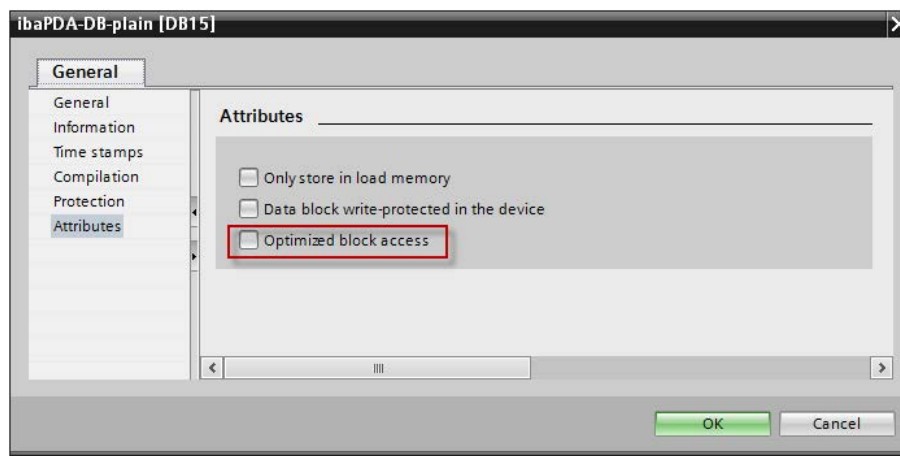


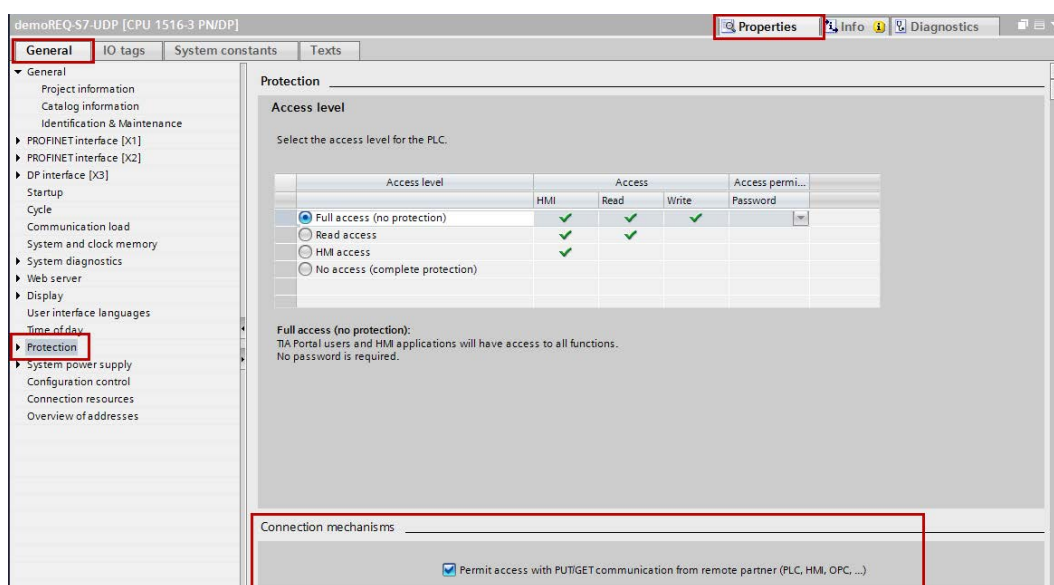
Fig. 38: Device configuration

1. Disable ibaREQ_DB (DB15) Properties – Attributes – Optimized block access



When using the connection mode TCP/IP (not TCP/IP S7-1x00) with S7-1500 CPUs, the following settings at the device configuration have to be made in addition:

2. Properties – General – Protection – Connection mechanisms: Permit access with PUT/GET communication



Access protection S7-1500

An access protection can be enabled for an S7-1500 CPU. The following dependencies on *ibaPDA* apply:

Access level	CPU access	ibaPDA reads symbolic from CPU	S7-Request access
Full access (no protection)	HMI, read, write	OK	OK
Read access	HMI, read	OK	OK
HMI access	HMI	No	OK
No access (complete protection)		No	No

Table 11: Access protection S7-1500

4.3.4 Configuration and engineering ibaPDA

4.3.4.1 General interface settings

The interface *ibaPDA-Request-S7-DP/PN* is configured in the *ibaPDA* "I/O Manager". Prerequisite is the installation of an *ibaFOB-D* card in the *ibaPDA* computer. When the *ibaFOB* card has been installed successfully, it is shown in the interface tree.

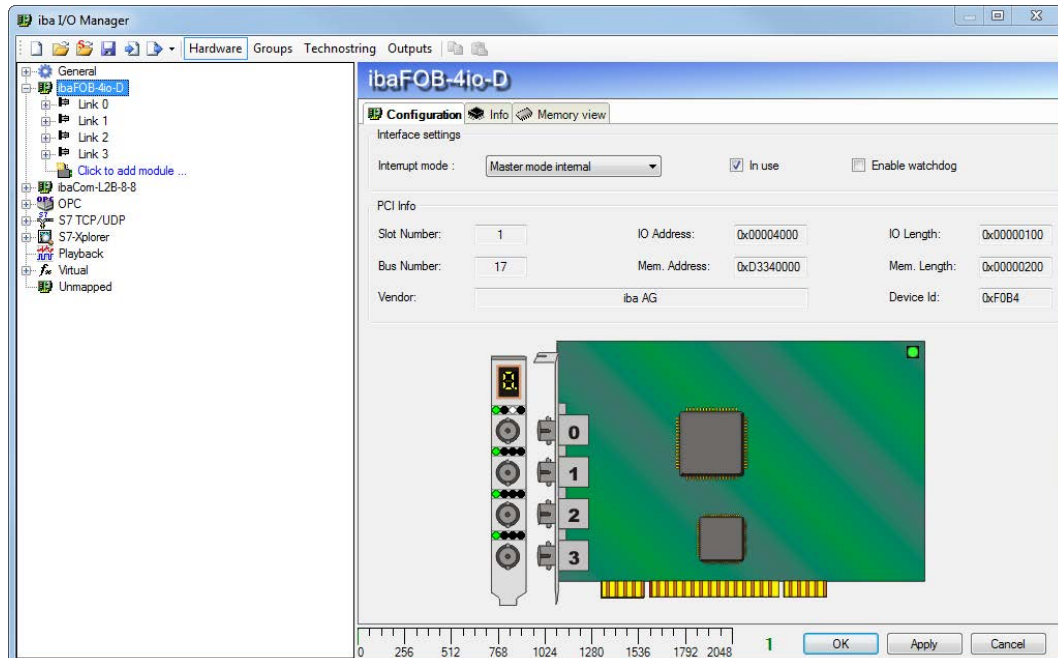




Fig. 39: I/O Manager, display ibaFOB-D card

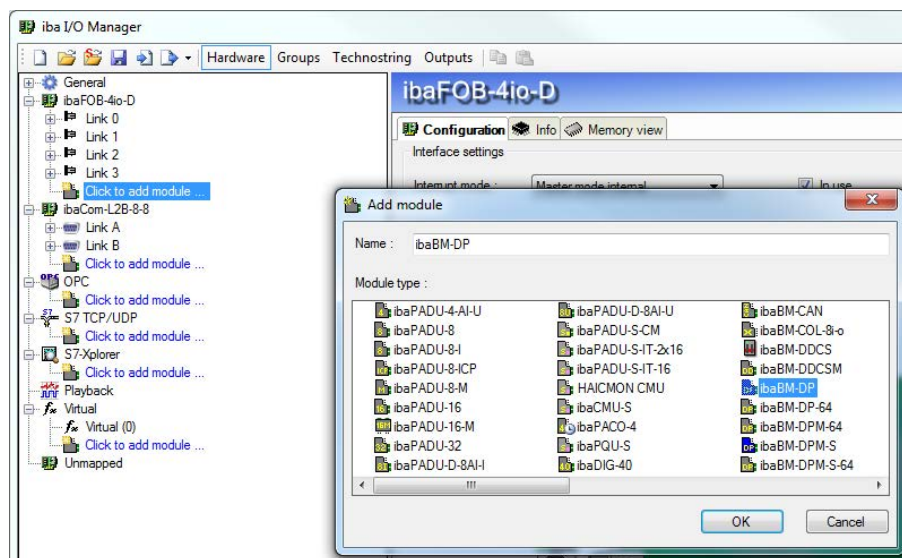
Other documentation



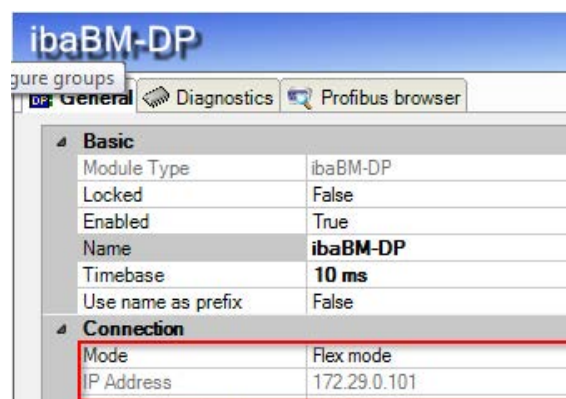
You can find more information about the *ibaFOB-D* card in the respective manual.

Configuring ibaBM-DP

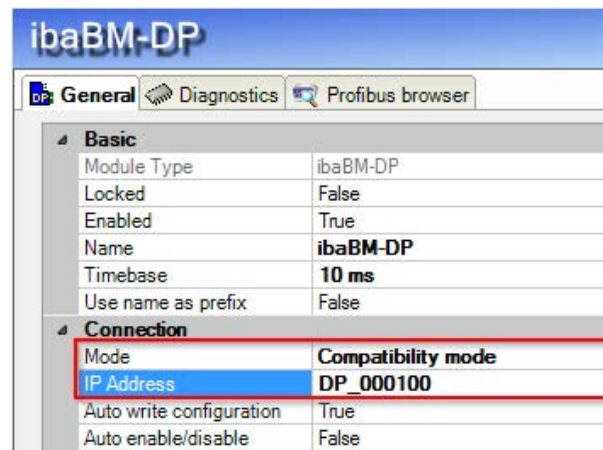
1. Start the *ibaPDA* client  and open the I/O Manager .
2. Mark in the interface tree (left) the link of the *ibaFOB* card to which *ibaBM-DP* is connected. Click on the link "Click to add module..." and select the *ibaBM-DP* module.



3. If you operate the device in Flex mode, the IP address of the device will be set automatically.



If you operate the device in *compatibility mode*, enter the IP address on the *General* tab of the *ibaBM-DP* module: either as name e.g. "DP_000100" or as IP address e.g. "192.168.11.123".



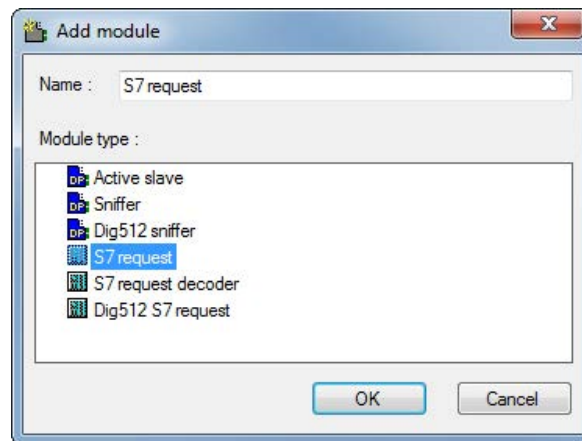
Note



The differences between Flex mode and compatibility mode are described in detail in the *ibaBM-DP* device manual.

An online connection to the device via Ethernet or USB is mandatory in compatibility mode. In the manual of the *ibaBM-DP* device, you can see how you can establish and check this connection.

4. Add a request module (or several modules, in case you need more connections to one or to different S7-CPU's) to the *ibaBM-DP* module on the respective PROFIBUS connection Bus 0 or Bus 1. Available modules are:
 - S7 request
(for acquiring analog and digital signals)
 - S7 request decoder
(for the acquisition of up to 1024 digital signals)
 - S7 request (ibaCom-L2B compatible)
(for acquiring analog and digital signals in ibaCom-L2B compatibility mode)
 - S7 request Dig512 (ibaCom-L2B compatible)
(for acquiring up to 512 digital signals in ibaCom-L2B compatibility mode)

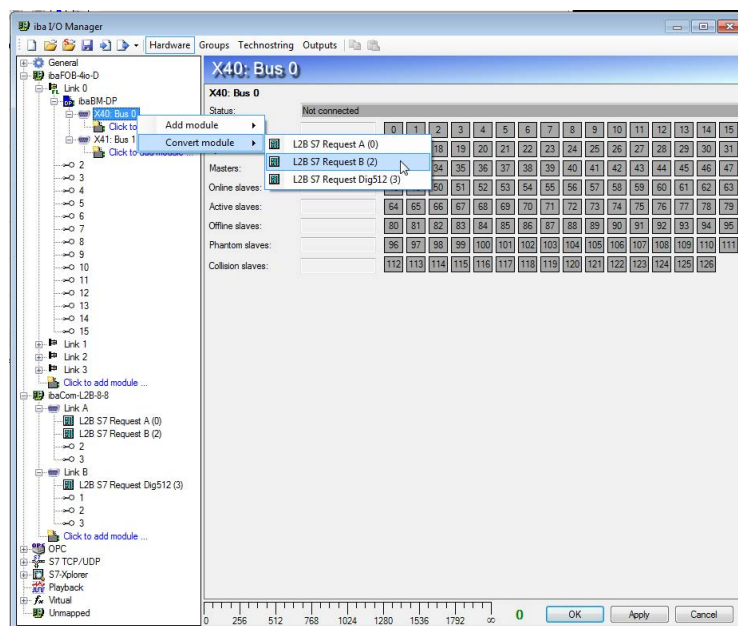


Note



If you want to migrate a Request-S7 solution based on an *ibaCom-L2B* card to an *ibaBM-DP* device, e.g. to replace an *ibaCom-L2B* card by an *ibaBM-DP* device, use the function in the I/O Manager for converting L2B Request-S7 modules.

Click with the right mouse button on the PROFIBUS links and select "Convert module" from the context menu. All available L2B Request-S7 modules are displayed. Select one by one the module, which should be converted to an *ibaCom-L2B* compatible module.



For more information about replacing an *ibaCom-L2B* card with an *ibaBM-DP* device, see chapter [Replacing Request-S7 on ibaCom-L2B by ibaBM-DP](#), page 166.

- Configure the required module settings and signals as described in the following chapters. The *General* and *Connection* tabs are identical for all Request modules. The Request modules only differ in the *Analog* and *Digital* tabs.

6. After you have finished the configuration, click <Apply> or <OK> to transfer the new configuration to the device and start data acquisition with *ibaPDA*.

4.3.4.2 General module settings

You find the description of all settings that are identical for all Request-S7 modules in chapter [↗ General interface settings](#), page 16.

The *ibaBM-DP* modules have the following common specific setting options:

Profibus

Profibus - Bus number

0 = connector X40 left, 1 = connector X41 right

Slave No.

PROFIBUS slave address assigned to the module.

Connection - Auto enable/disable

If TRUE, the acquisition is started, even if no connection can be established to the S7-CPU. The module is deactivated. During the acquisition, *ibaPDA* tries to reconnect to the S7-CPU. When it succeeds, the acquisition is restarted with this module enabled.

In case of FALSE, the acquisition is not started, if a connection to the configured S7-CPU cannot be established.

4.3.4.3 Connection settings

The connection settings are identical to those in chapter [↗ Connection settings](#), page 49.

4.3.4.4 Module S7 Request

With the "S7 request" module, up to 64 analog and 64 digital signals can be acquired.

A separate PROFIBUS slave and Request block call has to be configured for each module.

For a description of the module settings, see chapter [↗ General module settings](#), page 16 and [↗ General module settings](#), page 84.

4.3.4.5 Modules S7 Request Decoder

The "S7 Request Decoder" module allows the acquisition of up to 1024 digital signals sent as a maximum of 64 words (16 Bit). This module type is suited for applications where large amounts of digital signals have to be acquired and for which the max. 1024 directly addressable digital values of the *ibaBM-DP* are not sufficient.

A separate PROFIBUS slave and Request block call has to be configured for each module.

On the "Digital" tab, you can enter directly the words as basic signals for decoding over absolute S7 operands. Only word operands (e. g. PIW, MW, DBW) are allowed. It is also possible to use S7 symbols by generating address books. For detailed information, see chapter [↗ Selection via](#)

the operand symbols, page 20. The signals selected in S7 CFC and symbol browser are applied and the columns Name, S7 symbol, S7 Operand and Data Type are filled in automatically.

Click the <+> symbol to the left of each line to open a list in which you can enter a name for each of the 16 digital values.

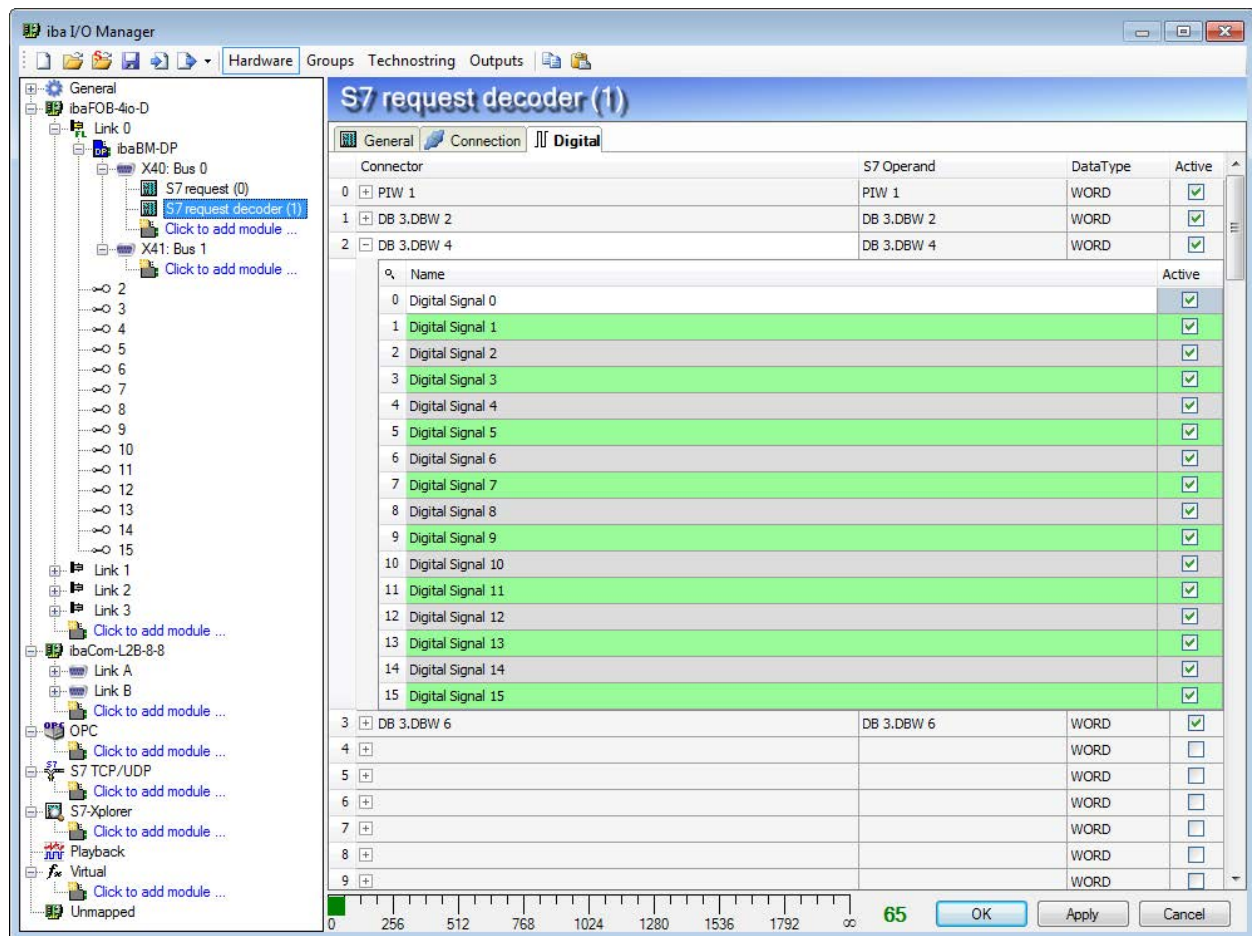


Fig. 40: Modules S7 Request Decoder

Note



The module type "Dig512 S7 Request" of the predecessor device *ibaBM-DPM-S* is not available any more for *ibaBM-DP*. Please use instead the "S7 Request Decoder" module type.

4.3.4.6 Module S7 Request (ibaCom-L2B compatible)

The module "S7 Request (ibaCom-L2B compatible)" is fully compatible to the module "L2B S7 Request" when using an ibaCom-L2B card (see chapter [↗ Configuration and engineering ibaPDA](#), page 129.)

The configuration on the SIMATIC S7 side has to be done according to chapter [↗ Configuration and engineering SIMATIC S7-300, S7-400 and WinAC](#), page 114.

Note

This module type usually is used to replace an existing Request-S7 solution based on an *ibaCom-L2B* card by an *ibaBM-DP* device. For more information about this application case, see chapter [↗ Replacing Request-S7 on ibaCom-L2B by ibaBM-DP](#), page 166.

Note

The module „S7 Request (ibaCom-L2B compatible)“ is not released for the connection to control systems of S7-1500 type.

4.3.4.7 Module S7 Request Dig512 (ibaCom-L2B compatible)

The module "S7 Request Dig512 (ibaCom-L2B compatible)" is fully compatible to the module "L2B S7 Request Dig512" when using an ibaCom-L2B card (see chapter [↗ Configuration and engineering ibaPDA](#), page 129.)

The configuration on the SIMATIC S7 side has to be done according to chapter [↗ Configuration and engineering SIMATIC S7-300, S7-400 and WinAC](#), page 114.

Note

This module type usually is used to replace an existing Request-S7 solution based on an *ibaCom-L2B* card by an *ibaBM-DP* device. For more information about this application case, see chapter [↗ Replacing Request-S7 on ibaCom-L2B by ibaBM-DP](#), page 166.

Note

The module „S7 Request Dig512 (ibaCom-L2B compatible)“ is not released for the connection to control systems of S7-1500 type.

4.3.5 Diagnostics

When you select the bus module node in the signal tree and choose the *Analog* or *Digital* tab, you get a list of all operands that have been acquired in the bus module with data type and actual value.

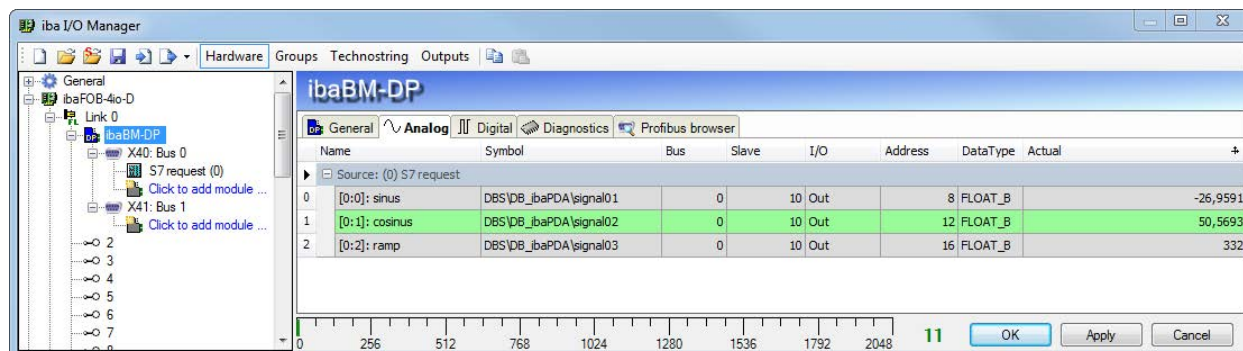


Fig. 41: List of the acquired operands on the Analog tab

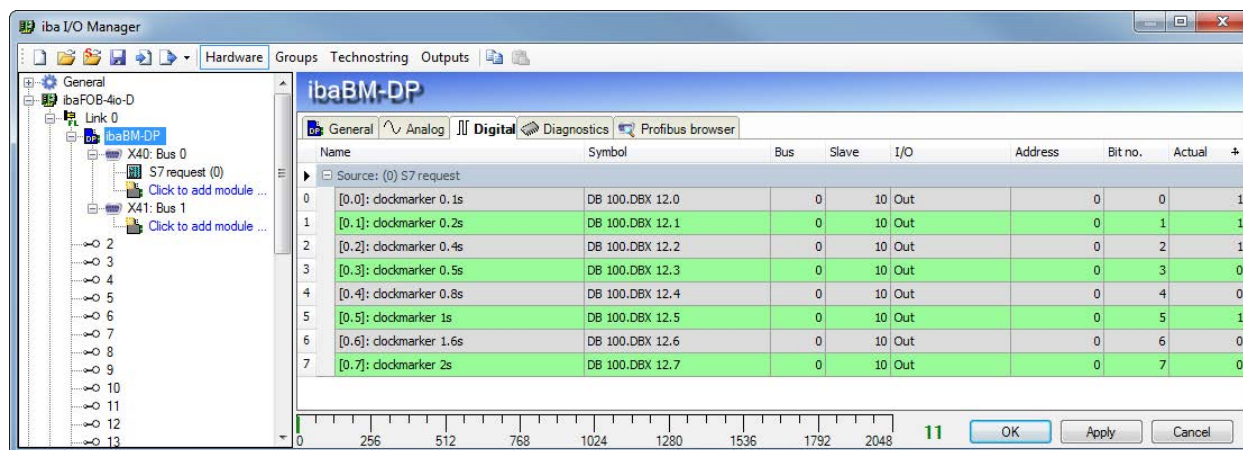


Fig. 42: List of the acquired operands on the Digital tab

Other documentation



You find a detailed description of the device-specific diagnostic options of *ibaBM-DP* in the device manual.

4.4 Request-S7 for ibaBM-DP in redundancy mode

In the following, the Request-S7 version for the PROFIBUS bus module *ibaBM-DP* in redundancy mode is described.

4.4.1 General information

With the redundancy mode of *ibaBM-DP*, the device can be operated on redundant PROFIBUS systems in combination with SIMATIC-400H controllers, whose measurement data are to be acquired.

You need an additional license for using the redundancy mode of the *ibaBM-DP*. Please contact the iba AG support team. The license is released using the administrator functions in the Web dialog of the *ibaBM-DP*.

Request-S7 redundant is suitable for operating *ibaBM-DP* with the Request functionality as single-channel periphery on a highly available SIMATIC S7-400H controller.

The following figure shows the exemplary integration of an *ibaBM-DP* in redundancy mode.

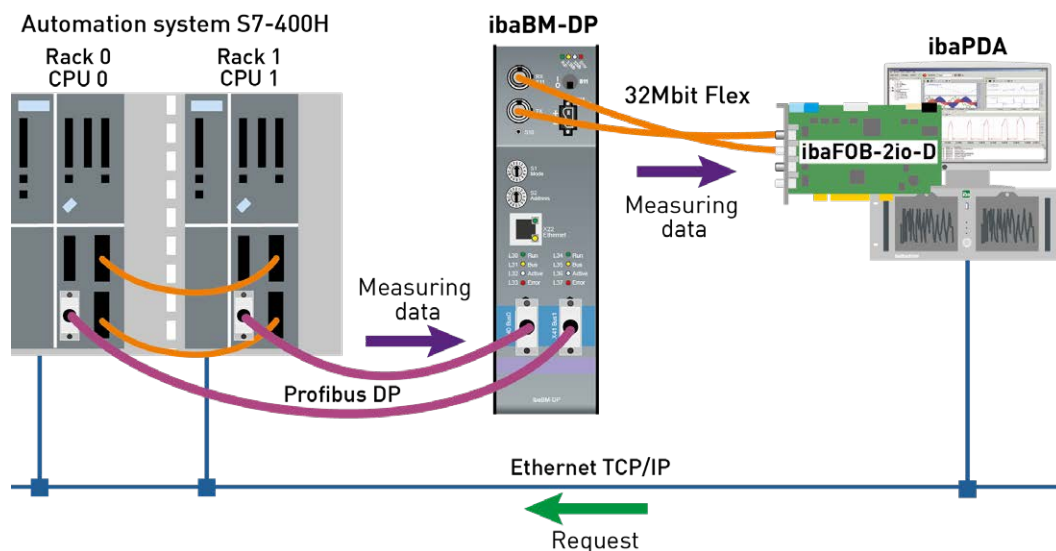


Fig. 43: Request-S7 for ibaBM-DP in redundancy mode

Note



The following description refers to the Request blocks V4.0 or higher.

Other documentation



For detailed information about the redundancy mode of the *ibaBM-DP*, please see the device manual.

The functionality of the Request-S7 for *ibaBM-DP* in redundancy mode mostly corresponds to the functionality in standard mode (see chapter [Request-S7 for ibaBM-DP](#), page 64). The differences and extensions are described in the following.

For information and application examples, see chapter [Application examples](#), page 148.

4.4.2 Configuration and engineering SIMATIC S7-300, S7-400 and WinAC

4.4.2.1 Description of the Request-FC ibaDP_Req_H (FC123)

With this function, the communication between the S7, *ibaPDA* and the *ibaBM-DP* device is initialized and controlled.

The ibaDP_Req_H function has to be called once per slave pair in the cyclic program.

The used blocks are part of the iba S7 library (see chapter [Iba S7 library](#), page 139).

Description of the formal parameters of the ibaDP_Req_H

Name	Type	Data Type	Description
DB_PDA	IN	BLOCK_DB	DB of the baPDA communication interface ibaDP_DB_PDA (2064 Byte)
DB_INTERN	IN	BLOCK_DB	DB of the internal data interface ibaDP_DB_work (1900 Byte)
OUTPUT_ADR_SLAVE	IN	INT	Start address of the peripheral output range, continuous and 244 Bytes long
INIT_FC	IN	BOOL	TRUE -> Initialize
INP_RANGE	IN	INT	Number of input bytes (evaluation only during initialization), 0:automatic detection (recommended)
OUT_RANGE	IN	INT	Number of output bytes (evaluation only during initialization), 0:automatic detection (recommended)
MARKER_RANGE	IN	INT	Number of marker bytes (evaluation only during initialization), 0:automatic detection (recommended)
ERROR_STATUS_INIT	OUT	BYTE	Initialization error
ERROR_STATUS_COM	OUT	BYTE	Communication error

The following SFCs are used internally:

- SFC 13 (DPNRM_DG)
- SFC 15 (DPRD_DAT)
- SFC 20 (BLKMOV)
- SFC 21 (FILL)
- SFC 24 (TEST_DB)

- SFC 49 (LGC_GADR)
- SFC 50 (RD_LGADR)
- SFC 51 (RDSYSST)

Detailed description:**DB_PDA**

DB in the range of 1 to n (see technical data of the CPU). The DB is used for data exchange with *ibaPDA*. Length at least 2064 Bytes.

DB_INTERN

DB in the range of 1 to n (see technical data of the CPU).
Length at least 1900 Bytes.

The following data are stored in this DB:

- Technical data of CPU which are determined during initialization
- The S7 pointers requested by *ibaPDA*
- The binary and analog transmission data

OUTPUT_ADR_SLAVE

DP start address of the *ibaBM-DP* in the peripheral output range for sending data. Length 244 Bytes, continuous (without gaps!)

INIT_FC

Used for initializing the Request block. The Request block is initialized automatically internally. Additionally, the initialization can be performed manually by an external logic via the input `INIT_FC = TRUE`.

INP_RANGE

Limits the number of input bytes to be measured.

If `INP_RANGE = 0`, the size of the available process image of the inputs will be evaluated by the Request block (recommended). Evaluation is done only during the initialization phase of the module.

OUT_RANGE

Limits the number of output bytes to be measured.

With `OUT_RANGE = 0` the size of the available process image will be evaluated by the Request block (recommended). Evaluation is done only during the initialization phase of the module.

MARKER_RANGE

Limits the number of marker bytes to be measured.

With `MARKER_RANGE = 0`, the number of the available markers will be evaluated by the Request block (recommended). Evaluation is done only during the initialization phase of the module.

ERROR_STATUS_INIT

The following error codes can be generated:

- 1: DB_PDA is read-only
- 2: DB_PDA-No = 0 or higher than the max. permissible DB number of this CPU

- 3: DB_PDA with the specified number is not available
- 5: DB_PDA is too short
- 11: DB_INTERN is read-only
- 12: DB_INTERN-No = 0 or higher than the max. permissible DB number of this CPU
- 13: DB_INTERN with the specified number is not available
- 15: DB_INTERN is too short
- 16: Error when reading the identification data of the CPU
- 19: Initialization not finished
- 21: Not enough memory space for the data set
- 22: SZL_ID is wrong or not known in this CPU
- 23: The index of the SZL is wrong or not permitted
- 30: OUT_ADR_SLAVE is no peripheral output range
- 31: OUT_ADR_SLAVE is not assigned to a PROFIBUS
- 32: RM⁴⁾: The configured "OUTPUT_ADR_SLAVE_BUS_0" is incorrect
- 33: RM: The configured "OUTPUT_ADR_SLAVE_BUS_0" is not assigned to a PROFIBUS DP Slave.
- 34: RM: The configured "OUTPUT_ADR_SLAVE_BUS_1" is incorrect.
- 35: RM: The configured "OUTPUT_ADR_SLAVE_BUS_1" is not assigned to a PROFIBUS DP Slave.
- 36: RM: SLAVE BUS0 and SLAVE BUS1 do not have the same DP address

ERROR_STATUS_COM

The following error codes can be generated:

- 100: Bit number not 0
- 101: Bit number not 0-7
- 103: The operand range is not defined.
- 104: The data type is not defined
- 105: DB0 has been requested as data source
- 106: DB number is higher than the max. permitted number of DBs for this CPU
- 107: DB with the specified number is not available
- 109: DB is too short for the accessed operand
- 110 Specified data address does not exist
- 111: Initialization aborted with error
(error cause can be seen in ERROR_STATUS_INIT)
- 112 Initialization not carried out
(error cause can be seen in ERROR_STATUS_INIT)
- 150: Order fragmentation is not supported
- 151: Order Request: Total number of values does not match values in the order
- 152: Order Request: Up to 64 binary values permitted
- 153: Order Request: Up to 64 analog values permitted
- 200: DP station is not available
- 201: RM⁵⁾: Slave Bus 0 not available
- 202: RM: Slave Bus 1 not available
- 203: RM: Slaves Bus 0 + 1 are not available
- 210: Output modules of the Slaves Bus 0 and 1 are configured differently

⁴⁾ RM: Redundancy mode

⁵⁾ RM: Redundancy mode

4.4.2.2 Hardware configuration

For each PROFIBUS master, a separate ibaBM-PROFIBUS slave has to be configured.

The GSD file "ibaDPMSi.gsd" has to be used (version V2.2 and higher).

Note



You find the GSD-file "ibaDPMSi.gsd" on the DVD "iba Software & Manuals" in the following directory:

\02_iba_Hardware\ibaBM-DP\02_GSD_Files\01_General\

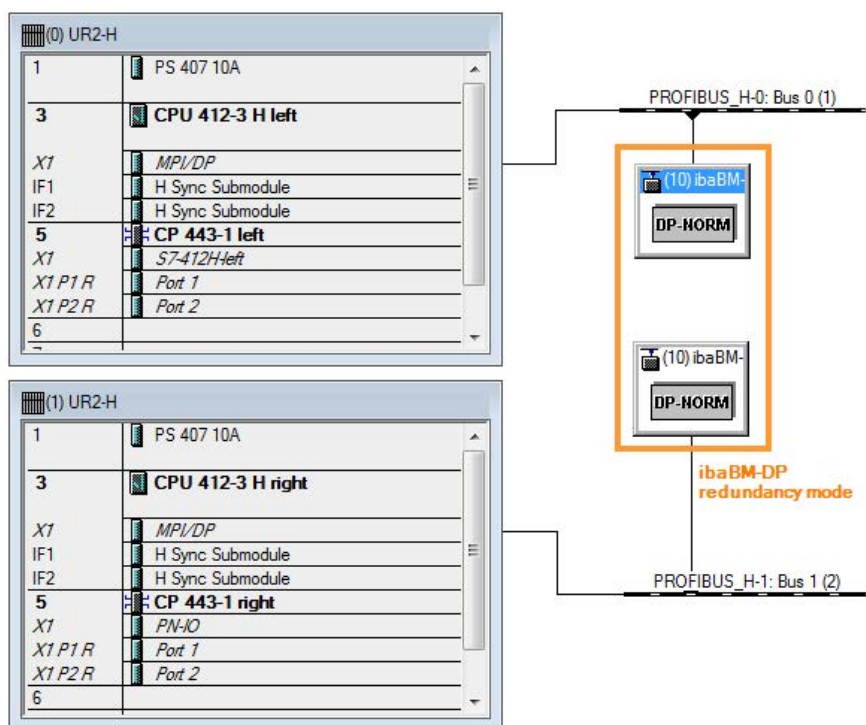


Fig. 44: HW config redundant system

Both configured slaves have to be configured with identical bus addresses. The I/O address ranges are different. Use the modules "S7-Req block consistent Slot 0 / Slot 1".

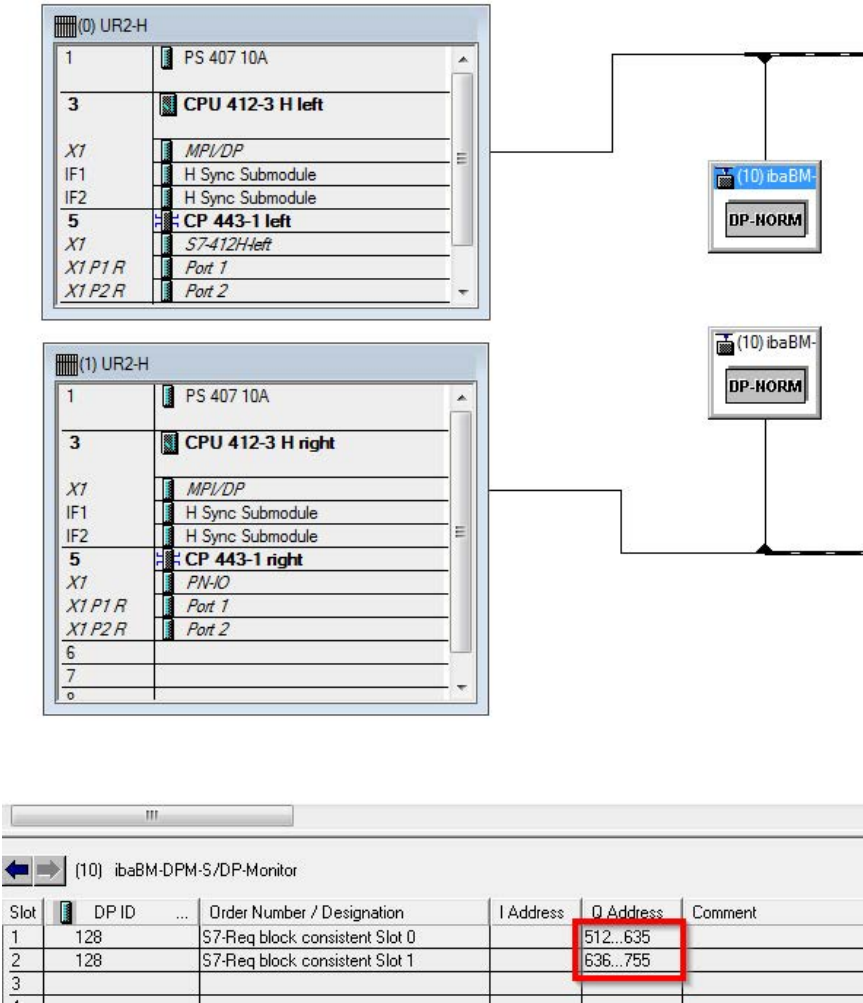


Fig. 45: Slave IO addresses Bus 0

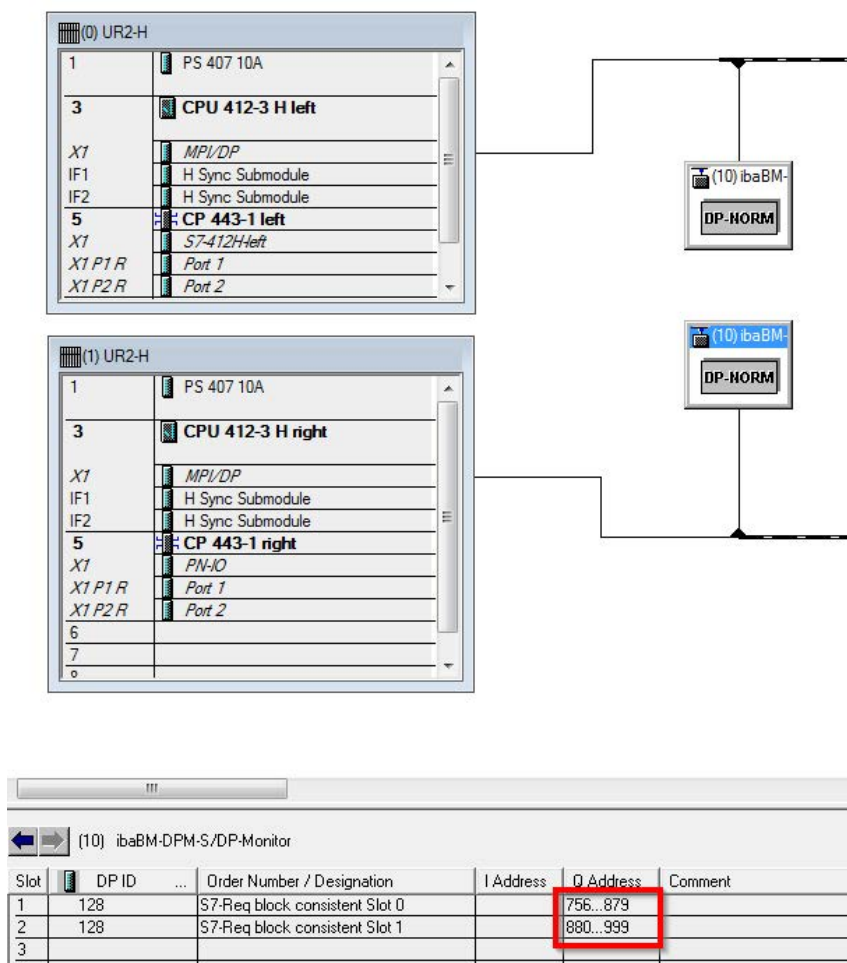


Fig. 46: Slave IO addresses Bus 1

Note

The modules have to be created for each slave without gaps and with sequential start addresses.

4.4.2.3 Configuration in STEP 7 (STL, LAD, FBD)

Request-S7 redundant is intended for the use with S7-400H CPUs with integrated DP interface as well as with external interface CP 443-5 (PROFIBUS master).

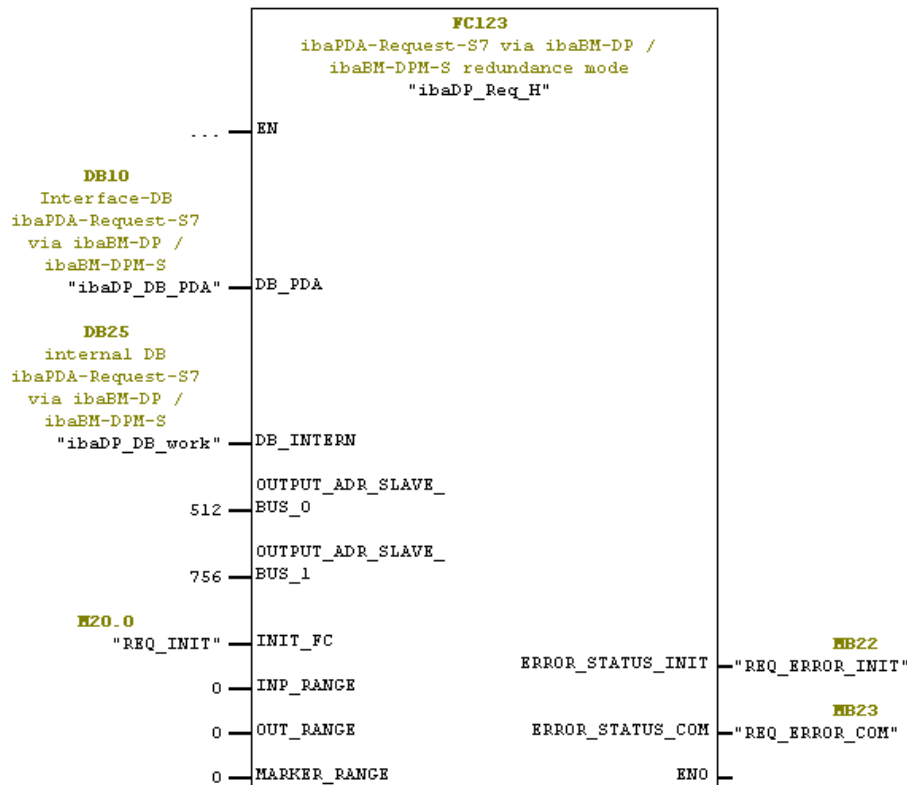
For Request-S7 with one PROFIBUS slave:

- Copy the following blocks from the iba S7 library (see chapter [Iba S7 library](#), page 139) to the blocks folder of your STEP 7 project:
 - FC123 (ibaDP_Req_H)
 - DB10 (ibaDP_DB_PDA)
 - DB25 (ibaDP_DB_work).

Note

If the block numbers FC123, DB10 and DB25 are already used in your project, assign other free numbers to the blocks from the iba S7 library when copying.

- Create the error OBs (OB82, OB85, OB86, OB87, OB122) in order to prevent CPU stops in case of an error.
- Call and parametrize the ibaDP_Req_H (FC123) in the cyclic program.

**For Request-S7 with multiple PROFIBUS slaves:**

- In the blocks folder, a data block ibaDP_DB_PDA (DB10) has to be available for each slave pair. Copy the DB10 in a DB with a new DB number.
- In the blocks folder, a data block ibaDP_DB_work (DB25) has to be available for each slave pair. Copy the DB25 in a DB with a new DB number.
- In the cyclic program, another call of the ibaDP_Req_H (FC123) with the new DB numbers and the respective peripheral addresses of the new PROFIBUS slave pair has to be done for each slave pair.
- The assignment of markers to the input INIT_FC and the outputs ERROR_STATUS_INIT or ERROR_STATUS_COM (or DB elements) should be unambiguous for each slave pair.

Finish:

- Load all blocks into the S7-CPU and restart.

4.4.3 Configuration and engineering ibaPDA

First of all, connect *ibaBM-DP* to a free link of an *ibaFOB-D* card. Insert a device module *ibaBM-DP* on the respective link in the I/O Manager.

Set the "Redundancy mode" to TRUE on the *General* tab. The activated redundancy mode is displayed by an orange colored symbol of the device module.

The device can be operated in the 32Mbit Flex mode as well as in compatibility mode with 32Mbit.

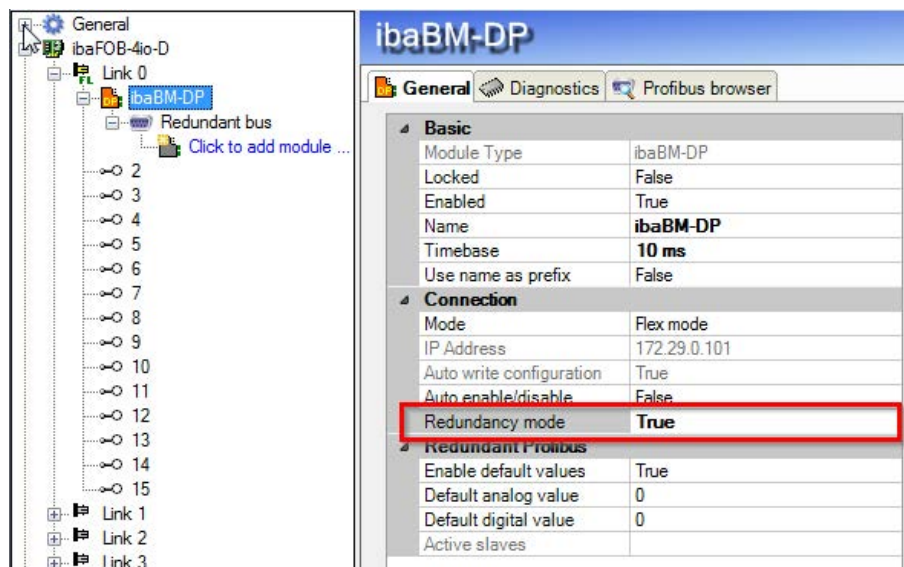


Fig. 47: Setting redundancy mode in I/O Manager

As on the redundant PROFIBUS both bus systems 0 and 1 are operated in parallel, there are only settings for the "Redundant Profibus" on the *General* tab.

In redundancy mode, the following modules are available with Request-S7:

- S7 Request
- S7 Request Decoder

Note



The following modules cannot be used in redundancy mode:

- Module S7 Request (ibaCom-L2B compatible)
- Module S7 Request Dig512 (ibaCom-L2B compatible)

These modules are deactivated automatically and cannot be activated manually, either.

The configuration of the modules corresponds to the configuration in standard mode. For more information, see chapter [General module settings](#), page 84.

Connection settings

For transmitting the operand data to both CPUs of the SIMATIC S7-400H, two different connections are established. These connections are switched on and off depending on availability and requirements.

Both connections are configured on the tabs *Connection 0* and *Connection 1*. The name of the tabs can be assigned in the "Connection name" fields.

Fig. 48: Connection settings for connection 0

Fig. 49: Connection settings for connection 1

The other settings are identical to the settings for operation without redundancy mode (see chapter [➤ Connection settings](#), page 84).

The following special features have to be considered:

- The rack numbers for an H system are 0 or 1 for the both redundant CPUs.
- The DB number, the PROFIBUS slave number and the CPU name for assigning an address book is set only once on the *Connection 0* tab.

4.5 Request-S7 for ibaBM-DPM-S

In the following, the Request-S7 version for the PROFIBUS bus module *ibaBM-DPM-S* is described.

The solution "Request-S7 for ibaBM-DPM-S" is replaced by "Request-S7 for ibaBM-DP" with full functional compatibility.

4.5.1 General information

For DP Request, the measurement values (Request Handshake) are not requested via the PROFIBUS, but via a separate connection.

Depending on the hardware and software, different access points can be selected for the Request:

- TCP/IP: the connection to the SIMATIC S7 is established over an integrated PN interface of the S7-CPU or the respective CP modules in the PLC and the standard network interface of the PC. No additional Siemens software is required the connection.
- PC/CP: this is the designation for different SIMATIC specific access points. In contrast to the TCP/IP connection, the SIMATIC communication software (and the corresponding licenses) must be installed on the PC for all connection types within the PC/CP group.
 - MPI, PROFIBUS: The connection to the SIMATIC S7 is established via the MPI or PROFIBUS interface of the PC; e.g. with the PCI card CP5611 or the MPI adapter for USB or serial PC interface.
 - TCP/IP, ISO: here, either the standard network interface of the PC or a suitable interface card is used for the connection to the S7.

System integration with ibaBM-DPM-S

The measured data is transmitted to the *ibaBM-DPM-S* device over PROFIBUS DP.

You need the following connections:

- Online connection between *ibaPDA* and S7-CPU (TCP/IP, MPI or DP)
- Online connection between *ibaPDA* and *ibaBM-DPM-S* (TCP/IP over Ethernet or USB)
- Fiber optic connection between *ibaPDA/ibaFOB-i-D* and *ibaBM-DPM-S*
- PROFIBUS connection between *ibaBM-DPM-S* and S7 PROFIBUS master

In the standard version a maximum of eight connections, i.e. eight PROFIBUS slaves, can be configured per device. Up to 244 Byte data can be transmitted per slave.

The following illustration shows the configuration using a TCP/IP online connection between *ibaPDA* and S7-CPU.

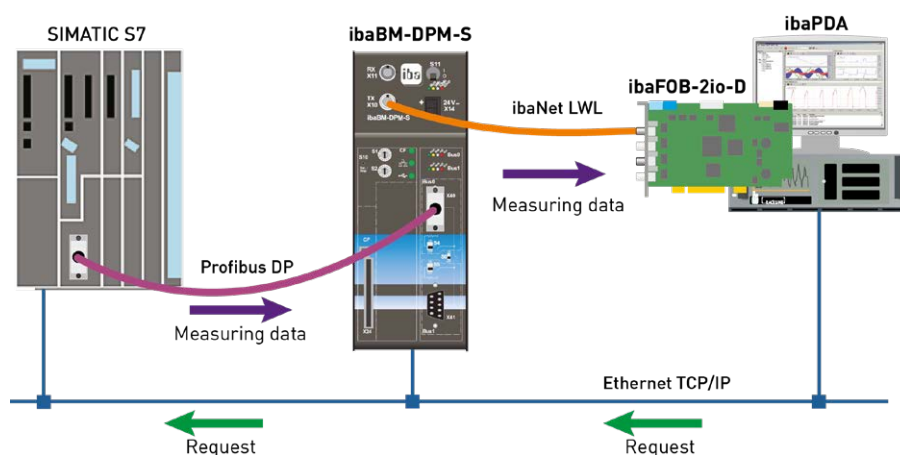


Fig. 50: Request-S7 for ibaBM-DPM-S

Note

The following description refers to the Request blocks V4.0 or higher. Should you require information regarding older versions, please contact our support.

Other documentation

For detailed information about *ibaBM-DPM-S*, see the device manual.

For information and application examples, see chapter ➤ *Application examples*, page 148.

4.5.2 Configuration and engineering SIMATIC S7-300, S7-400 and WinAC

The configuration on the SIMATIC S7 side corresponds to the configuration of *ibaBM-DP* (see chapter ➤ *Configuration and engineering SIMATIC S7-300, S7-400 and WinAC*, page 66).

4.5.3 Configuration and engineering ibaPDA

4.5.3.1 General interface settings

The interface is configured in the *ibaPDA* "I/O Manager". Prerequisite is the installation of an ibaFOB-D card in the ibaPDA computer.

After the *ibaFOB-D* card has been installed successfully, it is displayed automatically in the interface tree.

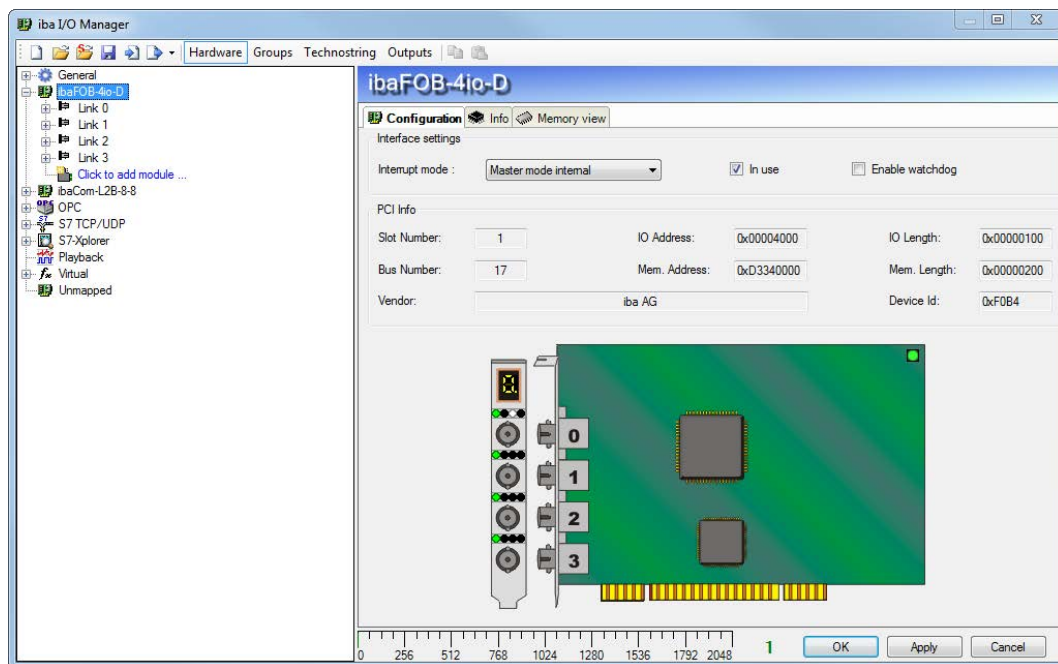




Fig. 51: I/O Manager, display ibaFOB-D card

Other documentation



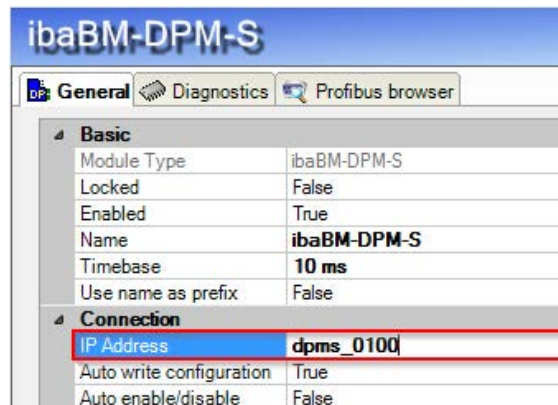
For detailed information about the *ibaFOB-D* card, please see device manual.

Configuring ibaBM-DPM-S

1. Start the *ibaPDA* client  and open the I/O-Manager .
2. Mark the link of the *ibaFOB* card in the interface tree (left), to which *ibaBM-DPM-S* is connected. Click with the right mouse-button on the "Add module..." link and select the *ibaBM-DPM-S* module.



3. Enter the IP address of the ibaBM-DPM-S device on the *General* tab of the *ibaBM-DPM-S* module: either as name e.g. "dpms_0100" or as IP address e.g. "192.168.11.123".

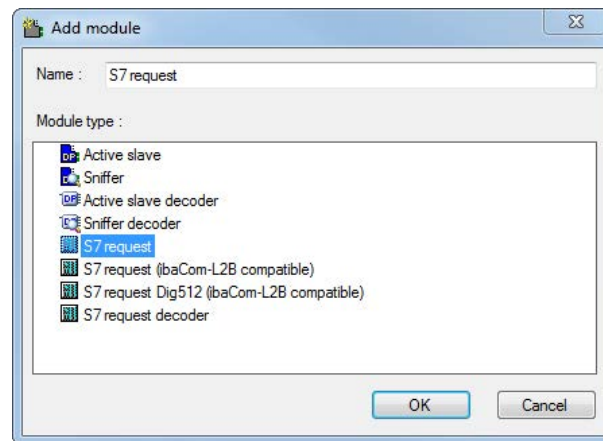


Note



An online connection to the device via Ethernet or USB is mandatory. How to establish and test such a connection is described in the manual of the *ibaBM-DPM-S* device.

4. Add an S7 request module to the *ibaBM-DPM-S* module (or several if you need more connections to one or multiple S7-CPU's). Available modules are:
 - S7 request (for acquiring analog and digital signals)
 - S7 request decoder (for acquiring up to 1024 digital signals)
 - Dig512 S7 request (for acquiring up to 512 digital signals)



5. Configure the required module settings and signals as described in the following chapters. The *General* and *Connection* tabs are identical for all Request modules. The Request modules only differ in the *Analog* and *Digital* tabs.
6. After you have finished the configuration, click <Apply> or <OK> to transfer the new configuration to the device and start data acquisition with *ibaPDA*.

4.5.3.2 General module settings

You find the description of all settings that are identical for all Request-S7 modules in chapter [↗ General module settings](#), page 16.

The *ibaBM-DPM-S* modules all have the following common specific settings options.

Profibus

Profibus - Bus number

0 = connector X40 top, 1 = connector X41 bottom

Slave No.

PROFIBUS slave address assigned to the module.

Connection - Auto enable/disable

If TRUE, the acquisition is started, even if no connection can be established to the S7-CPU. The module is disabled. During the acquisition, *ibaPDA* tries to reconnect to the S7-CPU. When it succeeds, the acquisition is restarted with this module enabled.

In case of FALSE, the acquisition is not started, if a connection to the configured S7-CPU cannot be established.

4.5.3.3 Connection settings

On the *Connection* tab, the corresponding connection of the Request module to the controller is configured.

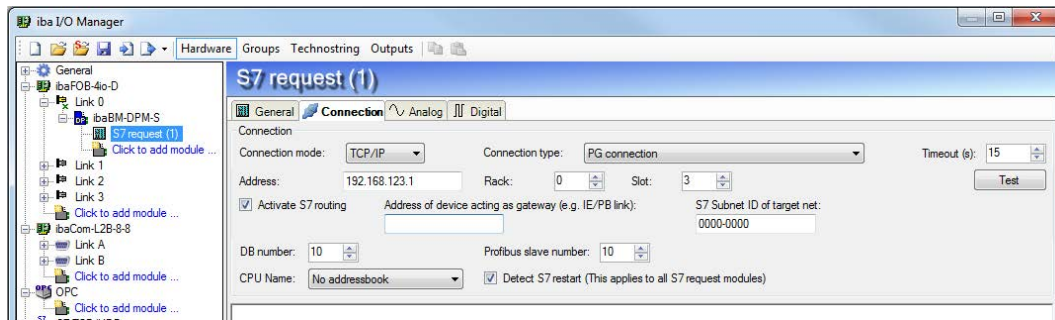


Fig. 52: Connection setting, Connection mode TCP/IP

On the *Connection* tab of the Request modules, you have to enter the connection parameters. With the <Test> button you can test the connection.

Selection of connection mode TCP/IP

Connection type

PG, OP connection or other

Timeout

15 s (default)

Address

IP address of the S7-CPU or the Ethernet-CP, e.g. "192.168.50.68"

Rack

Rack number of the S7 station, e.g. "0"

Slot

Slot of the CPU in the rack, e.g. "3"

Activate S7 routing

(also see chapter [S7 routing](#), page 157)

Activate this option, if the S7-CPU and the *ibaPDA*-PC are not in the same network, but only communicate over a gateway that supports S7 routing. Such a gateway can be e. g. an IE/PB link, over which a S7-CPU can be reached without an Ethernet connection.

Two additional input fields appear:

- Address of the device acting as gateway: Enter IP address of the gateway
- S7 Subnet ID of target net: Enter subnet ID from STEP 7 NetPro

DB number

Number of the DB (*ibaDP_DB_PDA*), which is assigned to this connection, e.g. "10"

Profibus slave number

Slave number from the *General* tab. If you change the entry on the *Connection* tab, the entry changes accordingly on the *General* tab and vice versa.

CPU Name

CPU name from the *General* tab. If you change the entry on the *Connection* tab, the entry changes accordingly on the *General* tab and vice versa.

Note



ibaPDA uses the following TCP port to communicate with *ibaBM-DPM-S* via Ethernet; it must be enabled in the firewall:

ibaPDA-PC --> *ibaBM-DPM-S*, port 999.

The S7-Request function requires the following port to be enabled, too.

ibaPDA-PC --> S7-CPU, Port 102.

Selection of connection mode PC/CP

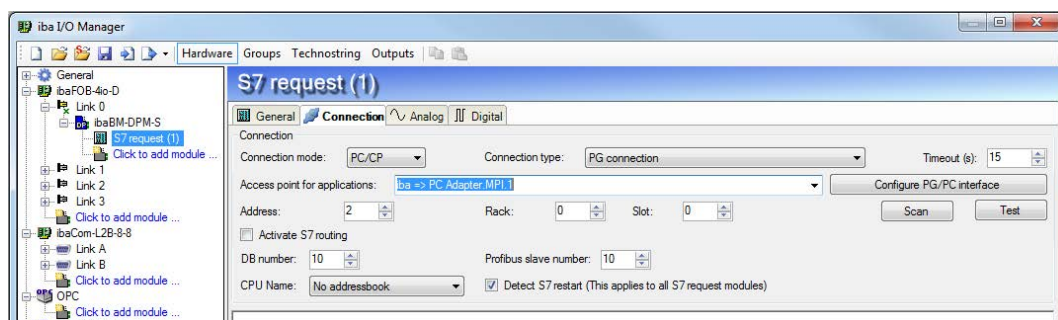


Fig. 53: Connection settings, connection mode PC/CP

Connection type

PG, OP connection or other

Timeout

15 s (default)

Access point for applications

Selection of an access point, e.g. S7 online, preferably create an own access point.

Address

MPI or DP address of the S7 CPU, e.g. "2" or
for ISO MAC address of the Ethernet interface, e.g. "08-00-06-01-00-00"

Rack

Rack number of the S7 station, e.g. "0"

Slot

Slot of the CPU in the rack. For MPI/DP as transfer medium, use "0" as slot. Addressing is done via the MPI/DP address only. With ISO/TCP, however, the slot in the rack must be used.

S7 routing, DB number, Profibus slave number, CPU name

see "Connection mode TCP/IP", and chapter [S7 routing](#), page 157.

<Configure PG/PC interface>

Here, you configure the assignment between access point and physical interface. Additionally, you can create an own access point for *ibaPDA* here (also see chapter [➤ Setting PG/PC interface / defining new access point](#), page 153)

Note



In order to use the connection type PC/CP, the SIMATIC communication software (e.g. SIMATIC Manager or driver for DP/MPI adapter) must be installed on the *ibaPDA* computer.

iba recommends generally setting up a special access point for *ibaPDA* for the connection to SIMATIC S7 via PC/CP connections. A separate access point prevents the risk that access for *ibaPDA* is impaired when changing the standard access point S7ONLINE in SIMATIC Manager.

Testing the connection

If you do not know the rack or slot number, you can click on the <Test> button to search the rack for CPUs (rack and slot are both set to "0").

As a result, the MLFB numbers of the available CPU modules are listed and displayed as blue hyperlinks.

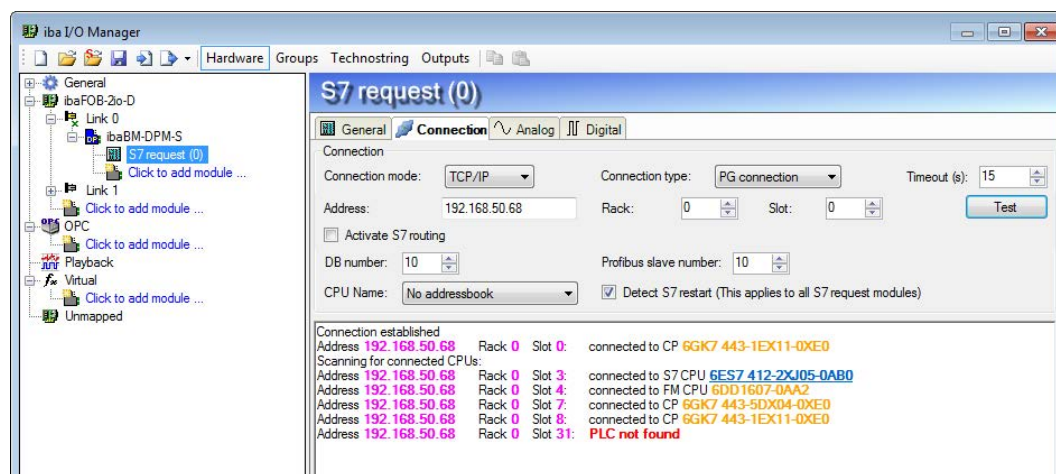


Fig. 54: Testing the connection

By clicking on the hyperlink of a CPU, the connection will be established and the rack and slot number will be filled in automatically. If the right DB number is entered, the connection to the Request agent is established immediately and the PROFIBUS diagnostic data that are entered in the DB are read and displayed.

Diagnostic information

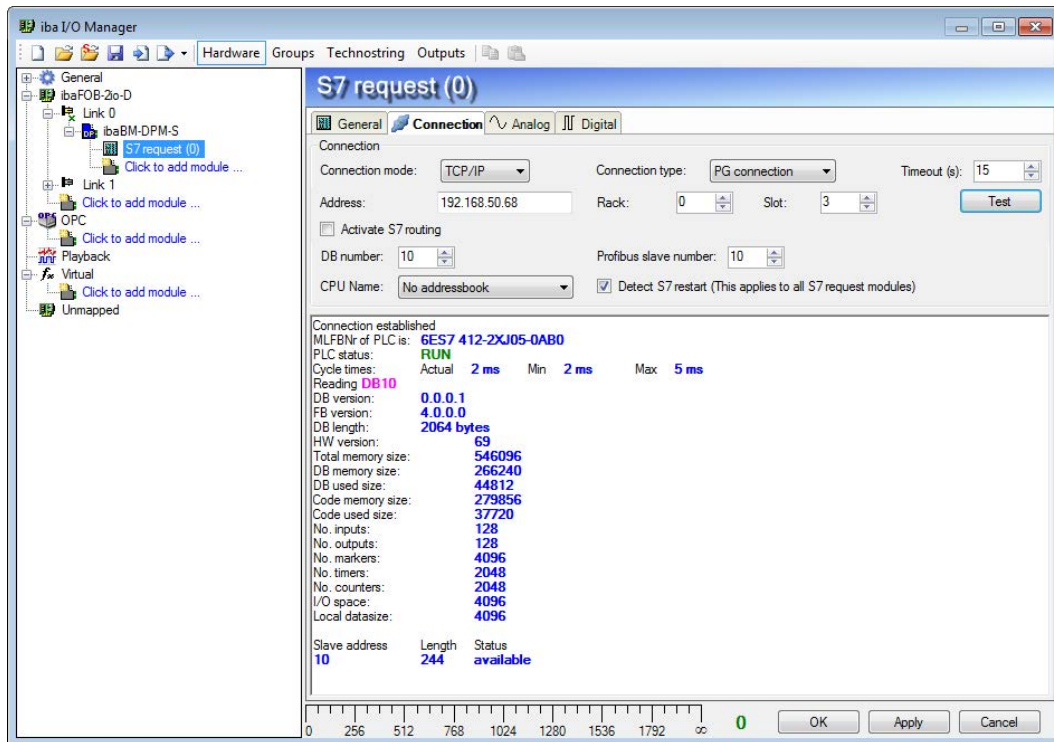


Fig. 55: Diagnostic information

The following diagnostic data are available in the DB:

- PLC status: for example, "RUN" or "STOP"
- Cycle time: current, minimum and maximum value of OB1 cycle
- DB version: version code, for example, "0.0.0.1"
- FB version: version code, for example, "4.0.0.0"
- DB Length: length of the communication DB (must be 2064 bytes)
- HW version: release of CPU
- Total memory size
- DB memory
- DB used
- memory size for code
- used code
- number of inputs
- number of outputs
- number of markers
- number of timers
- number of counters

- IO area
- local data
- Slave address: slave address determined by the start address of the peripheral output range (see input OUTPUT_ADR_SLAVE of the FC122 Request block, chapter [➤ Description of the Request-FC ibaDP_Req \(FC122\)](#), page 67).
- Length: length of the slave's output range. The length must be 244 bytes and match the slave length configured in HW Config.
- Status: indicates whether the slave is recognized by the CPU or not (unavailable/available). When setting up the S7 request module for the first time, the status is "not available", since the corresponding slave on the *ibaBM-DPM-S* is activated only when the configuration is applied in the I/O Manager by clicking <OK> or <Apply>.

If no connection is established to the S7-CPU, this might have the following causes:

- no TCP/IP connection possible (time out error), test network connection and IP address of controller and *ibaPDA* PC
- wrong address, rack or slot number
- wrong DB number
- DB too small (less than 2064 bytes)
- Error in the FC122 configuration; then an error number is issued, that corresponds to the connector ERROR_STATUS_COM - of the FC122.
- An initialization error of the FC122 is displayed on the function block connector ERROR_STATUS_INIT.

The PROFIBUS slave number is read from the diagnostics information and applied automatically to the settings of the S7 request module.

Note



For some older S7 CPU models the automatic detection of the station number is not possible. Then the slave number 0 is indicated in the diagnostic information. In this case, the user must manually enter the station number configured in the HW Config into the field "Slave number" under "General / PROFIBUS".

4.5.3.4 Module S7 Request

With the "S7 request" module, up to 64 analog and 64 digital signals can be acquired.

A separate PROFIBUS slave and Request block call has to be configured for each module.

For a description of the module settings, see chapter [➤ General module settings](#), page 16 and [➤ General module settings](#), page 102.

4.5.3.5 Modules S7 Request Decoder

With the modules "S7 Request Decoder" up to 1024 digital signals can be acquired that are sent in form of up to 64 words. This module type is suited for applications where large amounts of digital signals have to be acquired and for which the max. 512 directly addressable digital values of the *ibaBM-DPM-S* are not sufficient.

For a detailed description of the "S7 Request Decoder" module, please see the module description of the *ibaBM-DP* device in chapter [➤ Modules S7 Request Decoder](#), page 84.

4.5.3.6 Module Dig512 S7 Request

For the "Dig512 S7 request" module the same applies as for the module "S7 Request Decoder". But only 32 words (512 digital values) can be acquired.

The "Dig512 S7 Request" module is the predecessor of the „S7 Request Decoder" module and is still supported in *ibaPDA* for reasons of downward compatibility.

For new configurations, use the "S7 Request Decoder" module (see chapter [➤ Modules S7 Request Decoder](#), page 108).

4.5.4 Diagnostics

When you select the bus module node in the signal tree and choose the *Analog* or *Digital* tab, you get a list of all operands that are acquired in the bus module with data type and actual value.

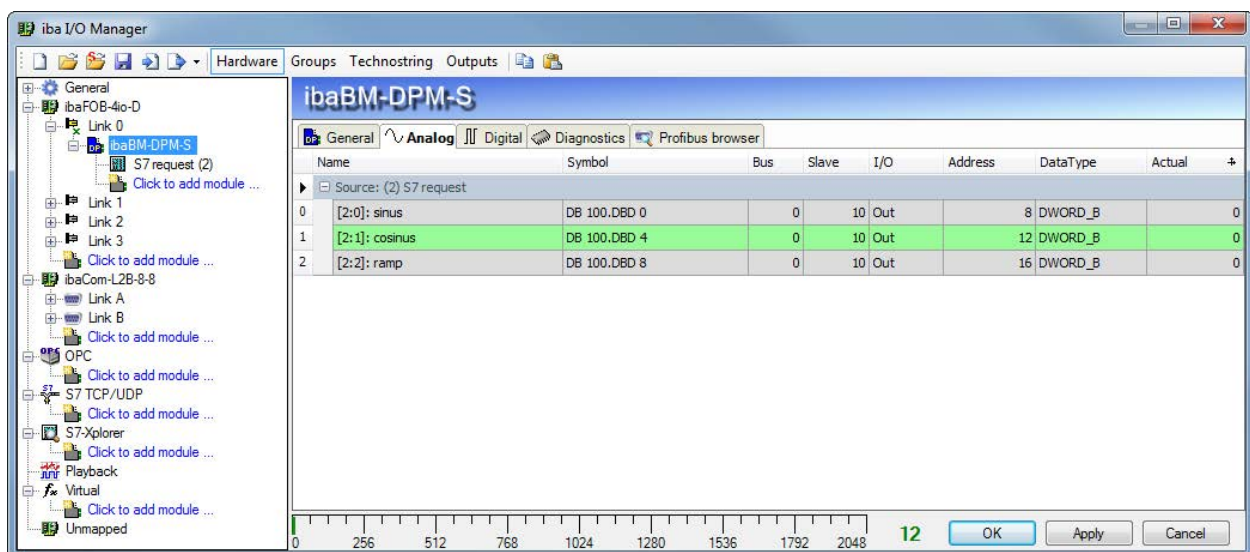


Fig. 56: List of the acquired operands on the Analog tab

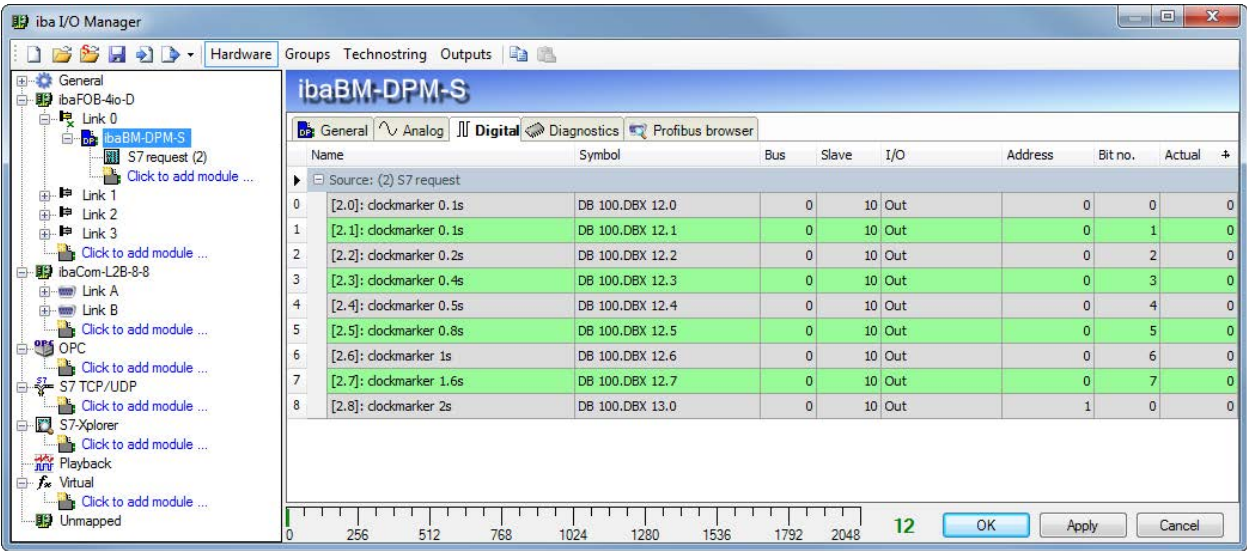


Fig. 57: List of the acquired operands on the Digital tab

Other documentation



You find a detailed description of the device-specific diagnostic options of *ibaBM-DPM-S* in the device manual.

4.6 Request-S7 for ibaBM-DPM-S in redundancy mode

In the following, the Request-S7 version for the PROFIBUS bus module *ibaBM-DPM-S* in redundancy mode is described. The solution "Request-S7 for ibaBM-DPM-S in redundancy mode" is replaced by "Request-S7 for ibaBM-DP in redundancy mode" with full functional compatibility.

4.6.1 General information

With the redundancy mode of *ibaBM-DPM-S*, the device can be operated on redundant PROFIBUS systems in combination with SIMATIC S7-400H controllers, whose measurement data are to be acquired. You need an additional license for using the redundancy mode of *ibaBM-DPM-S*. Please, contact the iba AG support team. The license is released using the administrator functions in the Web dialog of the *ibaBM-DPM-S*.

Request-S7 redundant is suitable for operating *ibaBM-DPM-S* with the Request functionality as single-channel periphery on a highly available SIMATIC S7-400H controller. The following figure shows the integration of an *ibaBM-DPM-S* in redundancy mode.

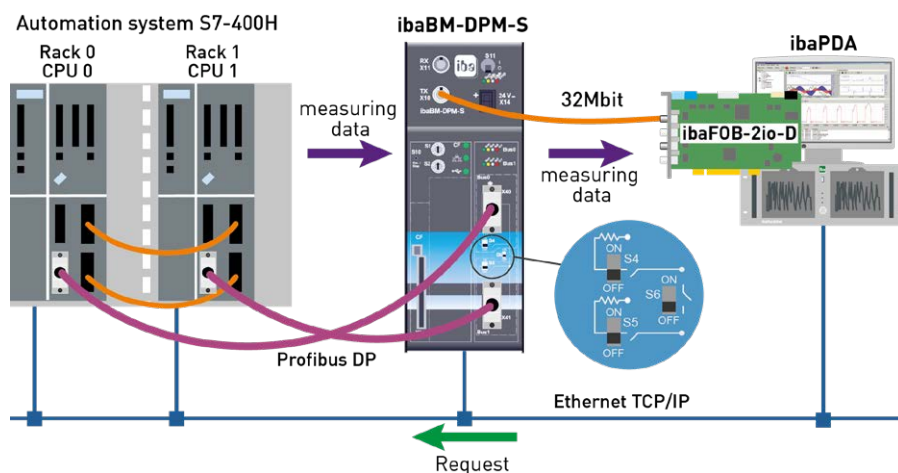


Fig. 58: Request-S7 for ibaBM-DPM-S in redundancy mode

Note



The following description refers to the Request blocks V4.0 or higher.

Other documentation



For detailed information about the redundancy mode of *ibaBM-DPM-S*, please see the device manual.

The functionality of the Request-S7 for *ibaBM-DPM-S* in redundancy mode mostly corresponds to the functionality in standard mode (see chapter [Request-S7 for ibaBM-DPM-S](#), page 98). The differences and extensions are described in the following.

For information and application examples, see chapter [Application examples](#), page 148.

4.6.2 Configuration and engineering SIMATIC S7-300, S7-400 and WinAC

The configuration on the SIMATIC S7 side corresponds to the configuration of *ibaBM-DP* (see chapter [Configuration and engineering SIMATIC S7-300, S7-400 and WinAC](#), page 89).

4.6.3 Configuration and engineering ibaPDA

First of all connect the *ibaBM-DPM-S* as usual to a free link of an *ibaFOBD* input card. In the I/O Manager, add an *ibaBM-DPM-S* device module on the respective link.

Set the "Redundancy mode" to TRUE on the *General* tab. The activated redundancy mode is displayed by an orange colored symbol of the device module.

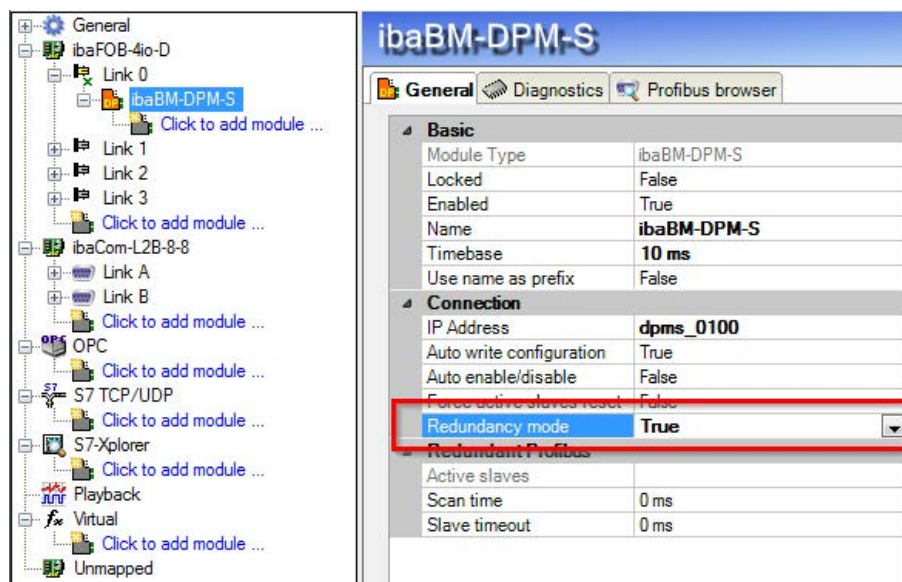


Fig. 59: Setting redundancy mode in I/O Manager

As on the redundant PROFIBUS, both bus systems 0 and 1 are operated in parallel, there are only settings for the "Redundant PROFIBUS" on the *General* tab.

In redundancy mode, the following modules are available with Request-S7:

- S7 Request
- S7 Request Decoder
- Dig512 S7 request

The configuration of the modules corresponds to the configuration in standard mode. For information, see chapter [Configuration and engineering SIMATIC S7-300, S7-400 and WinAC](#), page 99.

Connection settings

For transmitting the operand data to both CPUs of the SIMATIC S7-400H, two different connections are established. These connections are switched on and off depending on availability and requirements.

The connection settings are identical to the Request-S7 for *ibaBM-DP* in redundancy mode (see chapter [➤ Configuration and engineering SIMATIC S7-300, S7-400 and WinAC](#), page 89).

4.7 Request-S7 for ibaCom-L2B

In the following, the Request-S7 version for the *ibaCom-L2B* PROFIBUS card is described.

The solution "Request-S7 for ibaCom-L2B" is replaced by "Request-S7 for ibaBM-DP" with full functional compatibility.

For more information, see chapter ➤ *Replacing Request-S7 on ibaCom-L2B by ibaBM-DP*, page 166.

4.7.1 General information

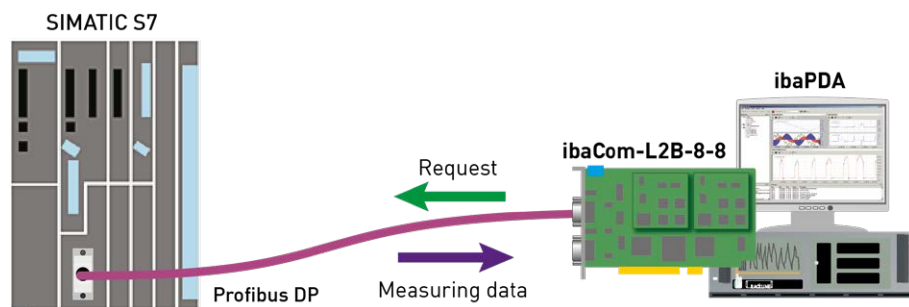


Fig. 60: Request-S7 for ibaCom-L2B

With the L2B Request, the measurement request (handshake request) is sent via PROFIBUS. A section of the IO range of the respective PROFIBUS slave is used for this purpose. No TCP/IP connection between the S7 controller and the ibaPDA computer is required for configuring.

Note



The following description refers to the Request blocks V3.4 or higher. Should you require information regarding older versions, please contact our support.

Older versions are required if ibaPDA-Request-S7-DP/PN is to be used in combination with older S7-CPU's, e.g. CPU 315 before release version 2AF03 or CPU 314 before release version 2AF04.

For information and application examples, see chapter ➤ *Application examples*, page 148.

4.7.2 Configuration and engineering SIMATIC S7-300, S7-400 and WinAC

4.7.2.1 Description of the Request FCs

With these functions, the communication between S7 controller, *ibaPDA* and *ibaCom-L2B* card is initialized and controlled.

The functions have to be called once for each activated PROFIBUS slave in the program.

The used blocks are part of the iba S7 library (see chapter [Iba S7 library](#), page 139).

4.7.2.1.1 Initialization-FC *ibaL2B_Init* (FC111)

This function checks parameters of the S7-CPU and defines the default settings for the DBs. The *ibaL2B_Init* (FC111) function has to be called in all available start-up OBs (OB100, OB101 and OB102).

Description of the formal parameters of the *ibaL2B_Init*:

Name	Type	Data Type	Description
DB_PDA	IN	BLOCK_DB	DB of the ibaPDA communication interface <i>ibaL2B_DB_work</i> Range: 1 through n (see technical data of the CPU); each slave requires its own DB.
INP_RANGE	IN	INT	Limits the number of input bytes to be measured (depends on CPU); we generally recommend the setting "0" (automatic detection). Only if the wrong number is displayed in the "S7 System Info" (<i>ibaPDA</i>), the number of input bytes can be set manually! „0": automatic detection (recommended).
OUT_RANGE	IN	INT	Limits the number of output bytes to be measured (depends on CPU); we generally recommend the setting "0" (automatic detection). Only if the wrong number is displayed in the "S7 System Info" (<i>ibaPDA</i>), the number of output bytes can be set manually. „0": automatic detection (recommended).
MARKER_RANGE	IN	INT	Limits the number of the marker bytes to be measured (depends on CPU); we generally recommend to define "0" (automatic detection). Only if the wrong number is displayed in the "S7 System Info" (<i>ibaPDA</i>), the number of marker bytes can be set manually! „0": automatic detection (recommended).

Name	Type	Data Type	Description
ERROR_STATUS	OUT	BYTE	<p>Parameterization error status</p> <p>The following error codes can be displayed:</p> <p>a) Concerning parameter "DB_NUMBER"</p> <p>"11" - DB is read-only (-> change DB attribute)</p> <p>"12" - DB no=0 or higher than the max. permissible DB number of this CPU (-> change to a valid DB)</p> <p>"13" - DB with the specified number does not exist. (-> copy/rename DB22)</p> <p>"14" – no sequence-relevant DB (-> contact iba hotline)</p> <p>"15" - DB too short (-> copy/rename DB22)</p> <p>b) While reading CPU parameters</p> <p>"21" - not enough memory space for data set (-> contact iba hotline)</p> <p>"22" - SZL_ID is wrong or unknown within this CPU. -> contact iba hotline</p> <p>"23" - Index is wrong or not allowed. -> contact iba hotline</p> <p>c) Exceptional status</p> <p>"19" – Initialization not finished (-> contact iba hotline)</p>

The following SFCs are used internally:

- SFC 6 (RD_SINFO)
- SFC 24 (TEST_DB)
- SFC 51 (RDSYSST)

Ranges for number of inputs, outputs and markers

The number of inputs, outputs and markers in the CPU can be evaluated reliably by the `ibaL2B_Init` function. For this purpose, the value "0" has to be assigned to the parameters as in our example.

In exceptional cases, these parameters may be adjusted/modified after consultation with the iba AG support. These parameters help to avoid configuration errors when selecting signals in *ibaPDA*. If, for example, the user selects a marker which does not exist in the S7, *ibaPDA* will generate an error message. In the *ibaPDA* diagnostics of the corresponding module you can recognize such conflicts quite easily.

In the "Value" column of the "Analog values" tab, the message "I/O address does not exist (6E) ---" appears, if the value does not comply with the range detected in `ibaL2B_Init`.

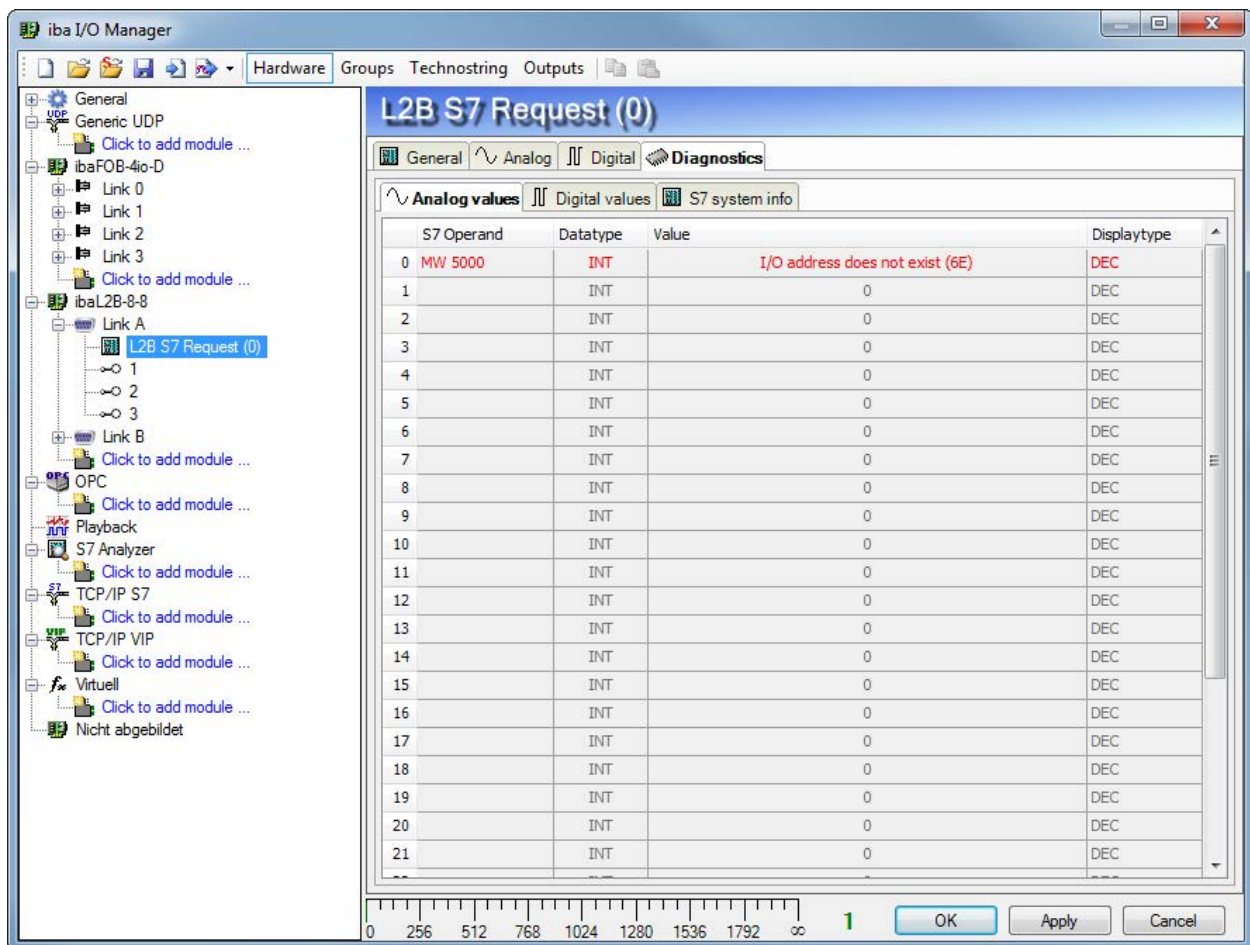


Fig. 61: Display of wrong configurations

In such case, please check the following:

- Did you select an operand address in *ibaPDA* which is available in the CPU? If not, use an available operand.
- Do the automatically determined address ranges in the *ibaPDA* menu „S7 System Info“ correspond to the technical data of the S7-CPU? If not, set the number of I, O and M Bytes manually (from the "technical data" of the S7-CPU) when parameterizing the `ibaL2B_Init`.

4.7.2.1.2 Communication-FC `ibaL2B_Req` (FC112)

This function provides the communication between the S7 and the *ibaCom-L2B* card within the *ibaPDA* PC. It also supervises the communication links.

The `ibaL2B_Req` (FC112) function has to be called once per slave in the cyclic program.

Description of the formal parameters of the ibaL2B_Req (FC112):

Name	Type	Data Type	Description
DB_PDA	IN	BLOCK_DB	DB of the ibaPDA communication interface ibaL2B_DB_work Range: 1 through n (see technical data of the CPU); each slave requires an own DB. The DB needs to be the same as the DB referenced by ibaL2B_Init.
DP_SEND_ADR	IN	INT	DP start address of the ibaCom-L2B card in the peripheral output range for sending data.
DP_RECEIVE_ADR	IN	INT	DP start address of the ibaCom-L2B card in the peripheral input range for the reception of data
TIME_OUT_VAL	IN	INT	Communication timeout Timeout = TIME_OUT_VAL x 100 ms
TIMER_NR	IN	Timer	Timer for the internal (stand-by) clock generator. Range: T0 to Tn (see technical data of the CPU); each slave requires an own timer.
TIME_OUT_FLAG	OUT	BOOL	Communication lost between S7 and ibaCom-L2B card

Name	Type	Data Type	Description
ERROR_STATUS	OUT	BYTE	<p>Parameterization error</p> <p>The following error codes can be displayed:</p> <p>" 92" - Invalid command from ibaPDA (-> contact iba hotline)</p> <p>"100" - not enough space in the DB address range for the requested analog values (-> reduce data amount in ibaPDA to a max. of 112 Byte)</p> <p>"101" – channel no. > 31 (-> contact iba hotline)</p> <p>"102" – data type is not supported (analog values only) (-> contact iba hotline)</p> <p>"103" – range identifier missing or cannot be interpreted (-> contact iba hotline)</p> <p>"104 – this variable type cannot be interpreted. -> contact iba hotline</p> <p>"105" – DB 0 was requested as data source. (-> use other allowed DB)</p> <p>"106" – DB number is higher than max. permissible number of DBs of this CPU. (-> change to allowed DB)</p> <p>"107" - DB with the specified number does not exist. (-> request existing DB data)</p> <p>"109" - DB is too short for the accessed operand (-> copy/rename DB22)</p> <p>"110" – actual operand address (I, O, M, P) does not exist. (-> request available operands)</p> <p>"111" – Initialization aborted with error (-> analyze error state of ibaL2B_Init, otherwise contact iba hotline)</p> <p>"112" – Initialization not performed (-> call ibaL2B_Init in the start-up OBs)</p>

The following SFCs are used internally:

- SFC 20 (BLKMOV)
- SFC 24 (TEST_DB)
- SFC 36 (MSK_FLT)
- SFC 37 (DMSK_FLT)
- SFC 50 (RD_LGADR)

4.7.2.1.3 Communication-FC ibaL2B_Req_CP (FC113) for CP342-5

Not every S7-300 CPU model has an integrated or free (unused) DP interface. As described before, in the S7-400 family, an external CP can be addressed with the functions ibaL2B_Init (FC111) and ibaL2B_Req (FC112). The external CP is connected to the so-called "communication bus" and acts like an internal DP interface.

With CPUs of the S7-300 family it is not as straightforward. They do not have a "communication bus". The connections to the iba slaves are realized by the standard blocks FC1 (DP_SEND) and FC2 (DP_RCV) from the S7 standard library (and not over the peripheral address space). The physical connection is carried out via the external CP342-5. For this case, the ibaL2B_Req (FC112) is replaced by the ibaL2B_Req_CP (FC113). Moreover, another DB ibaL2B_CP_SNDRCV (in example DB10) is required for all slave connections together, where the peripheral data of all slaves are stored temporarily.

The function ibaL2B_Req_CP (FC113) has to be called instead of the ibaL2B_Req (FC112) once per slave in the cyclic program.

Description of the formal parameters of the ibaL2B_Req_CP (FC113):

Name	Type	Data Type	Description
DB_PDA	IN	BLOCK_DB	DB of the ibaPDA communication interface ibaL2B_DB_work Range: 1 through n (see technical data of the CPU); each slave requires an own DB. The DB needs to be the same as the DB referenced by ibaL2B_Init.
DB_DP_Data	IN	BLOCK_DB	DB which buffers the send and receive data for FC_SEND and FC_RECV. Range: Range from 1 to n (see technical data of the CPU).The data of all slaves can be collected in one DB.
DB_ADR_Offset_SEND	IN	INT	DB-OFFSET address of the DB area for the SEND data (122 Byte are occupied per slave). Each slave must have an own OFFSET address for the SEND data.
DB_ADR_Offset_RECV	IN	INT	DB-OFFSET address of the DB range for receiving (RECEIVE) data (16 Byte are occupied per slave). Each slave must have an own OFFSET address for the RECEIVE data.
TIME_OUT_VAL	IN	INT	Communication timeout Timeout = TIME_OUT_VAL x 100 ms
TIMER_NR	IN	Timer	Timer for the internal (stand-by) clock generator. Range: T0 to Tn (see technical data of the CPU); each slave requires an own timer.

Name	Type	Data Type	Description
TIME_OUT_FLAG	OUT	BOOL	Communication lost between S7 and iba-Com-L2B card
ERROR_STATUS	OUT	BYTE	<p>Parameterization error</p> <p>The following error codes can be displayed:</p> <p>" 92" - Invalid command from ibaPDA (-> contact iba hotline)</p> <p>"100" - not enough space in the DB address range for the requested analog values (-> reduce data amount in ibaPDA to a max. of 112 Byte)</p> <p>"101" – channel no. > 31 (-> contact iba hotline)</p> <p>"102" – data type is not supported (analog values only) (-> contact iba hotline)</p> <p>"103" – range identifier missing or cannot be interpreted (-> contact iba hotline)</p> <p>"104 – this variable type cannot be interpreted. -> contact iba hotline</p> <p>"105" – DB 0 was requested as data source. (-> use other allowed DB)</p> <p>"106" – DB number is higher than max. permissible number of DBs of this CPU. (-> change to allowed DB)</p> <p>"107" - DB with the specified number does not exist. (-> request existing DB data)</p> <p>"109" - DB is too short for the accessed operand (-> copy/rename DB22)</p> <p>"110" – actual operand address (I, O, M, P) does not exist. (-> request available operands)</p> <p>"111" – Initialization aborted with error (-> analyze error state of ibaL2B_Init, otherwise contact iba hotline)</p> <p>"112" – Initialization not performed (-> call ibaL2B_Init in the start-up OBs)</p>

4.7.2.2 Hardware configuration

Perform the following steps:

- Link the DP interface of the CPU or CP as DP master to a PROFIBUS network.
- Install the GSD-file IBA_0F05.GSD either by the menu "Options - Installing GSD-files" or by copy/paste from the demo project.
- After the installation has been finished, you find the iba-GSD-file in the HW catalog under: Profibus-DP - Additional Field Devices - General - ibaL2B < S7 Direct Access >
- Attach the desired number of DP slaves „ibaL2B < S7 Direct Access >" to the PROFIBUS and set the DP addresses. The peripheral addresses are assigned automatically to the slaves as addresses in ascending order.
- HW Config. / Load system data to the S7-CPU.

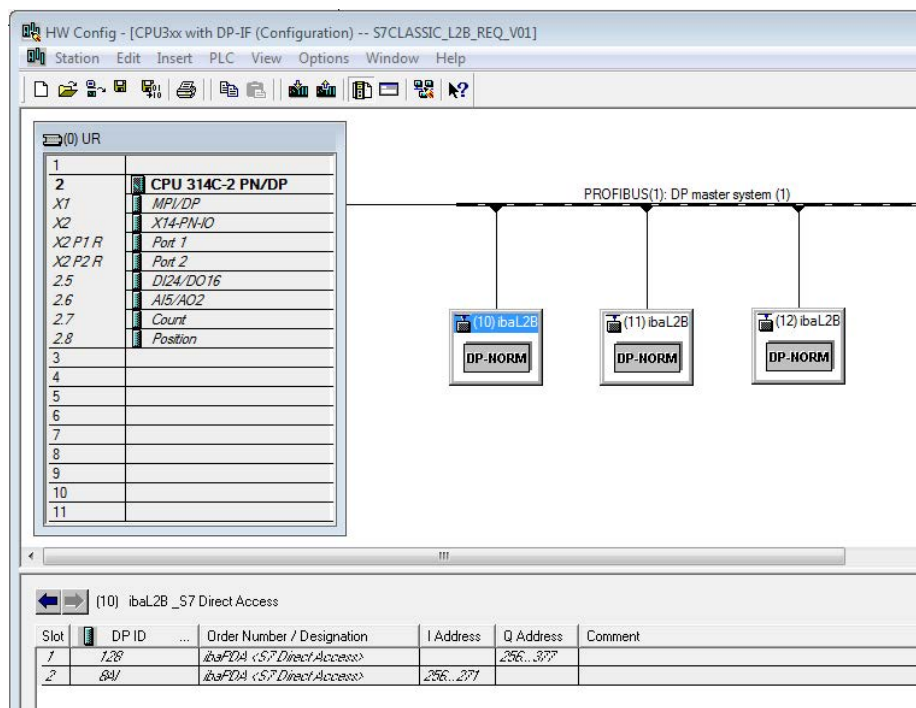


Fig. 62: Hardware configuration

4.7.2.3 Configuration in STEP 7 (STL, LAD, FBD)

4.7.2.3.1 CPU-internal DP interface or CP 443-5 (for S7-400)

If you use a CPU S7-300/400 with integrated DP interface or use the external interface CP 443-5 for a S7-400, proceed as follows. When using the external DP interface CP342-5 of a CPU S7-300, continue in chapter [External DP interface CP342-5](#), page 123.

For Request-S7 with one PROFIBUS slave:

- Copy the following blocks from the iba S7 library (see chapter [Iba S7 library](#), page 139) to the blocks folder of your STEP 7 project:
 - FC111 (ibaL2B_Init) and FC112 (ibaL2B_Req)
 - DB22 (ibaL2B_DB_work) and UDT22 (ibaL2B_DB_Struct)

Note

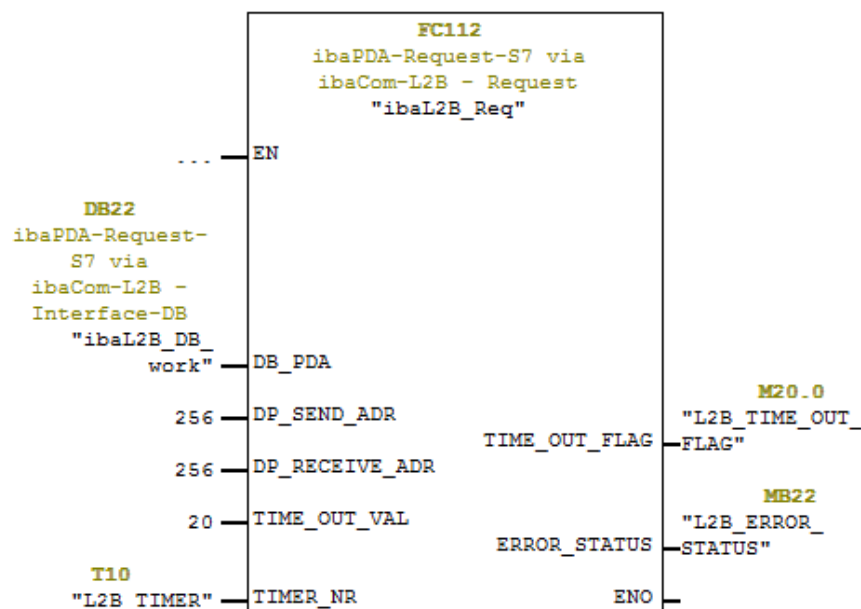
In case the block numbers FC111, FC112, DB22 and UDT22 are already used in your project, assign other free numbers to the blocks from the iba S7 library when copying.

- Create the error OBs (OB82, OB85, OB86, OB87, OB122) in order to prevent CPU stops in case of an error.

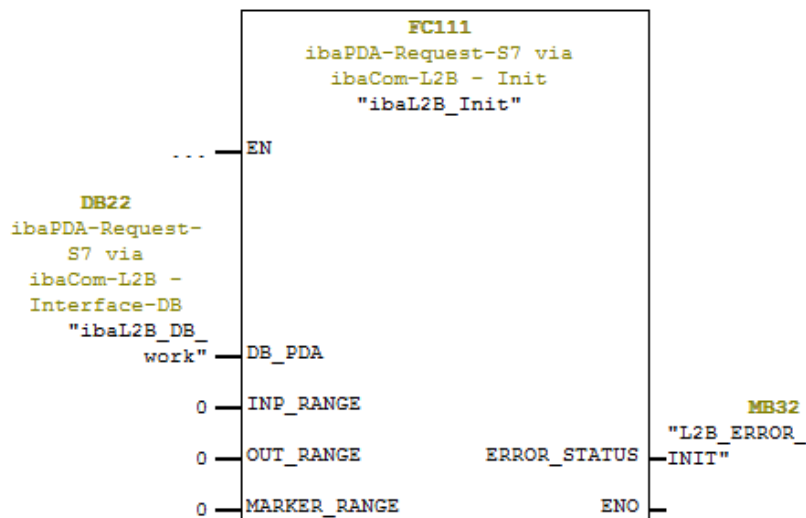
Note

DB22 and UDT22 contain a data structure which is not required for processing. Basically, a DB with a length of 1500 bytes would be sufficient. The data structure is useful for diagnostic purposes only.

- Call and parameterize the ibaL2B_Req (FC112) in the cyclic program.



- Call and parameterize the ibaL2B_Init (FC111) in the start-up OBs (OB100, OB101, OB102).



For Request-S7 with multiple PROFIBUS slaves:

- For each Request slave, there must be one data block (DB) in the blocks folder which is referenced by both FCs. Copy the ibaL2B_DB_work (DB22) to a DB with a new DB number.
- Call the ibaL2B_Init (FC111) once more with a new DB number in the start-up OBs.
- In the cyclic OB, you have to call the ibaL2B_Req (FC112) once more with the new DB number and the peripheral addresses of the new PROFIBUS slave.

Finish:

- Load all blocks into the S7-CPU and restart.

4.7.2.3.2 External DP interface CP342-5

When using an external DP interface CP342-5, perform the following steps.

For Request-S7 with one PROFIBUS slave:

- Copy the following blocks from the iba S7 library (see chapter [Iba S7 library](#), page 139) to the blocks folder of your STEP 7 project:
 - FC113 (ibaL2B_Req_CP) and FC111 (ibaL2B_Init)
 - DB10 (ibaL2B_CP_SNDRCV), DB22 (ibaL2B_DB_work) and UDT22 (ibaL2B_DB_Struct)

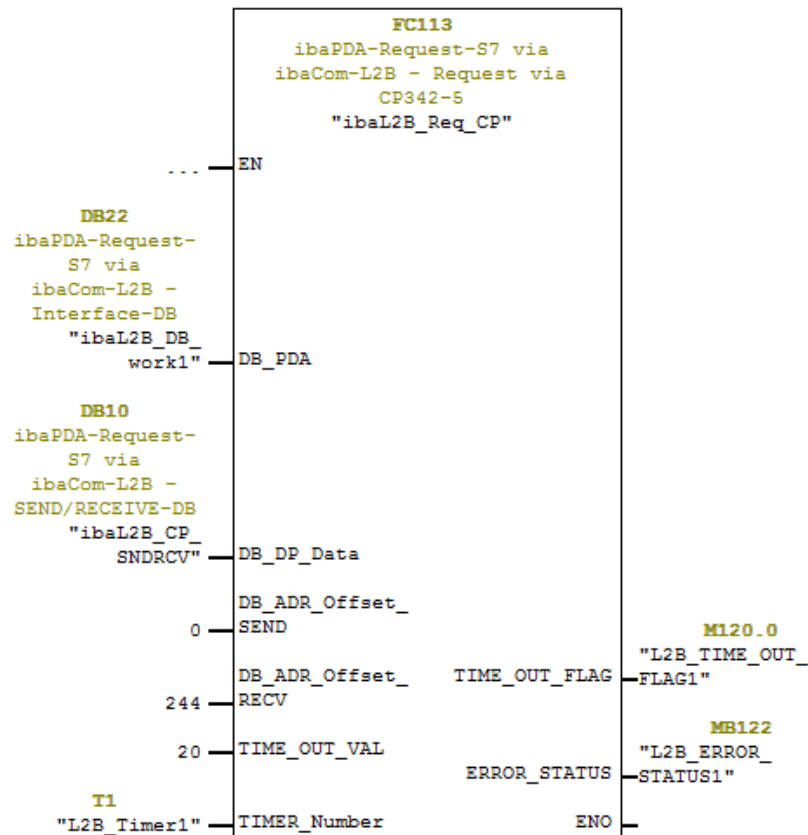
Note



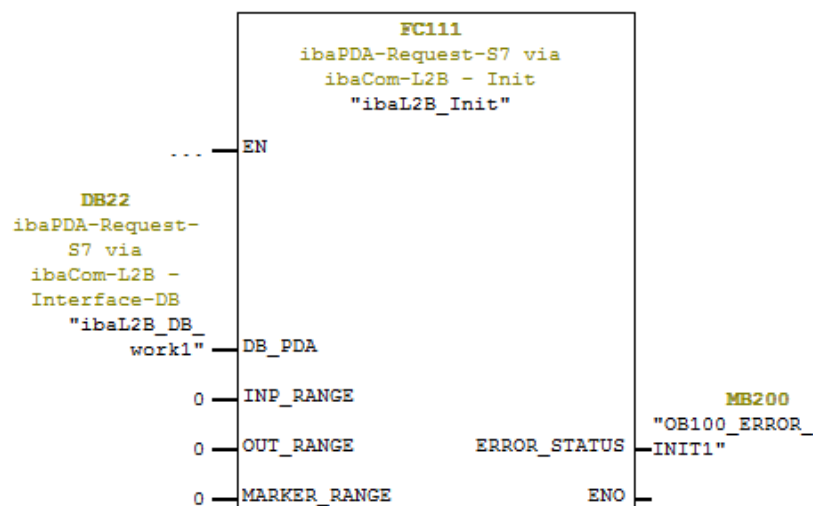
In case the block numbers FC111, FC113, DB10, DB22 and UDT22 are already used in your project, please assign other free numbers to the blocks from the iba S7 library when copying.

- Create the error OBs (OB82, OB85, OB86, OB87, OB122) in order to prevent CPU stops in case of an error.

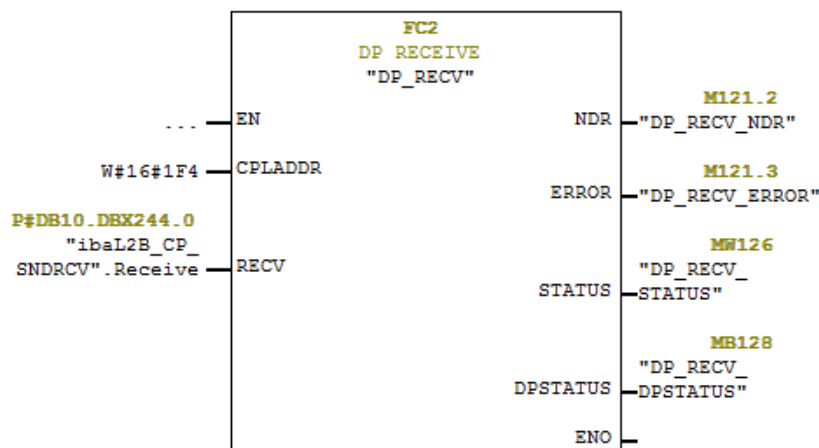
- Call and parameterize the ibaL2B_Req_CP (FC113) in the cyclic program.
As offset addresses for send and receive data, own areas within the DB „ibaPDA_SENDRECV" (DB10) have to be defined for each slave and handed over as parameters "DB_ADR_Offset_SEND" or „DB_ADR_Offset_RECV".



- Call and parameterize the ibaL2B_Init (FC111) in the start-up OBs (OB100, OB101, OB102).



- Call and parameterize FC1 (DP_SEND) and FC2 (DP_RECV) in the cyclic program.



For Request-S7 with multiple PROFIBUS slaves:

- For each Request slave, there must be one data block (DB) in the blocks folder which is referenced by both FCs. Copy the ibaL2B_DB_work (DB22) to a DB with a new DB number.
- In the ibaL2B_CP_SNDRCV (DB10), 122 Bytes have to be reserved in the "Send" array and 16 Bytes in the "Receive" array for each Request slave. If necessary, you have to extend the arrays.
- Call the ibaL2B_Init (FC111) once more with a new DB number in the start-up OBs.
- In the cyclic OBs, you have to call the ibaL2B_Req_CP (FC113) once more with a new DB number and the respective address offsets for the send and receive range in the ibaL2B_CP_SNDRCV (DB10) of the new PROFIBUS slave.

Finish:

- Load all blocks in the S7-CPU and restart.

Note



The structure of the send and receive data in DB10 must correspond to the output addresses and input addresses of the DP slaves.

If there are more than two slaves, the arrays in DB10 have to be extended by 122 Bytes for sending and 16 Bytes for receiving direction, respectively.

In case of four slaves, for example, the following address assignment applies:

	DB10		PROFIBUS P-addresses	ibaL2B_Req_CP. ...Offset_SEND	ibaL2B_Req_CP. ...Offset_RECV
1: Slave	Send	Offset 0	Q-address 0	0	
2: Slave		Offset 122	Q-address 122	122	
3: Slave		Offset 244	Q-address 244	244	
4: Slave		Offset 366	Q-address 366	366	

1. Slave	Receive	Offset 0	I-address 0		488
2. Slave		Offset 16	I-address 16		504
3. Slave		Offset 32	I-address 32		520
4. Slave		Offset 48	I-address 48		536

Note

All parameters like times and markers have to be different!

4.7.2.4 Configuration in STEP 7 (CFC)

4.7.2.4.1 CPU-internal DP interface or CP 443-5 (for S7-400)

If you use a CPU S7-300/400 with integrated DP interface or use the external interface CP 443-5 for a S7-400, proceed as follows.

When using the external DP interface CP342-5 of a CPU S7-300, continue in chapter [↗ External DP interface CP342-5](#), page 123.

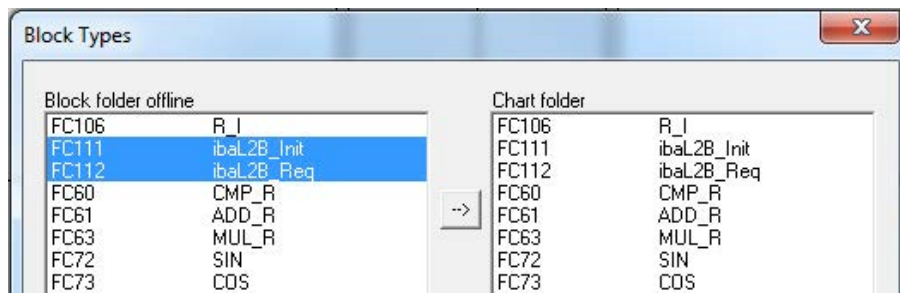
For Request-S7 with one PROFIBUS slave:

- Copy the following blocks from the iba S7 library (see chapter [↗ Iba S7 library](#), page 139) to the blocks folder of your STEP 7 project:
 - FC111 (ibaL2B_Init) and FC112 (ibaL2B_Req),
 - DB22 (ibaL2B_DB_work) and UDT22 (ibaL2B_DB_Struct)

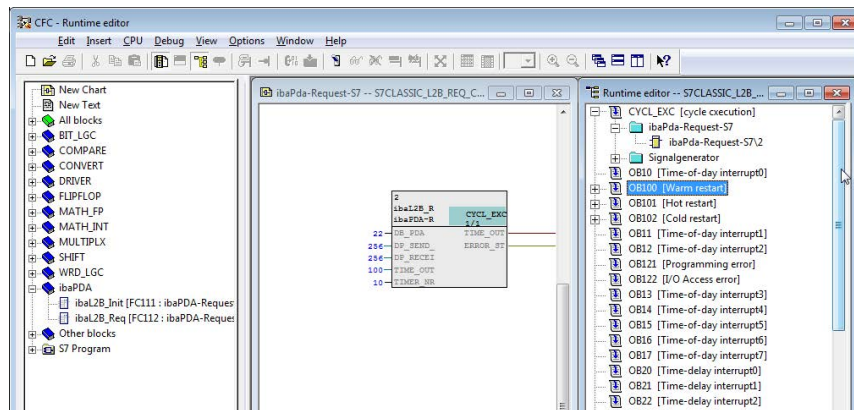
Note

In case the block numbers FC111, FC112, DB22 and UDT22 are already used in your project, assign other free numbers to the blocks from the iba S7 library when copying.

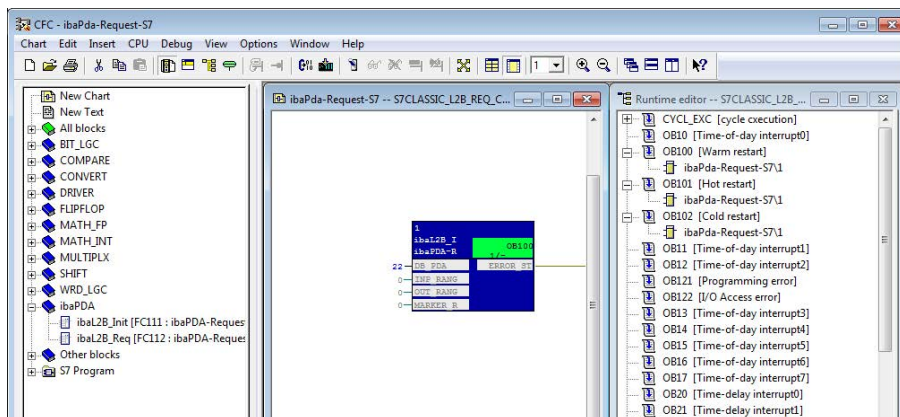
- Start the CFC editor and import these blocks (under menu "Options – Block Types"). The FCs are stored in the group "ibaPDA."



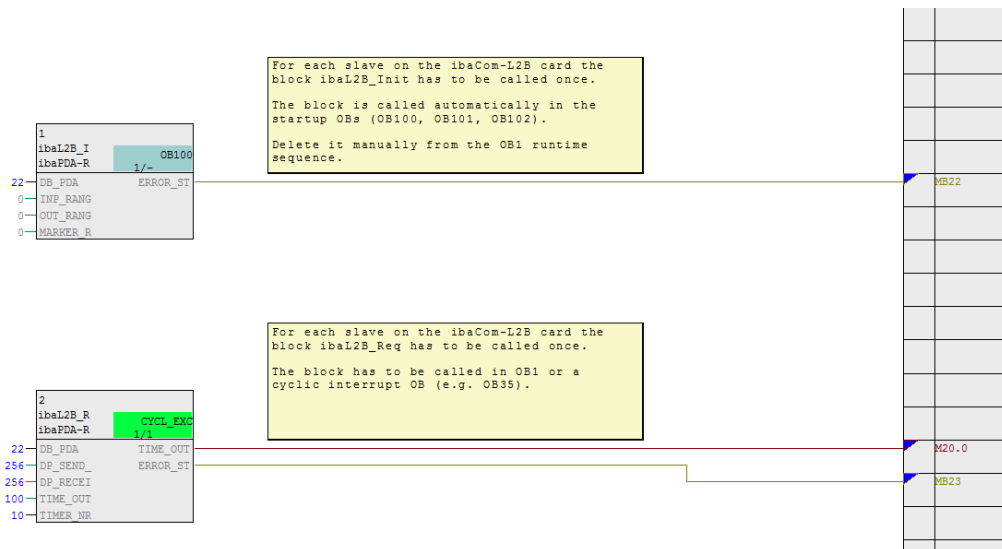
- Drag the ibaL2B_Req (FC112) to your CFC chart. Pay attention to the run sequence. The ibaL2B_Req has to be called in a cyclic task (e. g. cyclic interrupt OB35 or free-running OB1).



- Drag the ibaL2B_Init (FC111) to your CFC chart. Pay attention to the run sequence. The ibaL2B_Init is entered automatically in the start-up tasks (OB100, OB101 and OB102). Remove it from the cyclic tasks (e.g. OB35).



- Connect the blocks:
 - Enter the DB number of the Interface-DB (DB22) on both FCs on the input DB_NUMBER.
 - Enter the peripheral addresses of the PROFIBUS slave under DP_SEND_ADR and DP_RECEIVE_ADR of the ibaL2B_Req block from the hardware configuration.



Make sure to create the error OBs (OB82, OB85, OB86, OB87 and OB122) in order to prevent the CPU from running into STOP in case of an access failure.

To do so, you have to define an empty runtime group within the run sequence of these tasks if they are empty. Alternatively, you may place any dummy block into the error OB. This block must not be deleted, too.

Note



Do not choose the option "Delete empty runtime groups" for compilation otherwise the error OBs will be removed! Some older CFC versions have this option in the compilation dialog.

For Request-S7 with multiple PROFIBUS slaves:

- For each Request slave, there must be one data block (DB) in the blocks folder which is referenced by both FCs. Copy the ibaL2B_DB_work (DB22) to a DB with a new DB number.
- The ibaL2B_Init (FC111) has to be called once more on a CFC chart with a new DB number.
- The ibaL2B_Req (FC112) has to be called once more on a CFC chart with the new DB number and the peripheral addresses of the new slave.

After each program modification:

- Compile the complete program: Activate the option "Entire program", activate the option "Generate SCL source" (for CFC version 6.1 or higher), deactivate the option "Delete empty runtime groups" if available.
- Load the program and restart S7-CPU.

Note

Depending on the S7-CPU type, you may notice the following warnings after compiling and loading:

"W: OB101/ OB102 is not supported by this CPU" or

"W: OB101/OB102 could not be downloaded, because it is not supported by the connected online CPU".

These messages are displayed because the `ibaL2B_Init` is entered automatically in the start-up tasks OB101 and OB102. However, some S7-CPU types do not support OB101 and OB102. The warnings can be ignored.

4.7.3 Configuration and engineering ibaPDA

4.7.3.1 General interface settings

The interface is configured in the *ibaPDA* "I/O Manager". Prerequisite is the installation of an *ibaCom-L2B* card in the *ibaPDA* computer.

After the *ibaCom-L2B* card has been installed, it is displayed automatically in the interface tree.

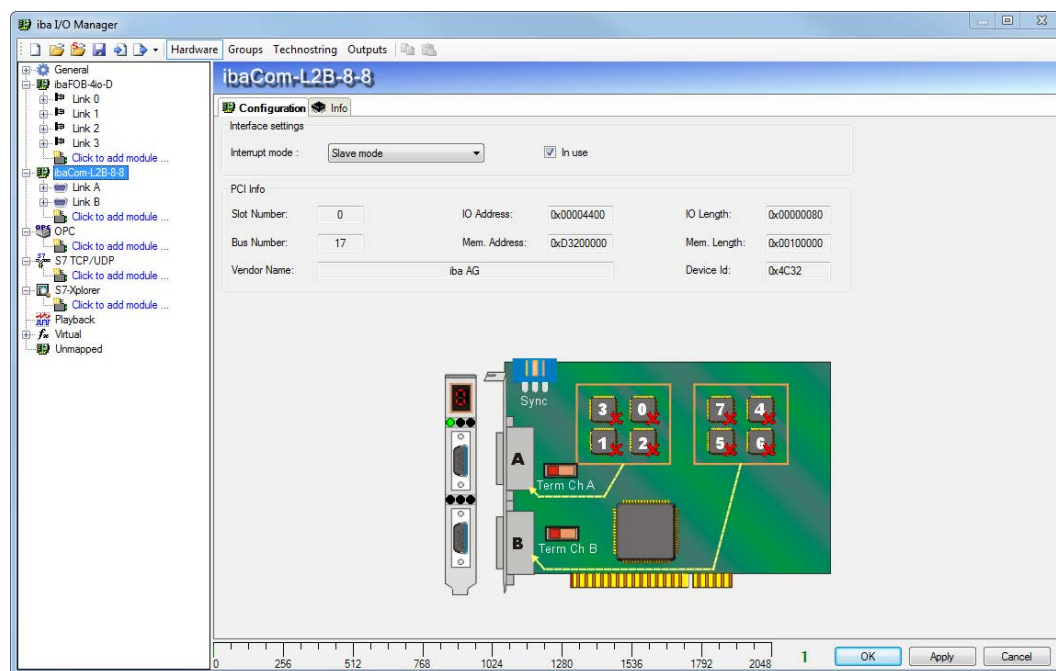


Fig. 63: I/O Manager, display ibaFOB-D card

Other documentation



For more information about the *ibaCom-L2B* card, please see the respective manual.

4.7.3.2 General module settings

The module types "L2B S7 Request" and "L2B S7 Request Dig512" are available.

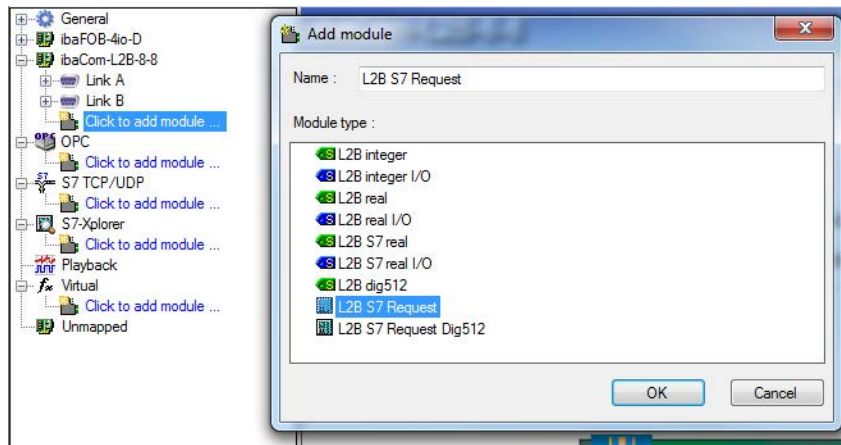


Fig. 64: Module types

The L2B modules have the following common specific setting options:

Profibus

Slave No.

PROFIBUS slave address assigned to the module.

■ Profibus – Timeout

Waiting time in seconds before the firmware regards the connection as interrupted.

For a description of the other general module settings, see chapter [↗ General module settings](#), page 16

In contrast to the Request-S7 solutions with the *ibaBM-DP* and *ibaBM-DPM-S* devices, no additional connection settings are required. The "Request-Handshake" is done via a part of the IO peripheral area.

4.7.3.3 Module L2B S7 Request

Up to 32 analog and 32 digital signals can be acquired with the "L2B S7 Request" module.

A separate PROFIBUS slave and Request block call has to be configured for each module.

For a description of the module settings, see chapter [↗ General module settings](#), page 16 and [↗ General module settings](#), page 130.

4.7.3.4 Module L2B S7 Request Dig512

With the "L2B S7 Request Dig512" module, up to 512 digital signals can be acquired that are sent as max. of 32 words (16 Bit). This is why this module type is especially suited for applications where many digital signals have to be acquired and the digital values of the modules „L2B S7 Request" which can be addressed directly are not sufficient.

A separate PROFIBUS slave and Request block call has to be configured for each module.

On the "Digital" tab, you can enter directly the words as basic signals for decoding over absolute S7 operands. Only word operands (e. g. PIW, MW, DBW) are allowed.

It is also possible to use S7 symbols by generating address books. For more information, see chapter [↗ Selection via the operand symbols](#), page 20. The signals selected in S7 CFC and symbol browser are applied and the columns Name, S7 symbol, S7 Operand and Data Type are filled in automatically.

When clicking on the <+>-symbol on the left in each row, a list is opened in which you can enter a name for each of the 16 digital values.

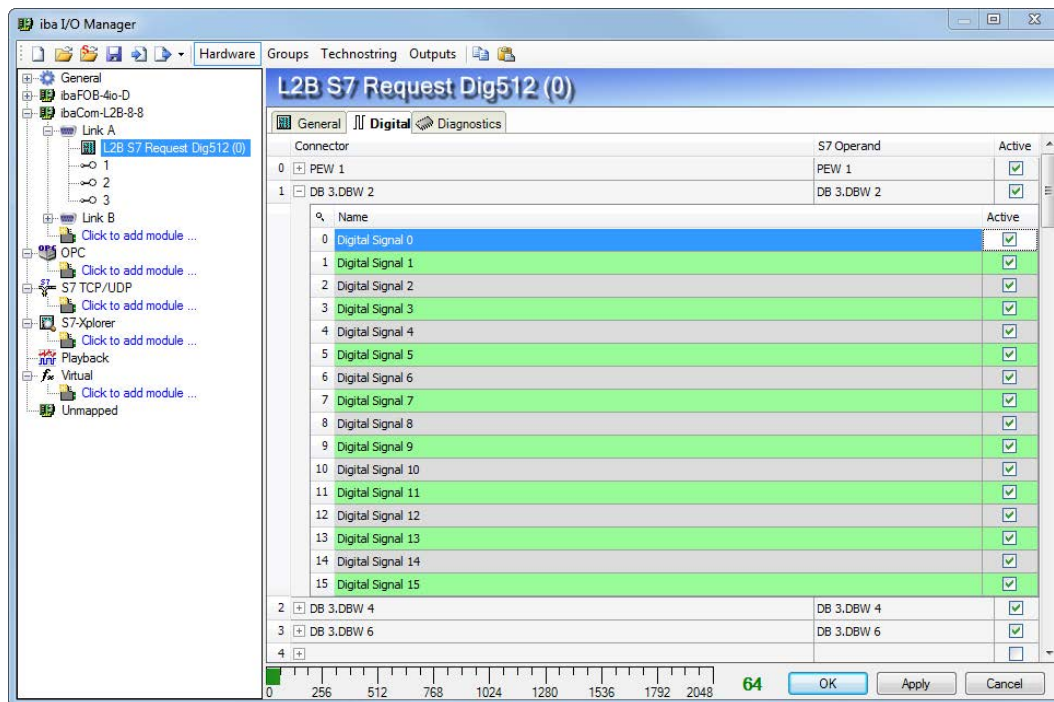


Fig. 65: Module L2B S7 Request Dig512, Digital tab

4.7.4 Diagnostics

4.7.4.1 ibaCom-L2B card

When marking the L2B card in the *ibaPDA* I/O Manager, a status overview of the card is displayed. The green checkmark on the slave number indicates an active connection (in the figure slaves 0-2 are active).

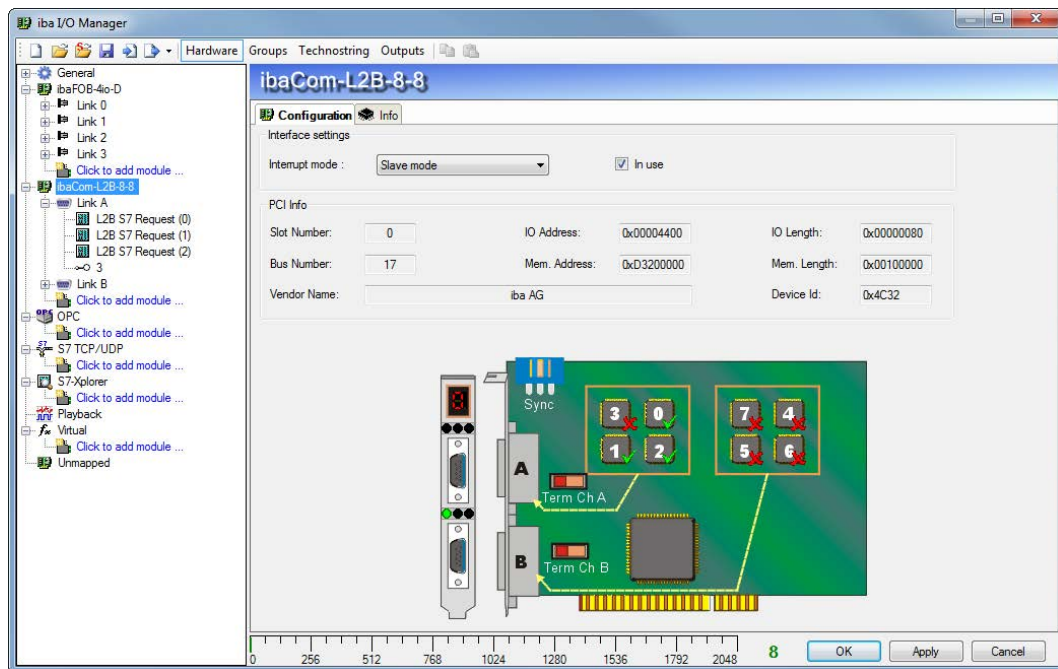


Fig. 66: I/O Manager, display ibaCom-L2B card

Select the PROFIBUS link to display detailed information regarding the status of each slave. These are mainly interesting for support purposes.

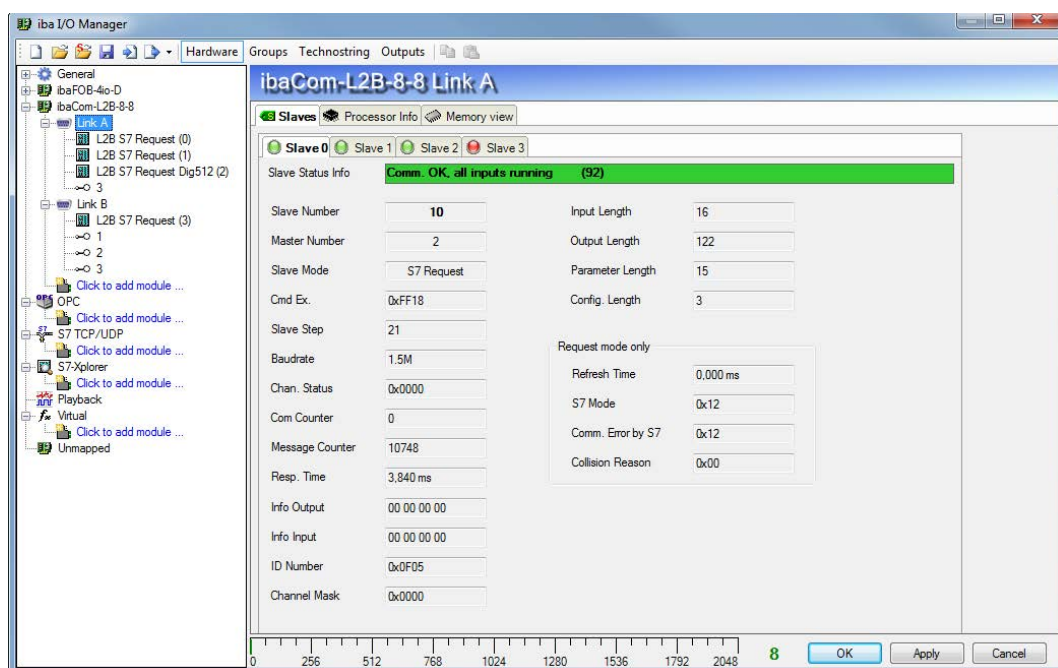


Fig. 67: Detailed information about the slave status

4.7.4.2 Module diagnostics

When marking an L2B-Request module, also a diagnostics tab is shown. It shows details of the individual measured values and system information of the S7-CPU.

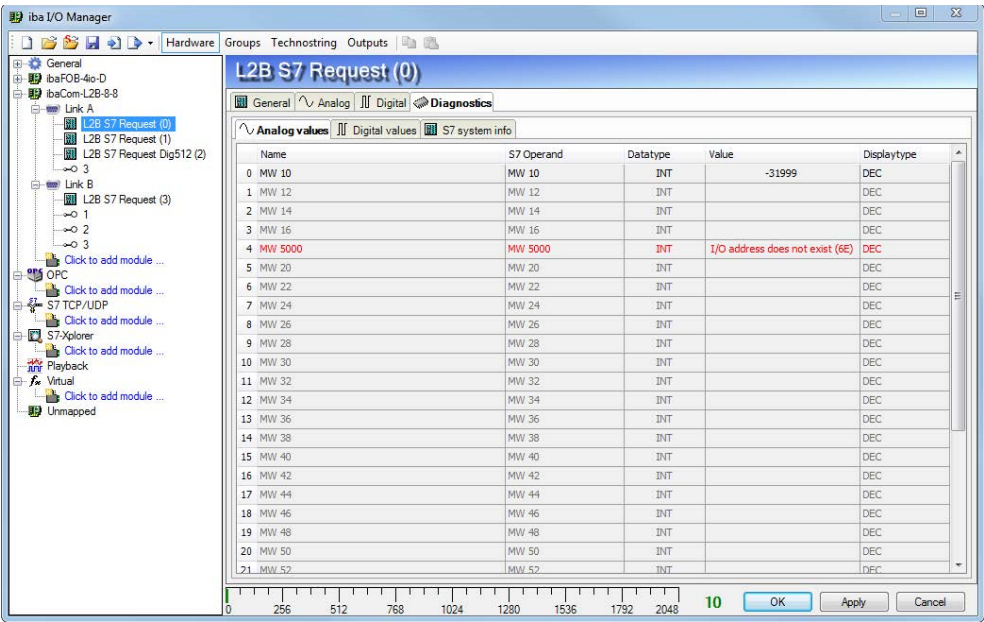


Fig. 68: Diagnostics of the measured values

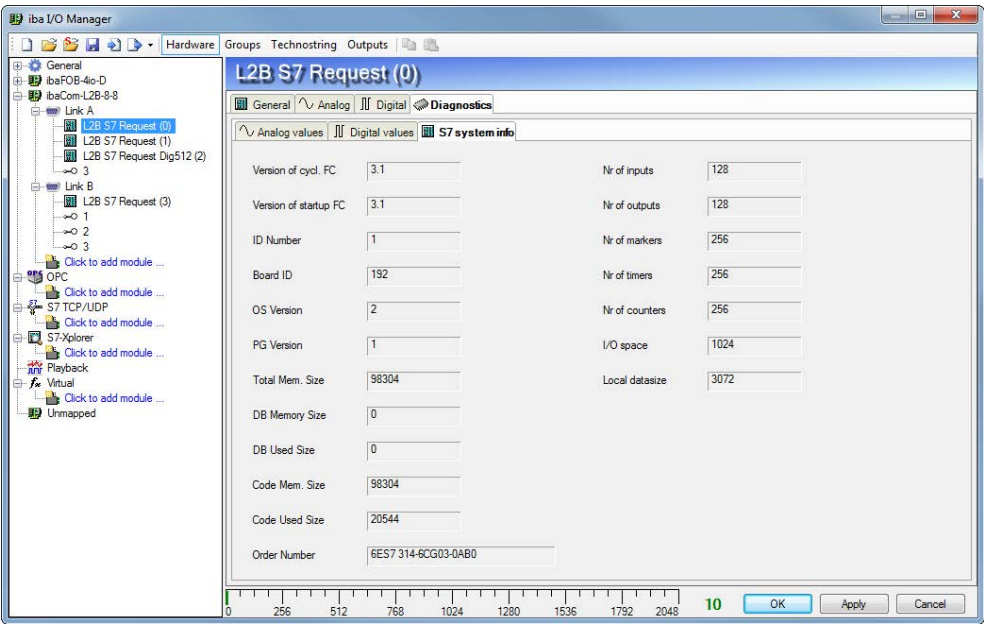


Fig. 69: S7 system information

5 Diagnostics

5.1 Checking the license

In case the "Request-S7" modules are not displayed in the signal tree, you can check in the I/O Manager under "General - Settings - License Info" if your license "ibaPDA-Request-S7-DP/PN" is detected correctly.

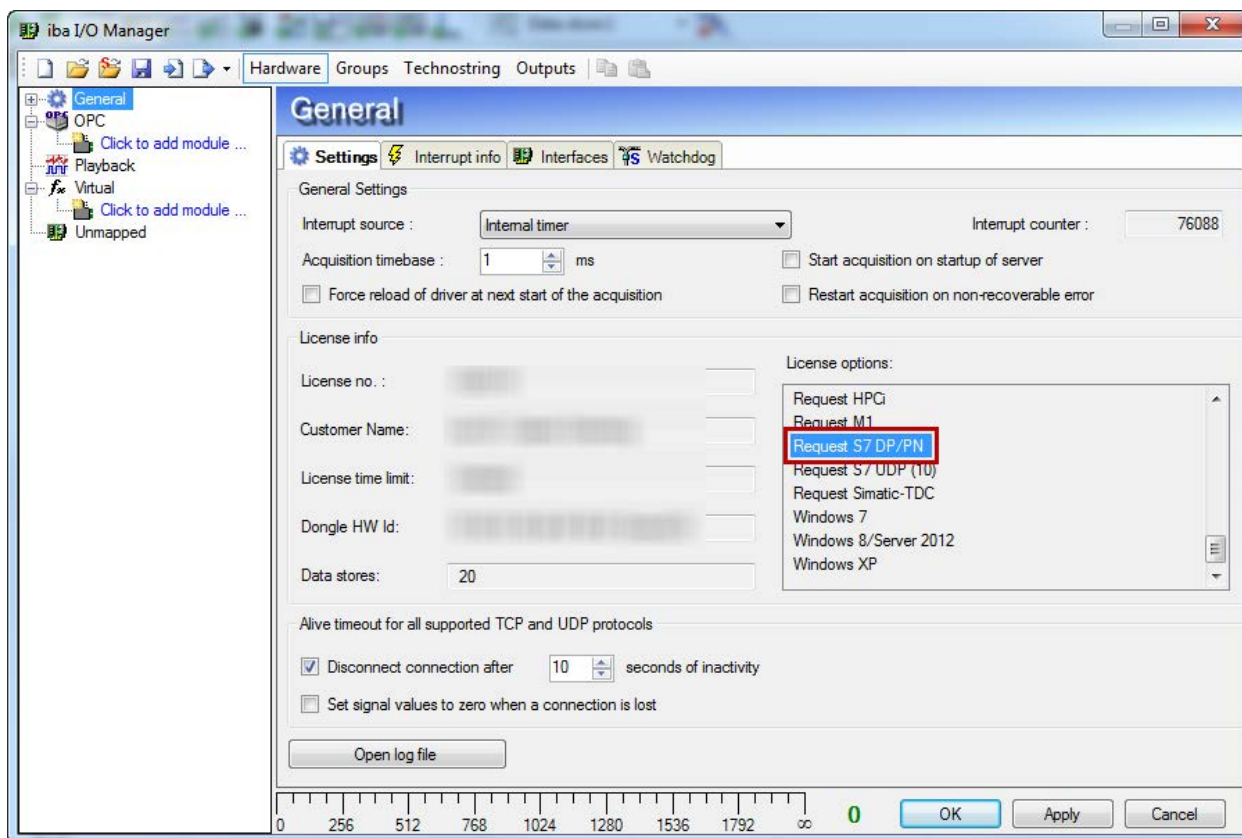


Fig. 70: Checking the license

5.2 Log files

If connections to target platforms or clients have been established, all connection-specific actions are logged in a text file. You can open this (current) file and, e.g., scan it for indications of possible connection problems.

The log file can be opened via the button <Open log file>. The button is available in the I/O Manager:

- for many interfaces in the respective interface overview
- for integrated servers (e.g. OPC UA server) in the *Diagnostics* tab.

In the file system on the hard drive, you will find the log files in the program path of the *ibaPDA* server (...\\Programs\\iba\\ibaPDA\\Server\\Log\\). The file names of the log files include the name or abbreviation of the interface type.

Files named `interface.txt` are always the current log files. Files named `Interface_yyyy_mm_dd_hh_mm_ss.txt` are archived log files.

Examples:

- `ethernetipLog.txt` (log of EtherNet/IP connections)
- `AbEthLog.txt` (log of Allen-Bradley Ethernet connections)
- `OpcUAServerLog.txt` (log of OPC UA server connections)

5.3 Connection diagnostics with PING

PING is a system command with which you can check if a certain communication partner can be reached in an IP network.

Open a Windows command prompt.



Enter the command “ping” followed by the IP address of the communication partner and press <ENTER>.

With an existing connection you receive several replies.

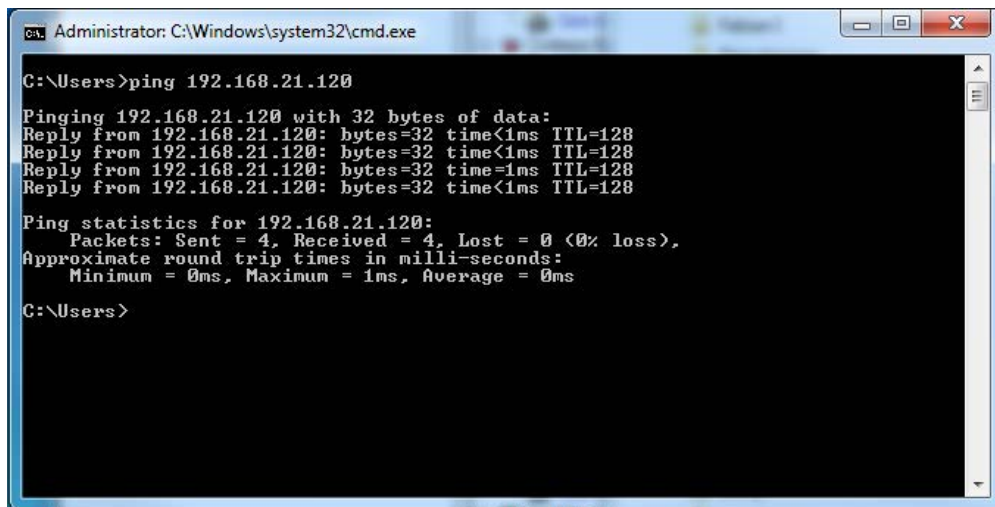


Fig. 71: PING successful

With no existing connection you receive error messages.

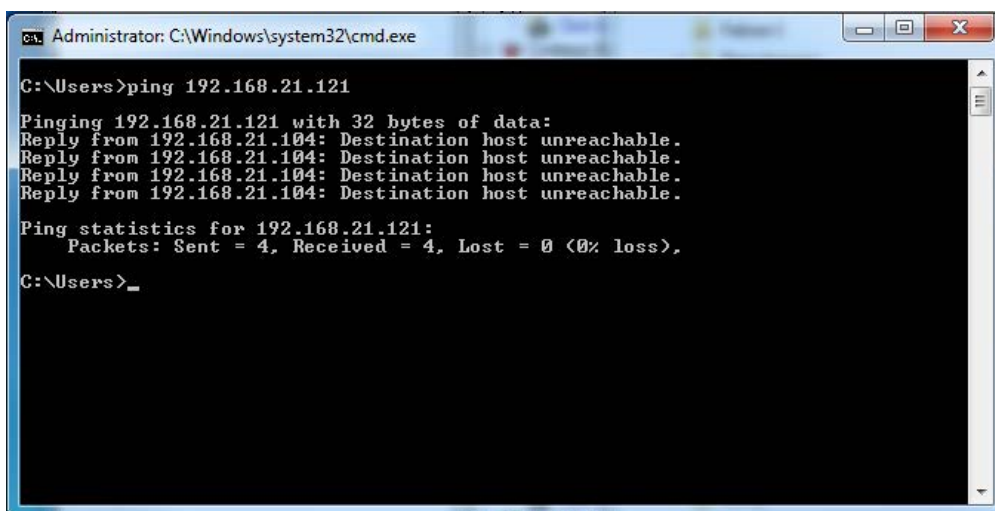
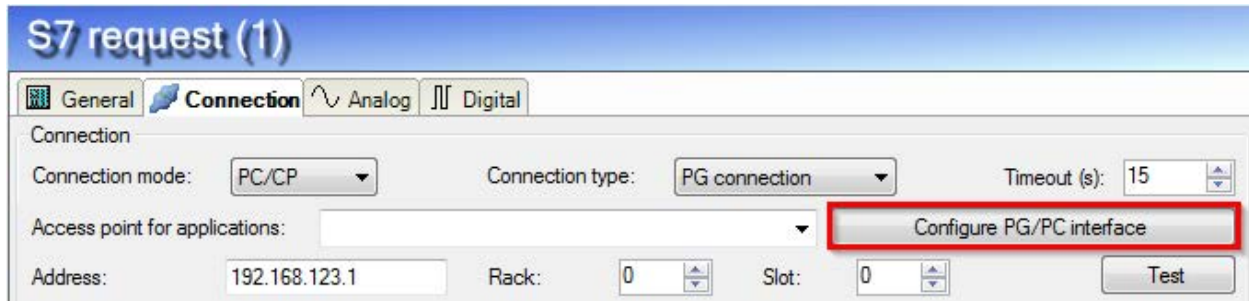


Fig. 72: PING unsuccessful

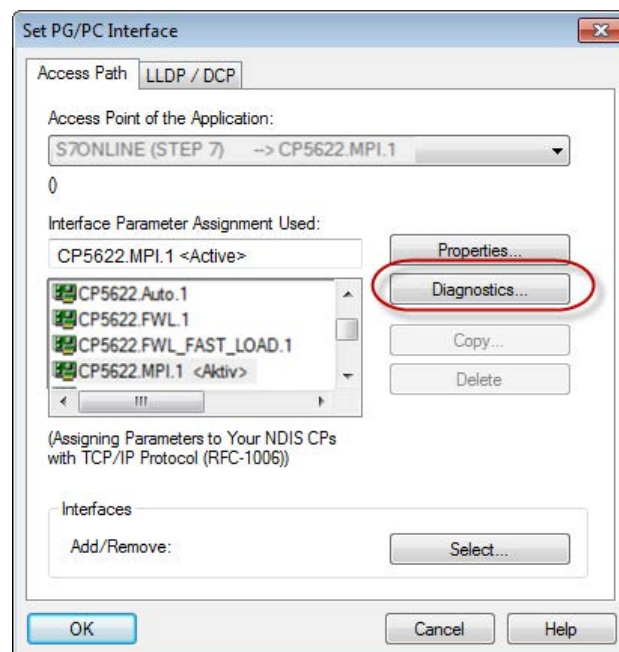
5.4 Connection diagnostics via PG/PC interface

The diagnostic function of the PG/PC interface can be used to test the functionality and connection configuration.

1. Open the PG/PC interface.



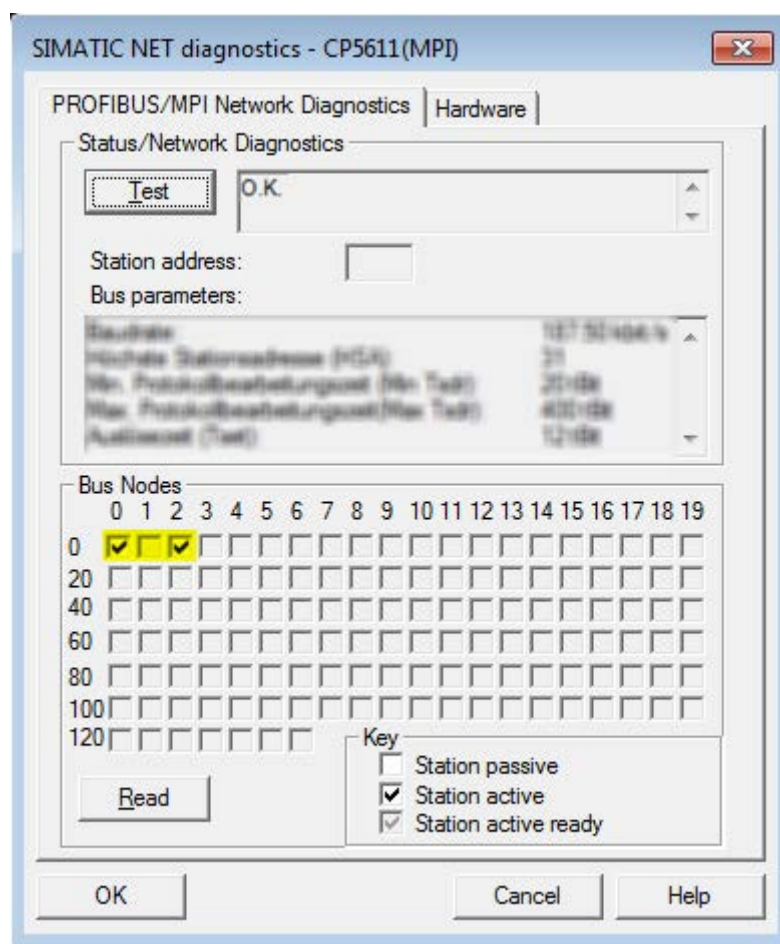
2. Open the diagnostics dialog.



3. The following screenshot shows the example of a diagnostics of a SIMATIC Net CP5622 (PROFIBUS).

With <Test> a net diagnosis is started.

With <Read> the availability of the bus participants is tested.



In this example, on the addresses 0 and 2 an active station was detected.

5.5 Module diagnostics

You will find a diagnostic help with a tabular display of the actual analog and digital values and the data types on the *Diagnostics* tab of each S7 Request module.

See chapter Request-S7 for ibaBM-PN ➔ *Diagnostics*, page 58

See chapter Request-S7 for ibaBM-DP ➔ *Diagnostics*, page 87

See chapter Request-S7 for ibaBM-DPM-S ➔ *Diagnostics*, page 108

See chapter Request-S7 for ibaCom-L2B ➔ *Diagnostics*, page 132

6 Appendix

6.1 Iba S7 library

The iba S7 library is available in two versions:

- SIMATIC Manager : STEP 7 ≥ V5.5
- SIMATIC TIA Portal STEP 7 ≥ V14 SP1

6.1.1 iba S7 library for SIMATIC Manager

The iba S7 library for SIMATIC Manager („ibaS7LibCLASSIC_Vx_y") is suitable for the use with SIMATIC Manager V5.5 or higher. It contains the Request blocks described in the manual, which are required for the use of *ibaPDA-Request-S7-DP/PN*.

You find the iba S7 library as an archived file on the DVD "iba Software & Manuals" under [\04_Libraries_and_Examples\10_Libraries\01_SIMATIC_S7\](#)

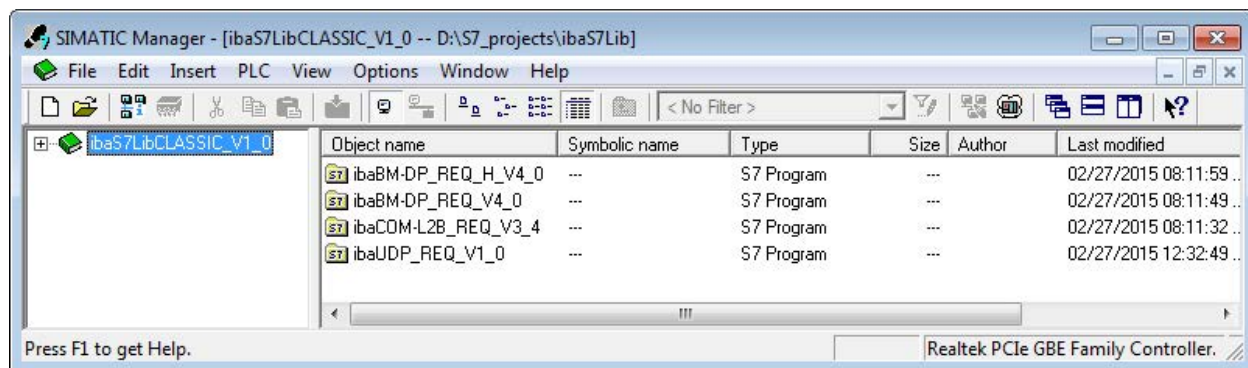


Fig. 73: iba S7 library for SIMATIC Manager

The following components are included:

iba connection	Block name	Block no.	Note
ibaBM-DP ibaBM-DPM-S	ibaDP_Req	FC122	
	ibaDP_DB_PDA	DB10	
	ibaDP_DB_work	DB25	
ibaBM-DP ibaBM-DPM-S Redundancy mode	ibaDP_Req_H	FC123	For S7-400H
	ibaDP_DB_PDA	DB10	
	ibaDP_DB_work	DB25	
ibaBM-PN	ibaREQ_M	FB140	
	ibaREQ_PN	FB141	
	ibaREQ_PNdev	FB150	
	ibaREQ_DB	DB15	
	ibaUDT_UDPact	UDT145	

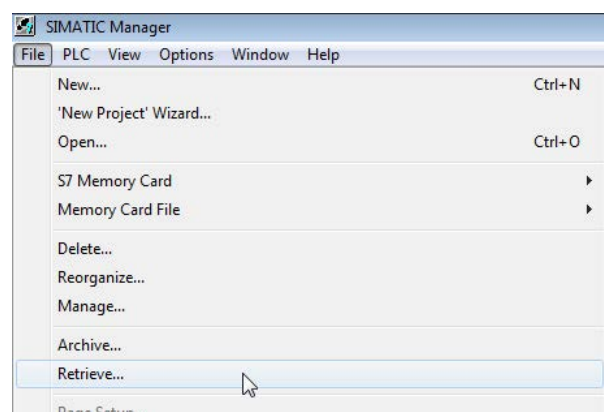
iba connection	Block name	Block no.	Note
ibaCom-L2B	ibaL2B_Init	FC111	formerly FC23 and FC101
	ibaL2B_Req	FC112	formerly FC22 and FC100
	ibaL2B_Req_CP	FC113	formerly FC26 and FC102 only necessary when using a CP342-5 instead of the FC112
	ibaL2B_DB_work	DB22	
	ibaL2B_DB_Struct	UDT22	
	ibaL2B_CP_SNDRCV	DB10	only necessary when using a CP342-5
ibaPDA-Interface-S7-TCP/UDP	ibaREQ_M	FB140	
	ibaREQ_UDPact	FB145	
	ibaREQ_UDPint	FB146	
	ibaREQ_UDPext3	FB147	
	ibaREQ_UDPext4	FB148	
	ibaREQ_DB	DB15	
	ibaUDT_UDPact	UDT145	

Table 12: ibaS7LibCLASSIC block overview

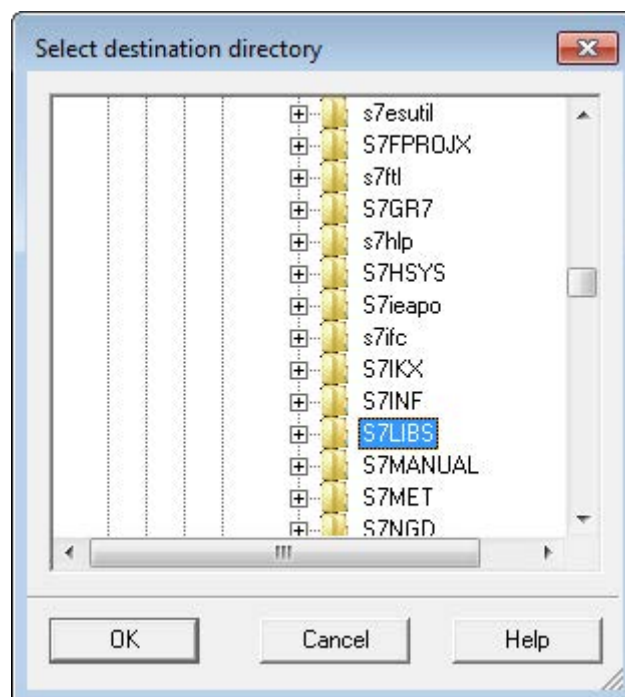
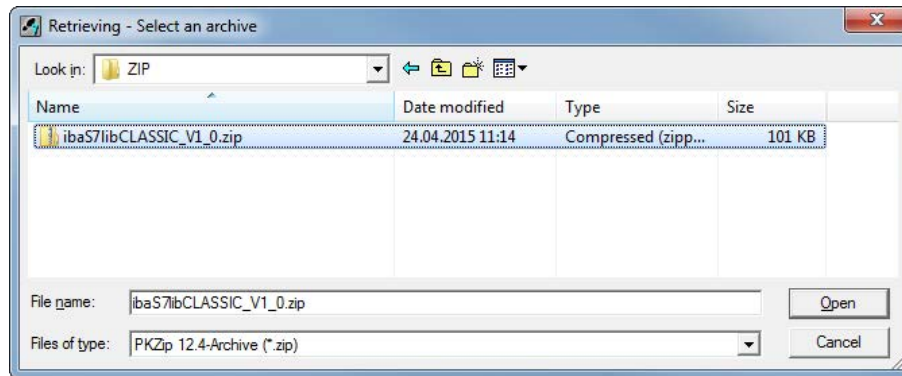
6.1.1.1 Integrating the library

For integrating the library, it has to be retrieved in the SIMATIC Manager. Copy the iba S7 library to a local directory of your computer, on which the SIMATIC Manager is executed.

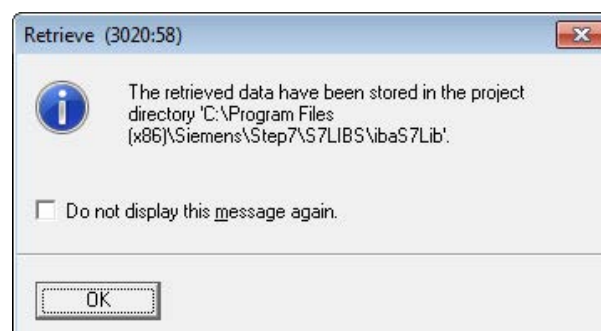
- Select the menu *File – Retrieve...*



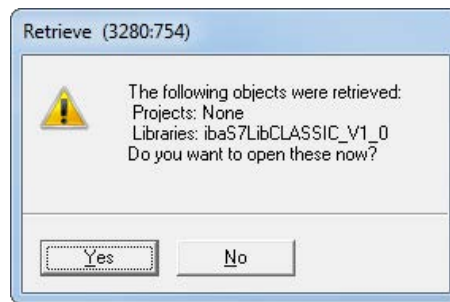
- Choose the archive file of the iba S7 library and select a storage location for the extracted library in a next step:



- Confirm the message for a successful extraction.



- Open the library by confirming the following dialog with <YES>.

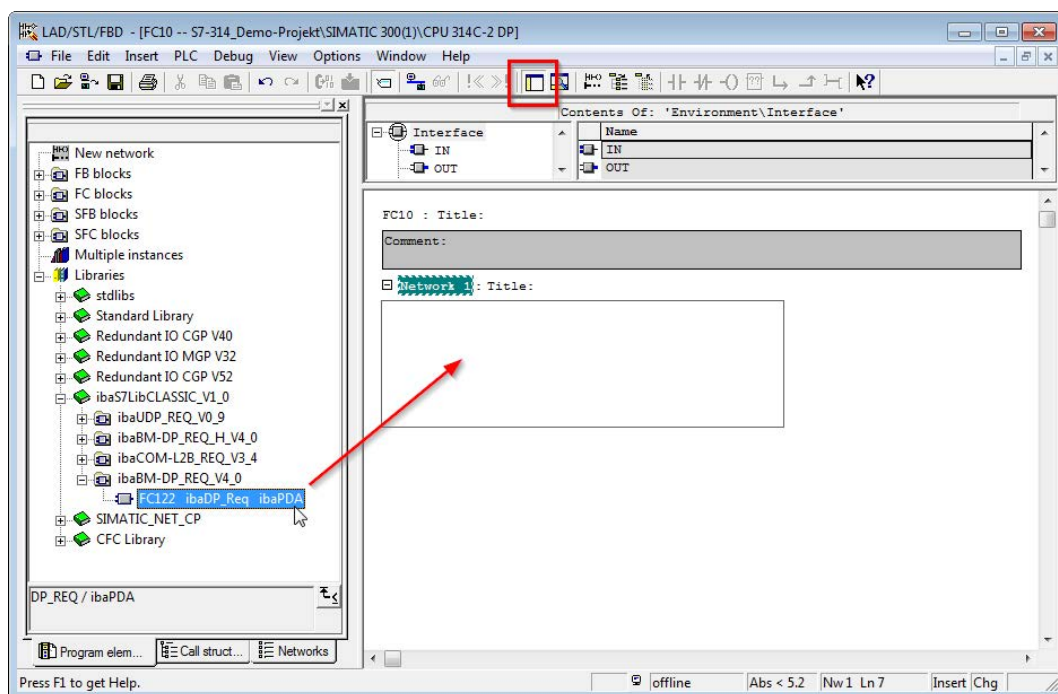


Now, the library is integrated and can be closed again.

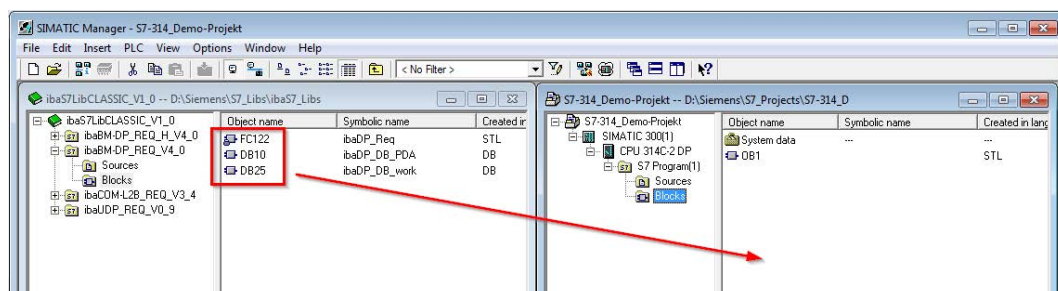
6.1.1.2 Using the blocks

There are two options for using the blocks from the library.

1. Display the block library on the left and drag the required blocks to the opened destination block.



2. Open the library via *File - Open - Libraries*, and the required destination project in parallel. With the <Tile horizontal> button, both projects can be displayed side by side. The blocks can be copied either via drag or copy & paste.



The blocks may now be used in the destination project.

6.1.2 iba S7 library for SIMATIC TIA portal

The iba S7 library for SIMATIC TIA Portal („ibaS7LibTIA_Vx_y") is suited for the use with SIMATIC TIA Portal. It contains the Request blocks described in the manual, which are required for the use of ibaPDA-Request-S7-DP/PN.

You find the iba S7 library as project archive on the DVD "iba Software & Manuals" under
 \04_Libraries_and_Examples\10_Libraries\01_SIMATIC_S7\

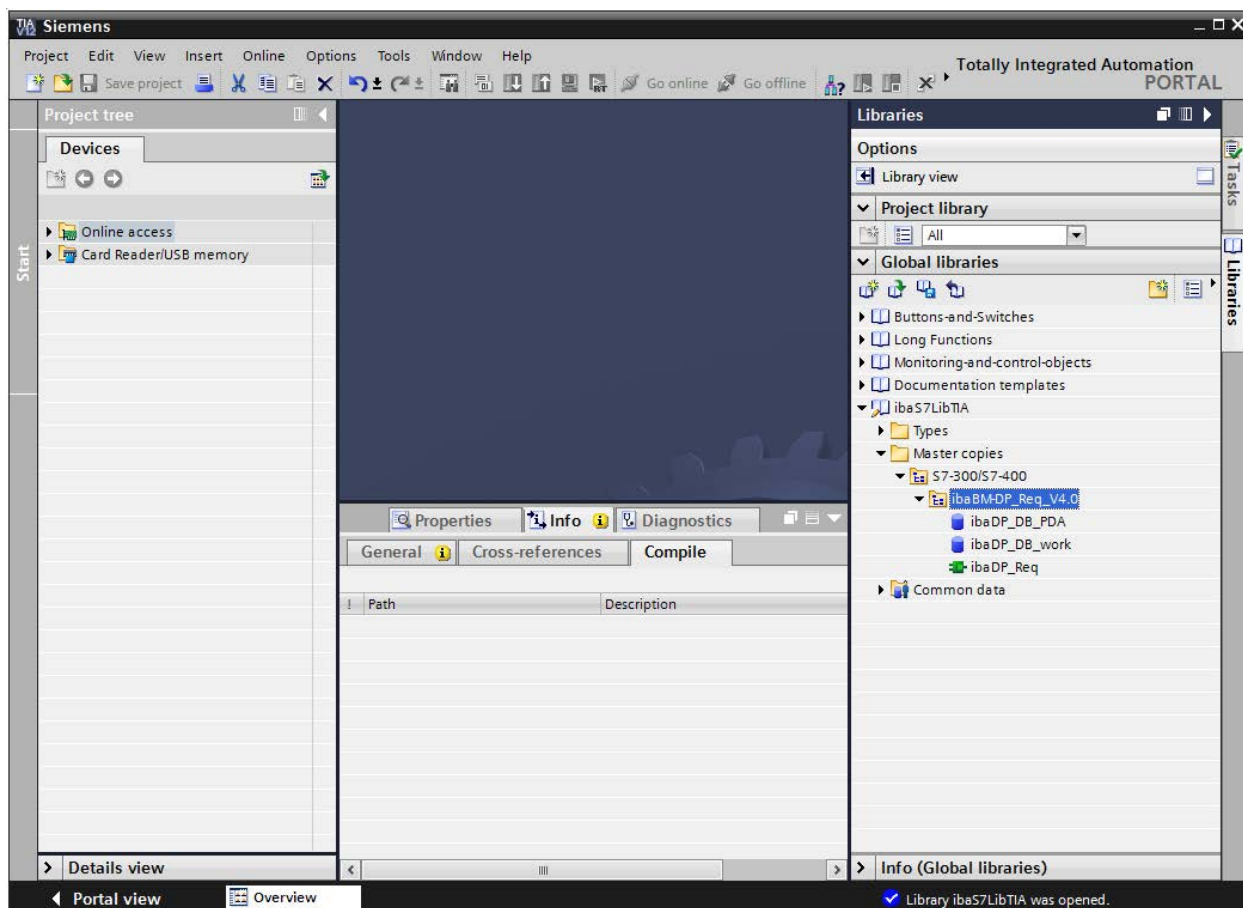


Fig. 74: iba S7 library for SIMATIC TIA portal

The following components are included:

Target platform S7-300, S7-400, WinAC

iba connection	Block name	Block no.	Note
ibaBM-DP	ibaDP_Req	FC122	
	ibaDP_DB_PDA	DB10	
	ibaDP_DB_work	DB25	
ibaBM-PN	ibaREQ_M	FB140	
	ibaREQ_PN	FB141	
	ibaREQ_PNdev	FB150	
	ibaREQ_DB	DB15	
	ibaUDT_UDPact	UDT145	

iba connection	Block name	Block no.	Note
ibaCom-L2B	ibaL2B_Init	FC111	
	ibaL2B_Req	FC112	
	ibaL2B_Req_CP	FC113	only necessary when using a CP342-5 instead of the FC112
	ibaL2B_DB_work	DB22	
	ibaL2B_DB_Struct	UDT22	
	ibaL2B_CP_SNDRCV	DB10	only necessary when using a CP342-5
ibaPDA-Interface-S7-TCP/UDP	ibaREQ_M	FB140	
	ibaREQ_UDPact	FB145	
	ibaREQ_UDPint	FB146	
	ibaREQ_UDPext3	FB147	
	ibaREQ_UDPext4	FB148	
	ibaREQ_DB	DB15	
	ibaUDT_UDPact	UDT145	

Target platform S7-1500

iba connection	Block name	Block no.	Note
ibaBM-DP	ibaREQ_M	FB1400	
	ibaREQ_DP	FB1402	
	ibaREQ_DB	DB15	
	ibaREQ_DB-Interface		
ibaBM-PN	ibaREQ_M	FB1400	
	ibaREQ_PN	FB1401	
	ibaREQ_DB	DB15	
	ibaREQ_DB-Interface		
ibaPDA-Interface-S7-TCP/UDP	ibaREQ_M	FB1400	
	ibaREQ_UDP	FB1405	only for TIA V13 and V14
	ibaREQ_UDP2	FB1406	
	ibaREQ_UDPact	FB1410	
	ibaREQ_DB	DB15	
	ibaREQ_DB-Interface		

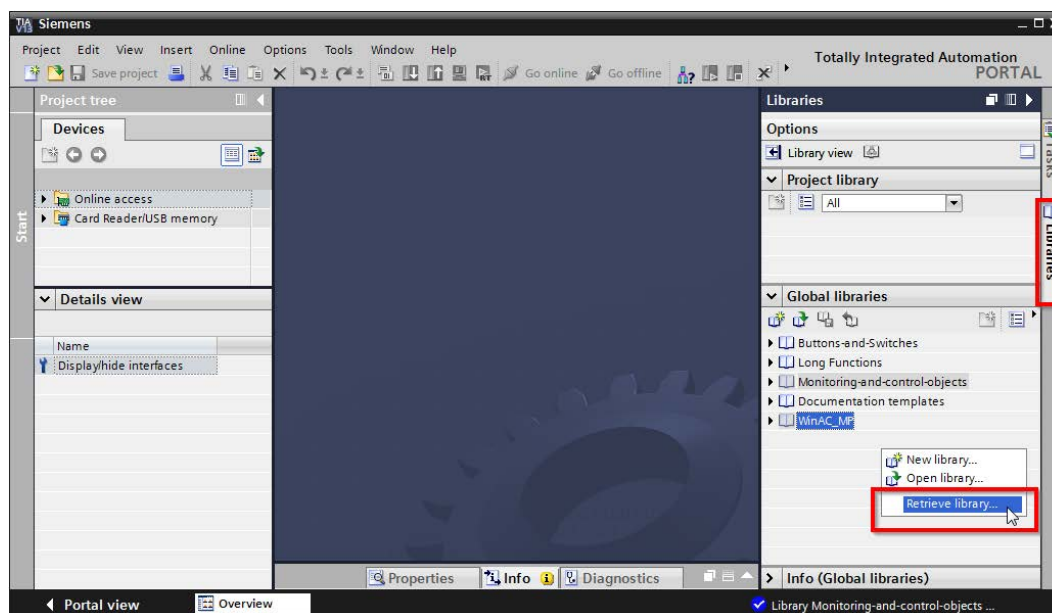
Note

TIA portal libraries are version-dependent. There may be upward compatibility depending on the TIA portal version.

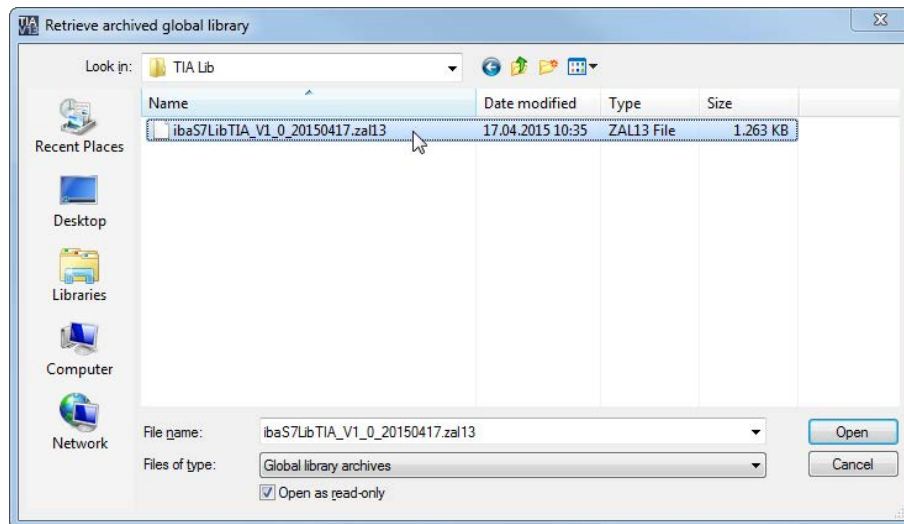
6.1.2.1 Integrating the library

For integrating the library, you have to retrieve it in the TIA portal. Copy the iba S7 library to a local directory of your computer, where the TIA Portal is executed.

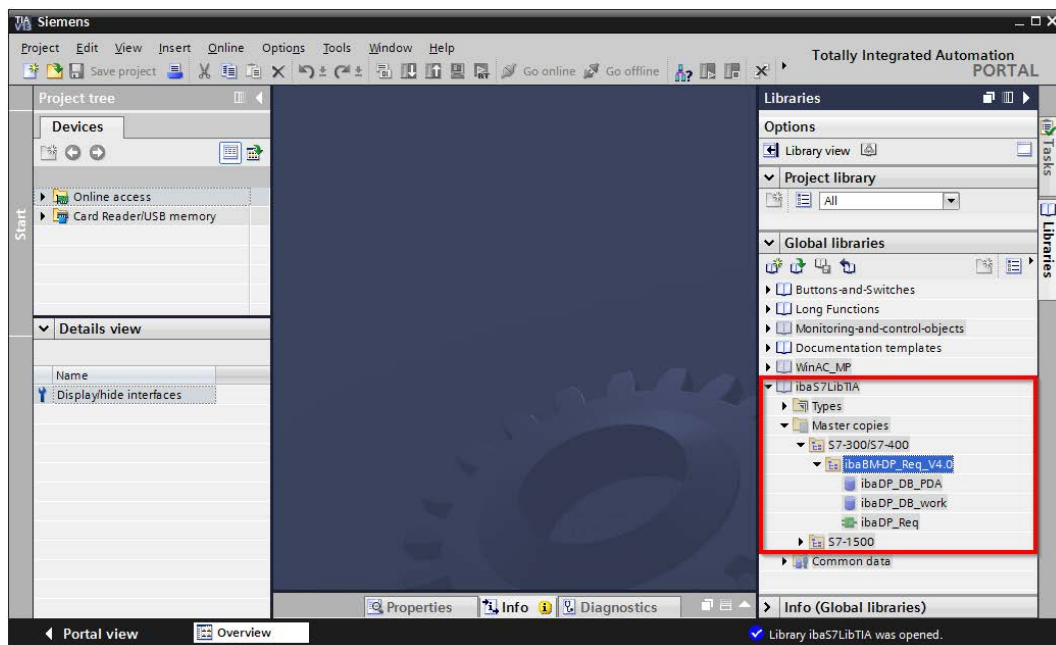
- Select the *Retrieve library...* command on the *Libraries* tab in the context menu (right mouse-button).



- Select the archive file of the iba S7 library and select a storage location for the extracted library in a next step:



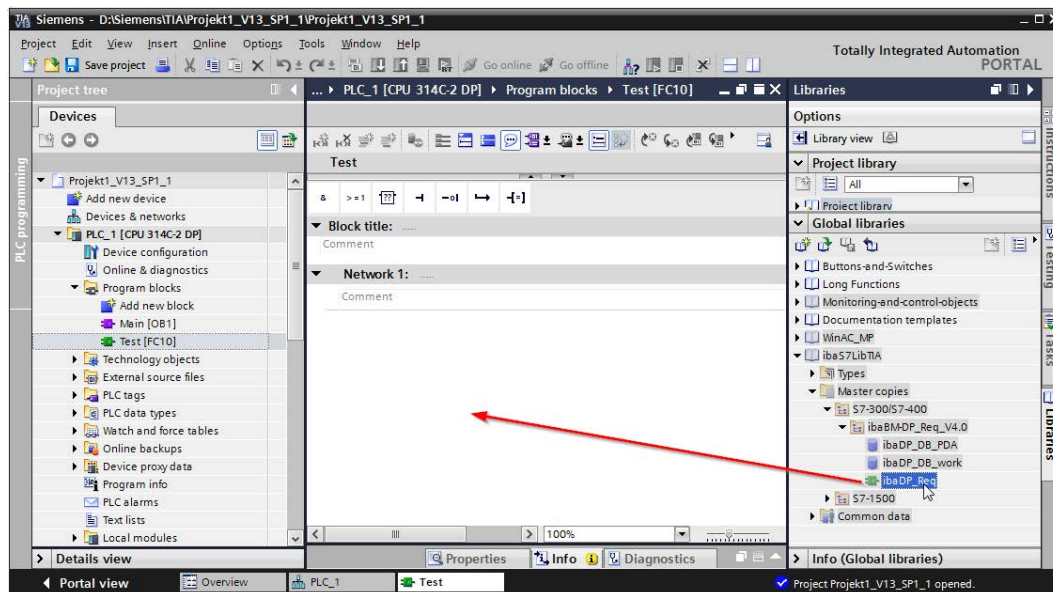
- Now, the library is integrated.



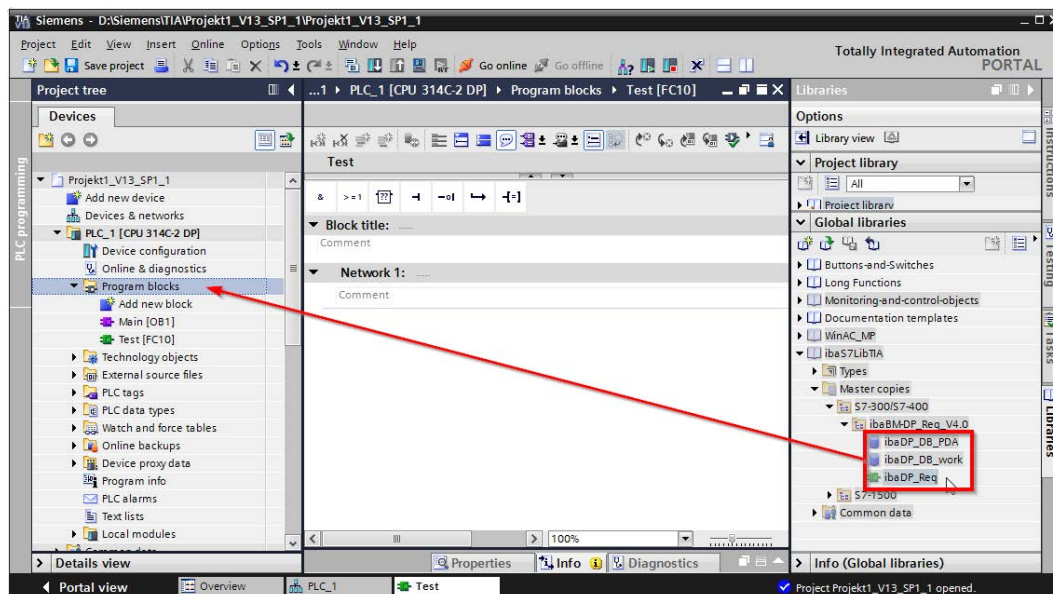
6.1.2.2 Using the blocks

There are two options for using the blocks from the library.

1. Display the library and drag the required block to the opened destination block.

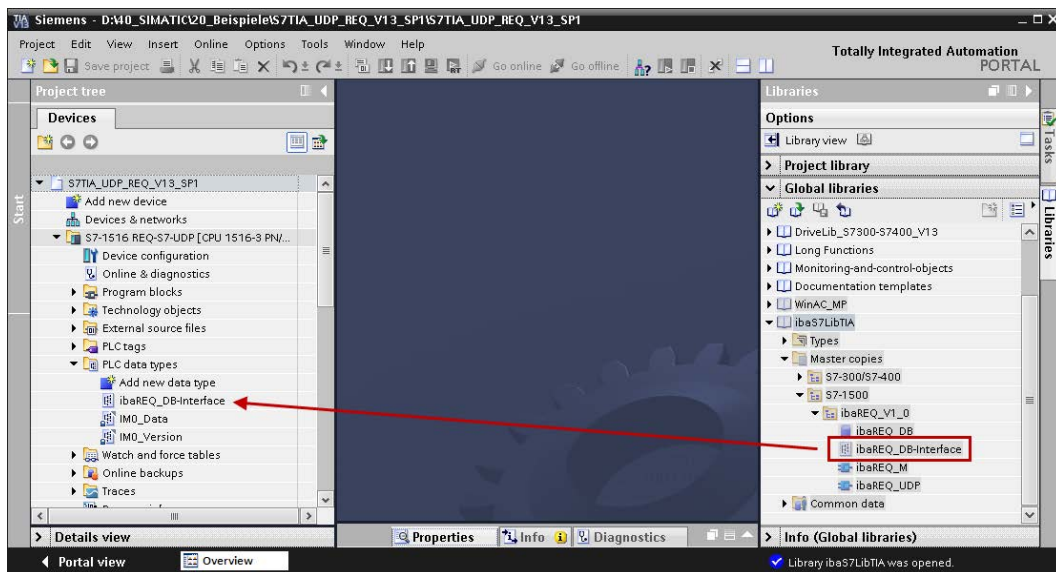


2. Drag or copy the blocks to the program blocks directory in the project navigation.



The blocks can now be called within a project block.

3. Drag or copy the PLC data type to the PLC data type directory in the project navigation.



6.2 Application examples

You find application examples on the DVD "iba Software & Manuals".

- For the ibaBM-DPM-S device in the path
 \04_Libraries_and_Examples\30_ibaBM-DPM-S\01_SIMATIC_S7\Request-S7\
- For the ibaBM-DP device in the path
 \04_Libraries_and_Examples\31_ibaBM-DP\01_SIMATIC_S7\Request-S7\
- For the ibaBM-PN device in the path
 \04_Libraries_and_Examples\32_ibaBM-PN\01_SIMATIC_S7\Request-S7\
- For the ibaCom-L2B card in the path
 \04_Libraries_and_Examples\40_ibaCOM-L2B\01_SIMATIC_S7\Request-S7\

Examples are available for the following configurations:

iba	S7-CPU	S7 project	ibaPDA project
ibaBM-PN	S7-300 PN-IF	S7CLASSIC_ PN_REQ_Vxx.zip	ibaPDA_S7CLASSIC_ PN_REQ_Vxx.zip
	S7-400 PN-IF		
	S7-1500 PN-IF	S7TIA_ PN_REQ_Vxx.zip	ibaPDA_S7TIA_ PN_REQ_Vxx.zip

iba	S7-CPU	S7 project	ibaPDA project
ibaBM-DP	S7-300 DP-IF	S7CLASSIC_ DP_REQ_Vxx.zip	ibaPDA_S7CLASSIC_ DP_REQ_CPU3xx_Vxx.zip
	S7-400 DP-IF		ibaPDA_S7CLASSIC_ DP_REQ_CPU4xx_Vxx.zip
	S7-400 + CP443-5		ibaPDA_S7CLASSIC_ DP_REQ_CPU4xx_Vxx.zip
	WinAC		ibaPDA_S7CLASSIC_ DP_REQ_WinAC_Vxx.zip
	S7-400 with CFC	S7CLASSIC_ DP_REQ_CFC_Vxx.zip	ibaPDA_S7CLASSIC_ DP_REQ_CFC_Vxx.zip
	S7-400H	S7CLASSIC_ DP_REQ_H_Vxx.zip	ibaPDA_S7CLASSIC_ DP_REQ_CPU4xxH_Vxx.zip
	S7-300 DP-IF	S7CLASSIC_ L2B_REQ_Vxx.zip	ibaPDA_S7CLASSIC_ DP_L2Bcomp_REQ_Vxx.zip
	S7-1500 DP-IF	S7TIA_ DP_REQ_Vxx.zip	ibaPDA_S7TIA_ DP_REQ_Vxx.zip
ibaBM-DPM-S	S7-300 DP-IF	S7CLASSIC_ DP_REQ_Vxx.zip	ibaPDA_S7CLASSIC_ DPMS_REQ_CPU3xx_Vxx.zip
	S7-400 DP-IF		ibaPDA_S7CLASSIC_ DPMS_REQ_CPU4xx_Vxx.zip
	WinAC		ibaPDA_S7CLASSIC_ DPMS_REQ_WinAC_Vxx.zip
	S7-400 with CFC	S7CLASSIC_ DP_REQ_CFC_Vxx.zip	ibaPDA_S7CLASSIC_ DPMS_REQ_CFC_Vxx.zip
	S7-400H	S7CLASSIC_ DP_REQ_H_Vxx.zip	ibaPDA_S7CLASSIC_ DPMS_REQ_CPU4xxH_Vxx.zip
ibaCom-L2B	S7-300 DP-IF	S7CLASSIC_ L2B_REQ_Vxx.zip	ibaPDA_S7CLASSIC_ L2B_REQ_Vxx.zip
	S7-300 + CP342-5		
	S7-400 with CFC	S7CLASSIC_ L2B_REQ_CFC_Vxx.zip	ibaPDA_S7CLASSIC_ L2B_REQ_CFC_Vxx.zip

Table 13: Configuration examples available on DVD

6.3 S7 cycle time measurements

6.3.1 ibaCom-L2B

The following tables provide information on the cycle time extension you can expect for *ibaPDA* in conjunction with L2B-Request depending on the S7-CPU type. Test series with up to 256 analog and 256 digital signals (8 slaves) were used to obtain the time values.

6.3.1.1 S7-CPU with external DP interface (CP)

S7-CPU type/ CP	Max. cycle time extension per slave (32 analog + 32 digital signals)
316- 2AG00	13 ms
314- 6CF00	8.63 ms
315- 2AG10	6.88 ms
317- 2EJ10	0.875 ms
317- 6FF00	1.125 ms
318- 2AG00	0.75 ms
416- 2XK00	1.25 ms
413- 2XG02	2.625 ms
414- 2XG03	0.875 ms
416- 2XK04	0.375 ms

6.3.1.2 S7-CPU with external DP interface (CP)

S7-CPU type/ CP	Max. cycle time extension per slave (32 analog + 32 digital signals)
CPU315 with CP342- 5	6.0 ms
CPU318 with CP342- 5	0.75 ms
CPU416 with CP443- 5	0.625 ms

Note



When using the external PROFIBUS-CP CP342-5 with the S7-300, a cycle-precise measurement is usually not possible, since the data transmission over the back-plane bus of the S7-300 is too slow. Otherwise, precise cycle measurements are possible if the PROFIBUS cycle is sufficiently fast.

6.3.2 ibaBM-DP

The following tables provide information about the code runtimes that the request blocks require when using *ibaPDA* in combination with Request-S7 for *ibaBM-DP*.

The measurement values were determined in a test environment and only serve as reference points. The values may deviate in other system environments.

SIMATIC S7 CPU	Number of signals	Data amount	ibaDP_Req FC122
CPU412-2 PN 6ES7 412-2EK06-0AB0	1 INT + 0 BOOL (1 Pointer)	2 Byte	564 µs
	59 REAL + 64 BOOL (2 Pointers)	244 Byte	1614 µs
	59 REAL + 64 BOOL (123 Pointers)	244 Byte	1632 µs

SIMATIC S7 CPU	Number of signals	Data amount	ibaREQ_M FB1400	ibaREQ_DP FB1402
CPU1516-3 PN/DP 6ES7 516-3AN00-0AB0	1 INT + 0 BOOL (1 Pointer)	2 Byte	192 µs	286 µs
	59 REAL + 64 BOOL (2 Pointers)	244 Byte	194 µs	296 µs
	59 REAL + 64 BOOL (123 Pointers)	244 Byte	192 µs	834 µs
	122 INT + 0 BOOL (1 Pointer)	244 Byte	195 µs	287 µs
	122 INT + 0 BOOL (122 Pointer)	244 Byte	198 µs	706 µs

6.3.3 ibaBM-PN

The following tables provide information about the code runtimes that the request blocks require when using *ibaPDA* in combination with Request-S7 for *ibaBM-PN*.

The measurement values were determined in a test environment and only serve as reference points. The values may deviate in other system environments.

SIMATIC S7 CPU	Number of signals	Data amount	ibaREQ_M FB140	ibaREQ_PN FB141
CPU412-2 PN 6ES7 412-2EK06-0AB0	1 INT + 0 BOOL (1 Pointer)	2 Byte	128 µs	302 µs
	59 REAL + 64 BOOL (2 Pointers)	244 Byte	126 µs	376 µs
	59 REAL + 64 BOOL (123 Pointers)	244 Byte	132 µs	937 µs
	122 INT + 0 BOOL (1 Pointer)	244 Byte	126 µs	342 µs
	122 INT + 0 BOOL (122 Pointer)	244 Byte	132 µs	954 µs

SIMATIC S7 CPU	Number of signals	Data amount	ibaREQ_M FB1400	ibaREQ_PN FB1401
CPU1516-3 PN/DP 6ES7 516-3AN00-0AB0	1 INT + 0 BOOL (1 Pointer)	2 Byte	198 µs	276 µs
	59 REAL + 64 BOOL (2 Pointers)	244 Byte	194 µs	282 µs
	59 REAL + 64 BOOL (123 Pointers)	244 Byte	192 µs	806 µs
	122 INT + 0 BOOL (1 Pointer)	244 Byte	195 µs	274 µs
	122 INT + 0 BOOL (122 Pointer)	244 Byte	198 µs	672 µs

6.4 Setting PG/PC interface / defining new access point

ibaPDA-Request-S7-DP/PN cannot establish a connection to a S7-CPU, if the parametrization "AUTO" for an access point (MPI-adapter or CP) has been set in the SIMATIC Manager.

There are 2 possible remedies:

Changing the interface with remaining access point name

Change interface in the SIMATIC Manager e.g. from "CP5622 (AUTO)" to "CP5622 (MPI)" or "CP5622 (PROFIBUS)".

Disadvantage of this method: If the setting of the access point will be changed again in the SIMATIC Manager, the measurement does no longer work because *ibaPDA* no longer has any access.

Adding a special access point for *ibaPDA*

To avoid conflicts with the setting of SIMATIC Manager and *ibaPDA* when both programs run on the same computer, a new access point should be defined.

There is the <Configure PG/PC interface> button in the dialog window of the PC/CP module. It can be used to open the dialog for configuring the PG/PC interface.

The setting for the SIMATIC Manager will also be changed.

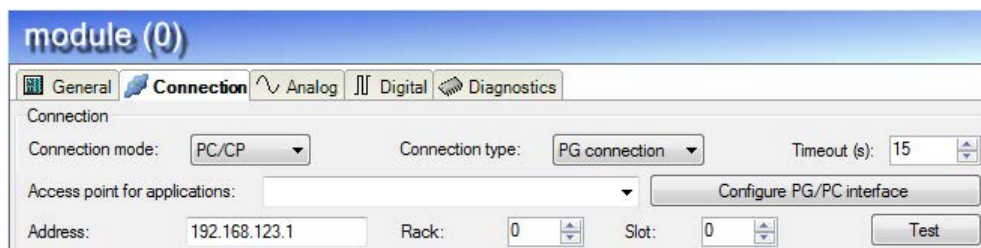
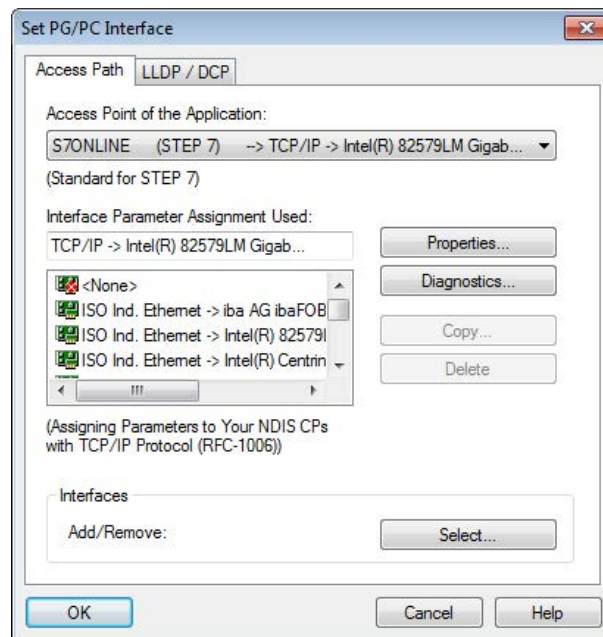


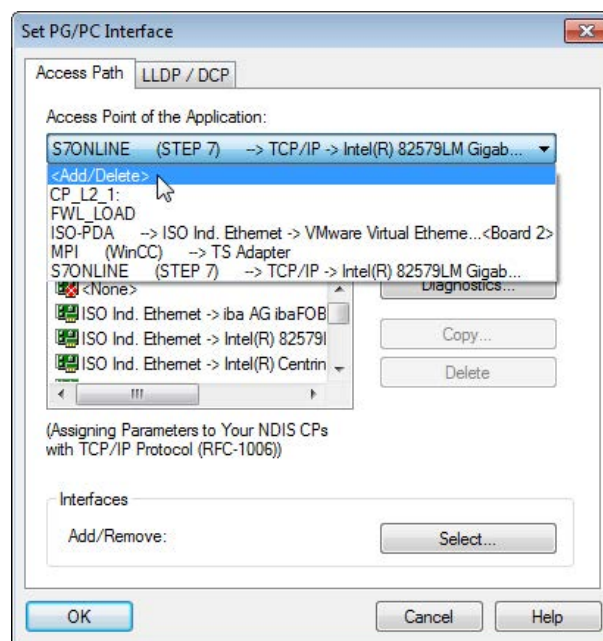
Fig. 75: Configure PG/PC interface

Procedure

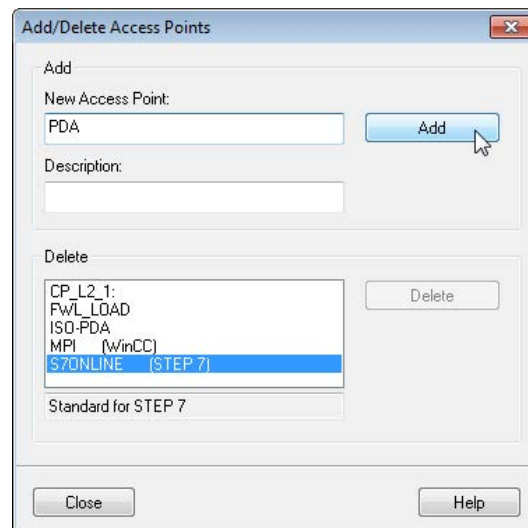
1. Open the dialog box with the <Configure PG/PC interface> button.



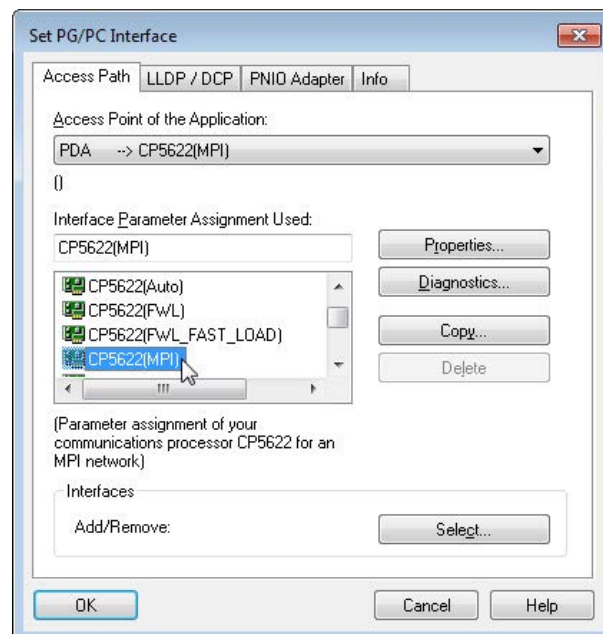
2. Select the row <Add/Delete> under "Access Point of the Application".



3. Define a new access point; enter a name, e.g. PDA and optionally a description for a better understanding, click on <Add> and <Close>.



4. Add an interface to the access point, e.g. "CP5622 (MPI)" and exit with <OK>.



The newly defined access (e.g. PDA --> CP5622.MPI.1) is displayed subsequently in the connection dialog of *ibaPDA* under "Access points for applications".

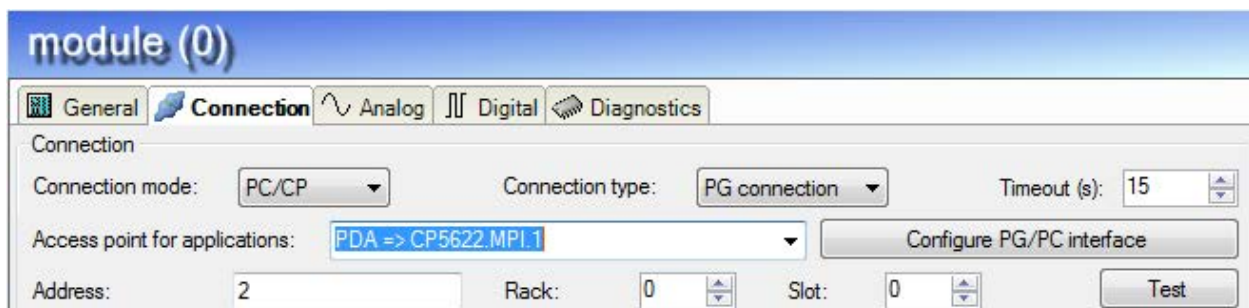


Fig. 76: Set access point

Notes on the different access points

Depending on which access points have been configured in the Engineering PC, there are different access points available for selection in the *ibaPDA* system.

Basically, there are 3 types of access points:

- TCP/IP
- ISO
- Bus system PROFIBUS or MPI

TCP/IP

If you select an access point using TCP/IP, you need to enter the IP address, rack and slot number of the CP in the module configuration dialog. If you do not know the rack and/or slot number, enter "0" for slot and click on the <Test> button.

ISO

If you select an access point using an ISO interface, you need to enter the MAC address, rack and slot number. For the rack and slot number, use the <Search> button.

Bus system (PROFIBUS or MPI)

If you select an access point using a bus interface, like e.g. PROFIBUS or MPI, you need to enter the bus address, the rack and slot number. You can also use the <Search> button and then click on one of the CPU links found to test the connection.

6.5 S7 routing

S7 routing is defined as the possibility to use S7 controls as router to access secondary target systems, i.e. controls or drives, which are in different subnets. This also includes changing the bus system (Ethernet / PROFIBUS / MPI).

6.5.1 Routing from Ethernet to Ethernet

Please do not mix up the “S7 Routing” function with “IP Routing”.

The following constellation will make this clear:

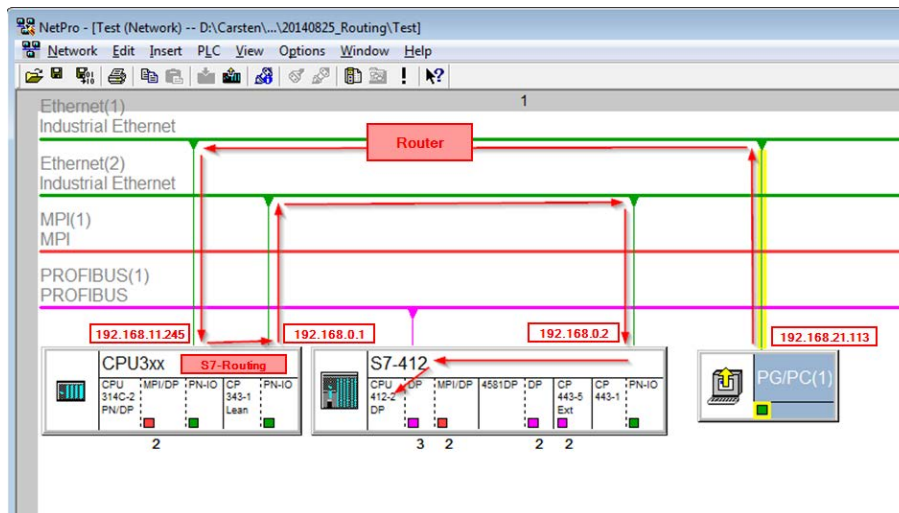


Fig. 77: S7 Routing, example system topology

We want to access the CPU412 controller from the engineering PC (also with *ibaPDA*). The computer and the controller are not directly connected via a common network/bus. We want to run the connection over the CPU314C controller. “Passing” the communication in this controller is called “S7 Routing”.

In our example, engineering PC and CPU314C are also located in two different (logic) subnets. You need an (IP-) Router for establishing a communication connection. This is completely independent of the “S7 Routing” function and should not be confounded with it.

6.5.1.1 Configuration of STEP 7/ NetPro

The following configuration steps are required to be able to access the secondary CPU412 control with the SIMATIC STEP 7 programming software. These are not required for using *ibaPDA*.

Inserting a PG/PC station:

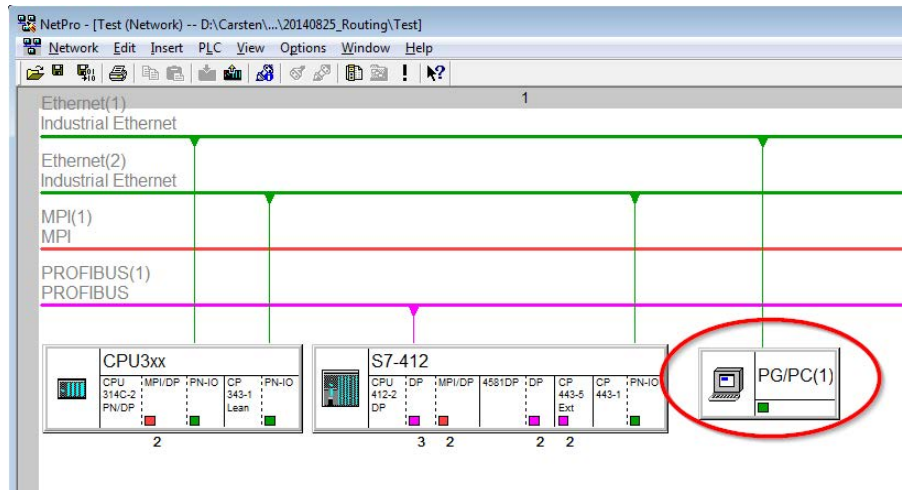


Fig. 78: NetPro configuration

Assigning an interface (network interface card):

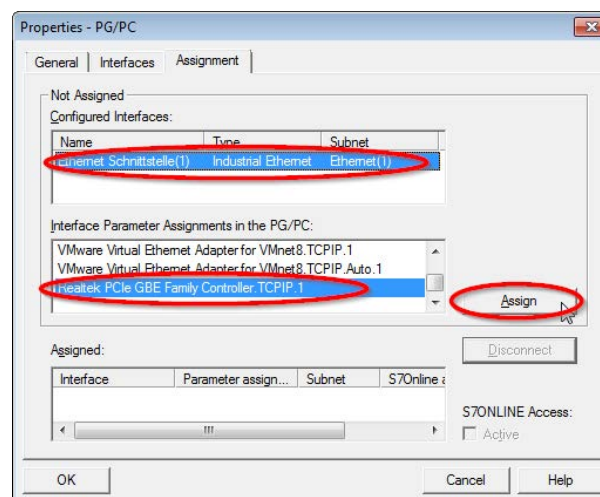


Fig. 79: PG/PC interface assigned

Result:

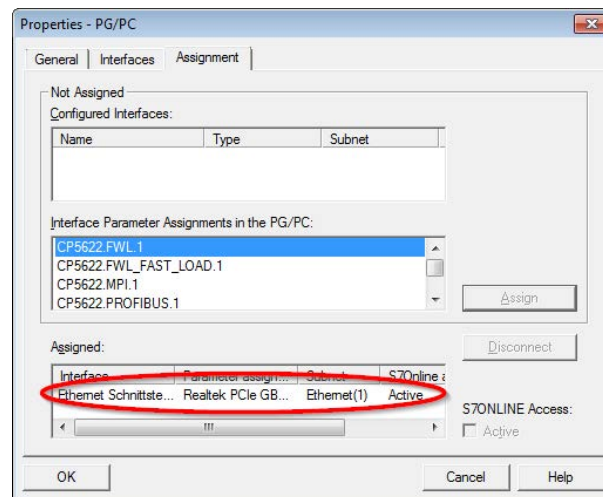


Fig. 80: Interface assigned

The connection line of PG/PC to the network should be highlighted yellow now.

The following figure shows the communication channel using arrows (these are not shown in SIMATIC NetPro).

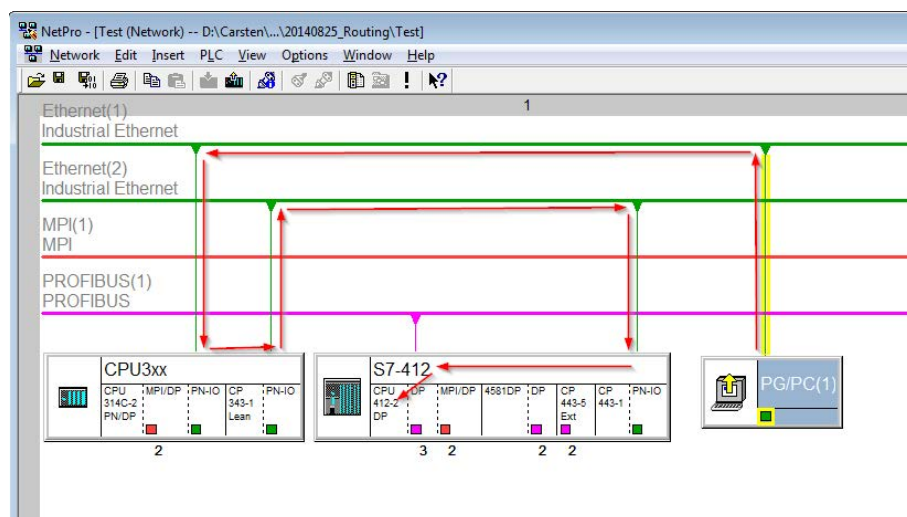


Fig. 81: Communication channel

Finally, load all hardware configurations and connection data from NetPro.

6.5.1.2 ibaPDA configuration

The following entries have to be made:

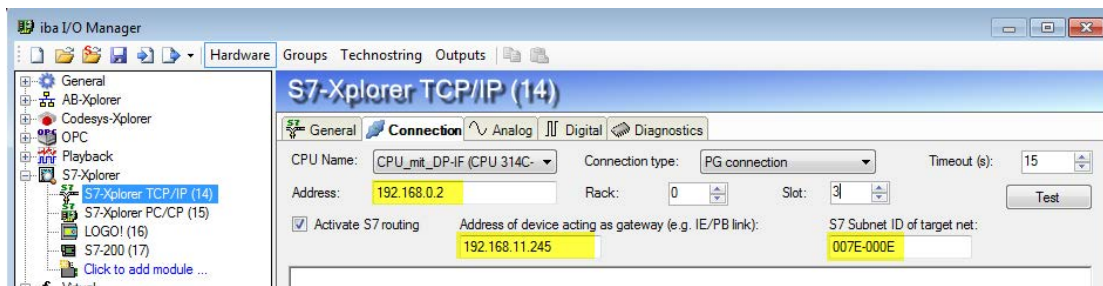


Fig. 82: Enable S7 routing

Activate S7 routing

Enable to use S7 routing

Address

Address of the target control (here CPU412)

Address of device acting as gateway

Enter address of the gateway (here CPU314C)

S7 subnet ID of target net

Enter subnet ID from STEP 7 NetPro

You can identify the S7 subnet ID in NetPro. For doing so, right-click on the secondary bus system and open the "Properties".

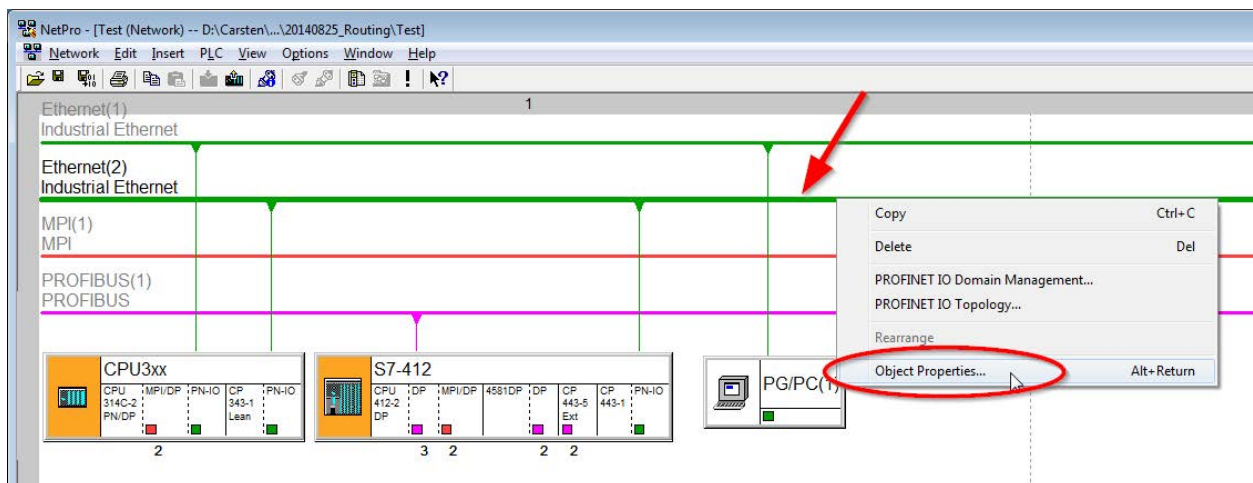


Fig. 83: Determine S7 subnet ID

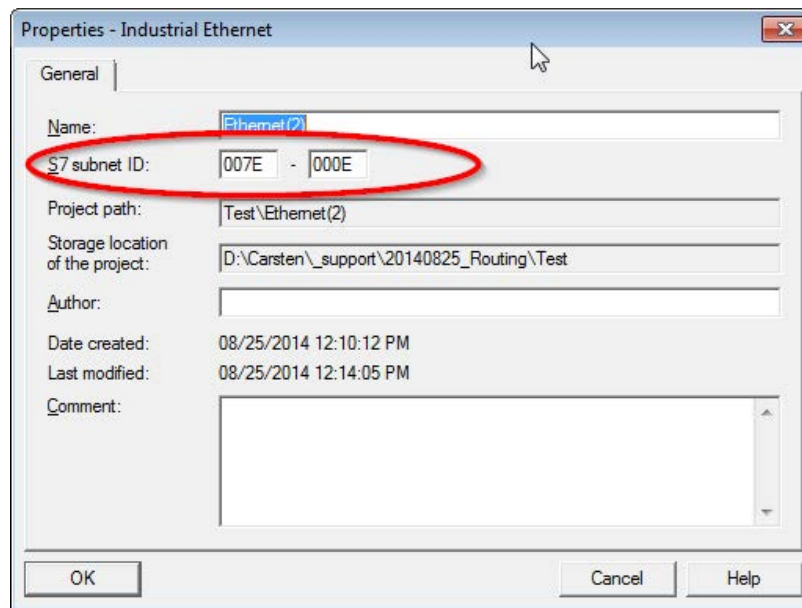


Fig. 84: S7 subnet ID

6.5.2 Routing from Ethernet to PROFIBUS

We want to implement the following way of access:

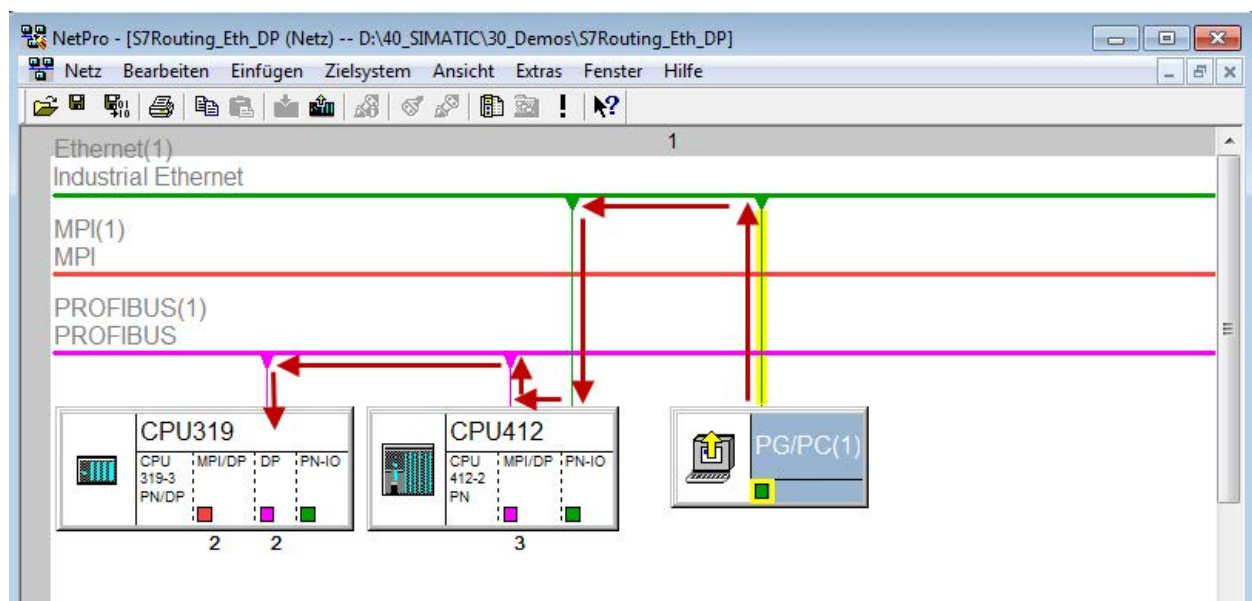


Fig. 85: S7 Routing, example system topology Ethernet PROFIBUS

We want to access the CPU319 controller from the engineering PC (also with *ibaPDA*). The computer and the controller are not directly connected via a common network/bus. We want to run the connection over the CPU412 controller. “Passing” the communication in this controller is called “S7 Routing”.

6.5.2.1 Configuration STEP 7/ NetPro

The following configuration steps are exclusively required for accessing the subordinate controller CPU319 via the SIMATIC STEP 7 programming software. For using *ibaPDA*, these configuration steps are not required. Adding a PG/PC station:

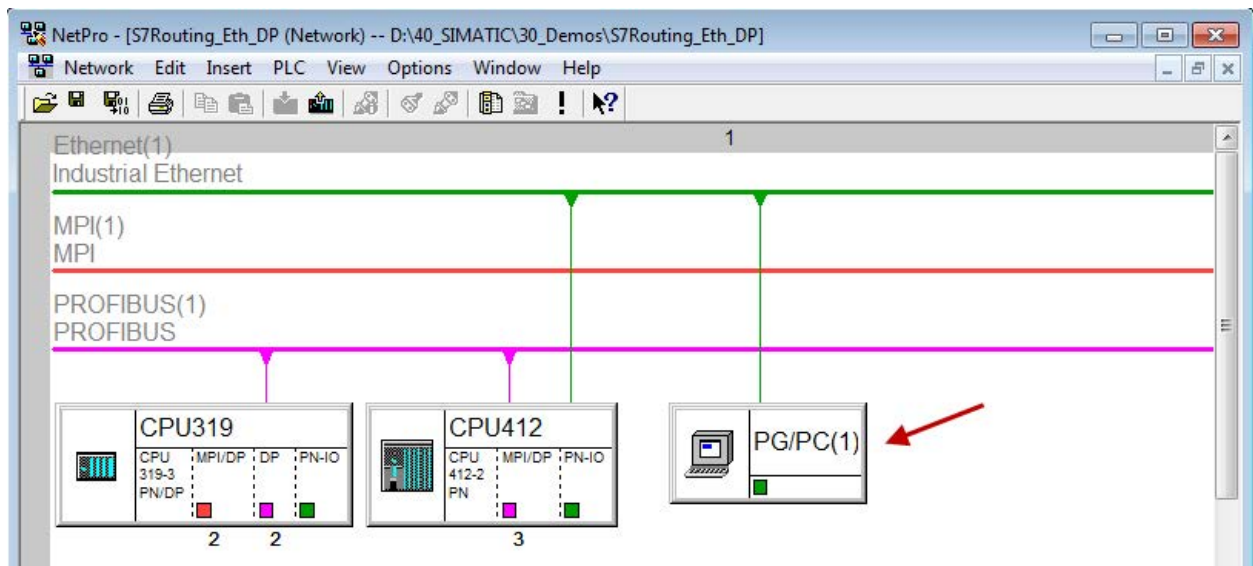


Fig. 86: Configuration NetPro

Assigning an interface (network card):

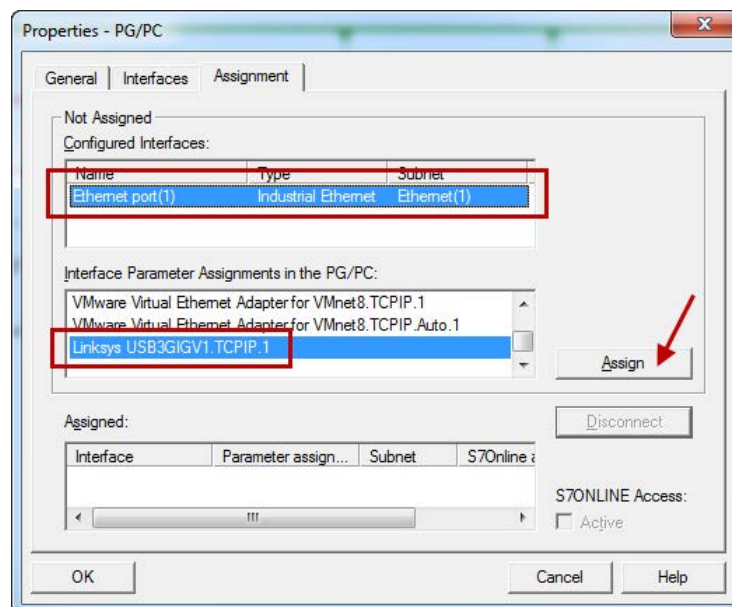


Fig. 87: Assign PG/PC interface

Result:

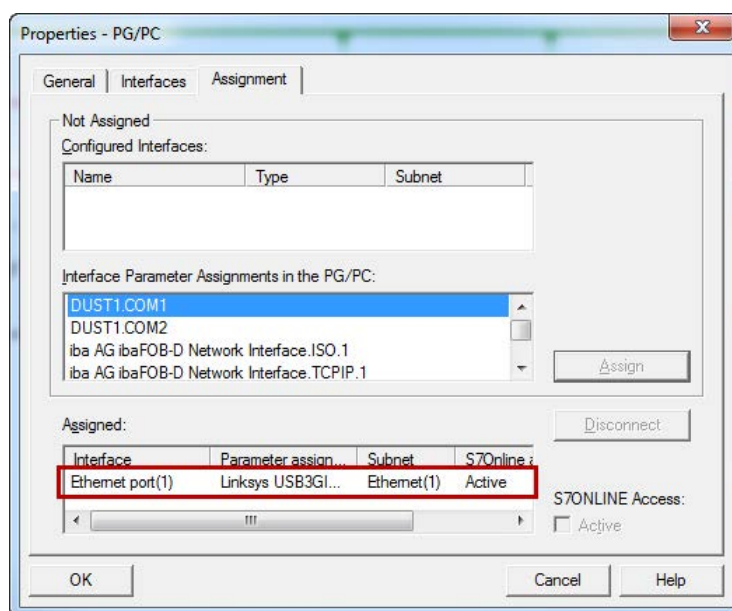


Fig. 88: Interface has been assigned

Now, the connection line from PG/PC to the network has to be marked in yellow. In the following figure, the communication path is shown using arrows (these are not displayed in SIMATIC NetPro).

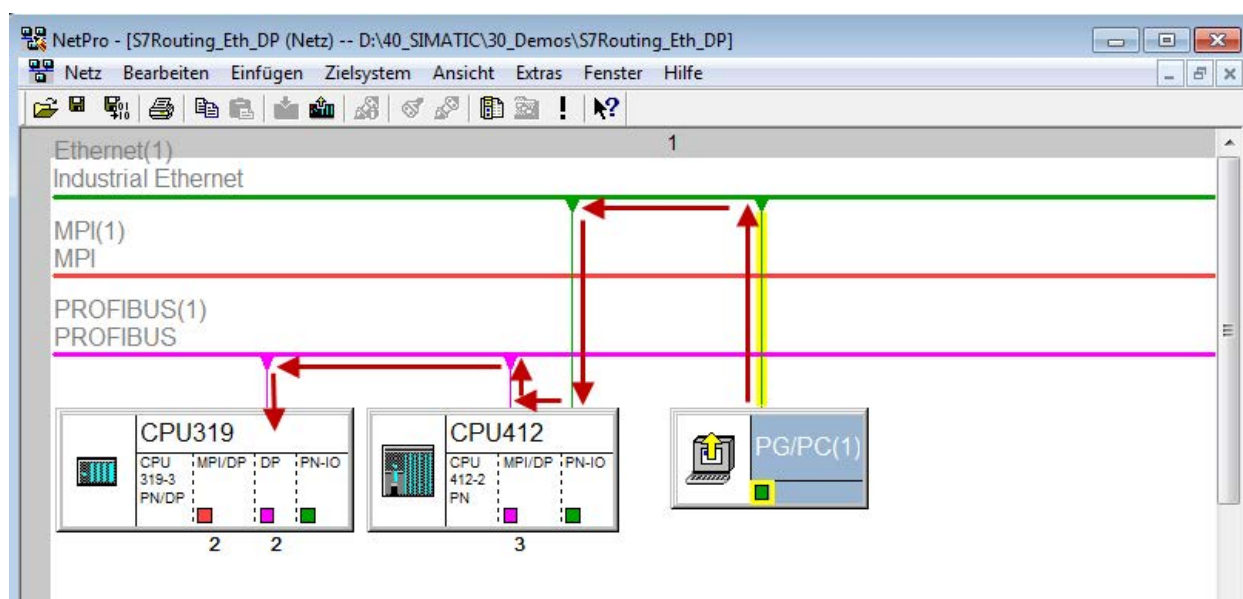


Fig. 89: Communication path

Finally, all HW configurations and connection data are loaded from NetPro.

6.5.2.2 ibaPDA configuration

The following entries have to be made:

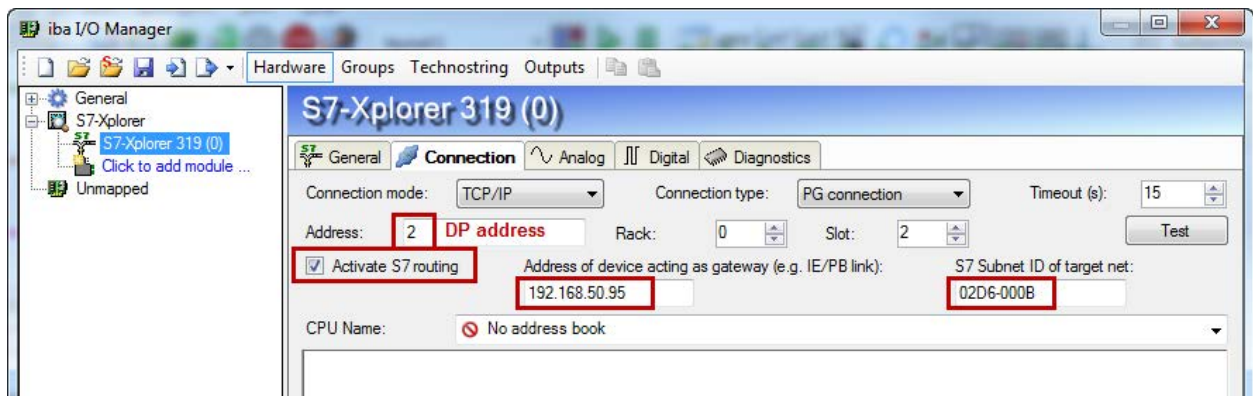


Fig. 90: Enable S7 routing

Activate S7 routing

Enable to use S7 routing

Address

Address of the target control (here CPU319)

Address of device acting as gateway

Enter address of the gateway (here CPU412)

S7 subnet ID of target net

Enter subnet ID from STEP 7 NetPro

You can identify the S7 subnet ID in NetPro. For doing so, right-click on the secondary bus system and open the "Properties".

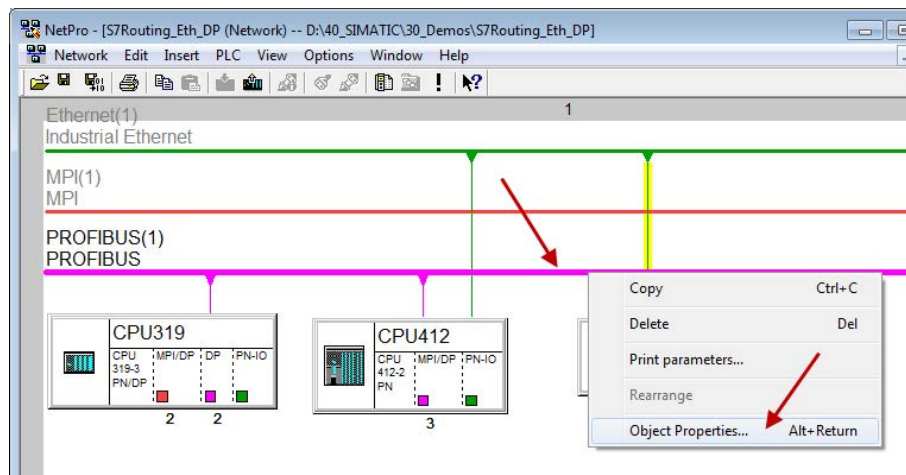


Fig. 91: Determine S7 subnet ID

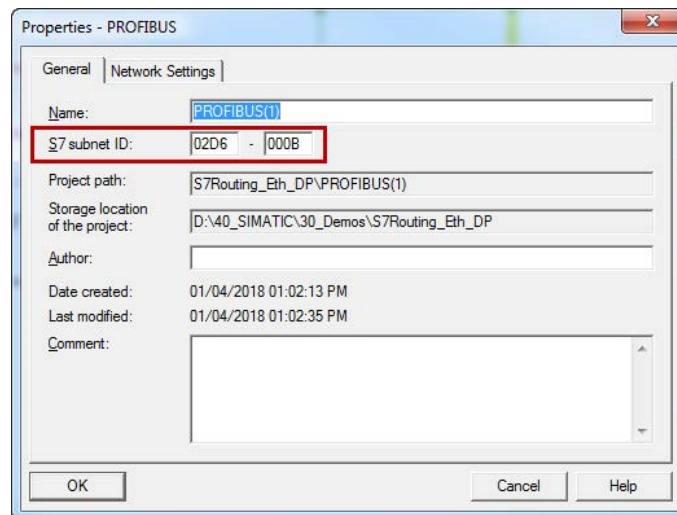


Fig. 92: S7 subnet ID

Reference



For more information about the S7-Routing, please see:

Which modules support the "S7 Routing" function in S7 subnets?

<http://support.automation.siemens.com/ww/view/en/584459>

Which requirements must be fulfilled and what do I have to observe if I want to execute routing? <https://support.industry.siemens.com/cs/ww/de/view/2383206>

How do you enable cross-project S7 Routing in the TIA Portal and in STEP 7 V5.x?

<https://support.industry.siemens.com/cs/ww/en/view/109474569>

6.6 Replacing Request-S7 on ibaCom-L2B by ibaBM-DP

A common task is the replacement of an existing Request-S7 solution based on the *ibaCom-L2B* card by an *ibaBM-DP*. This might be required in case of spare parts (the *ibaCom-L2B* card is discontinued) or when a new computer without PCI slots is used.

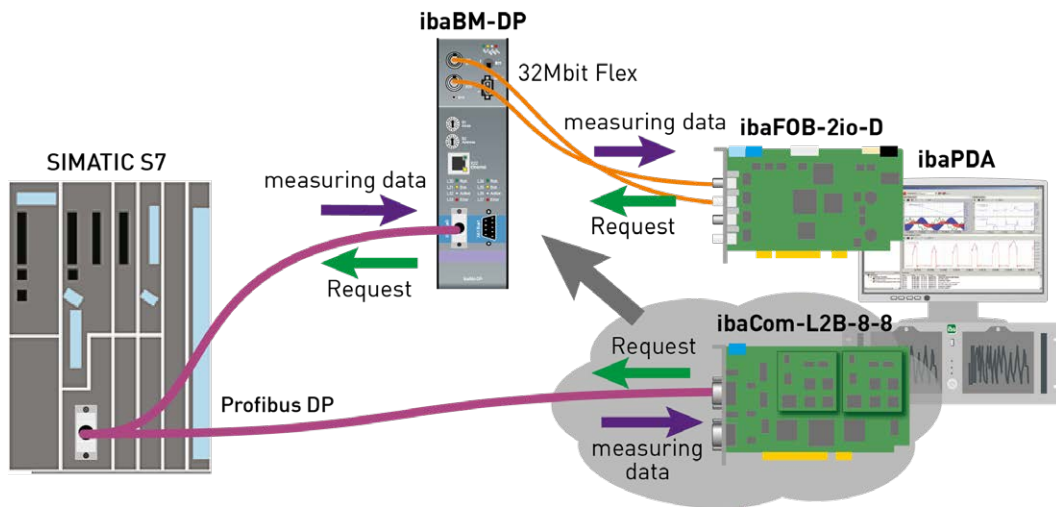


Fig. 93: Request-S7 with ibaBM-DP, replaces ibaCom-L2B

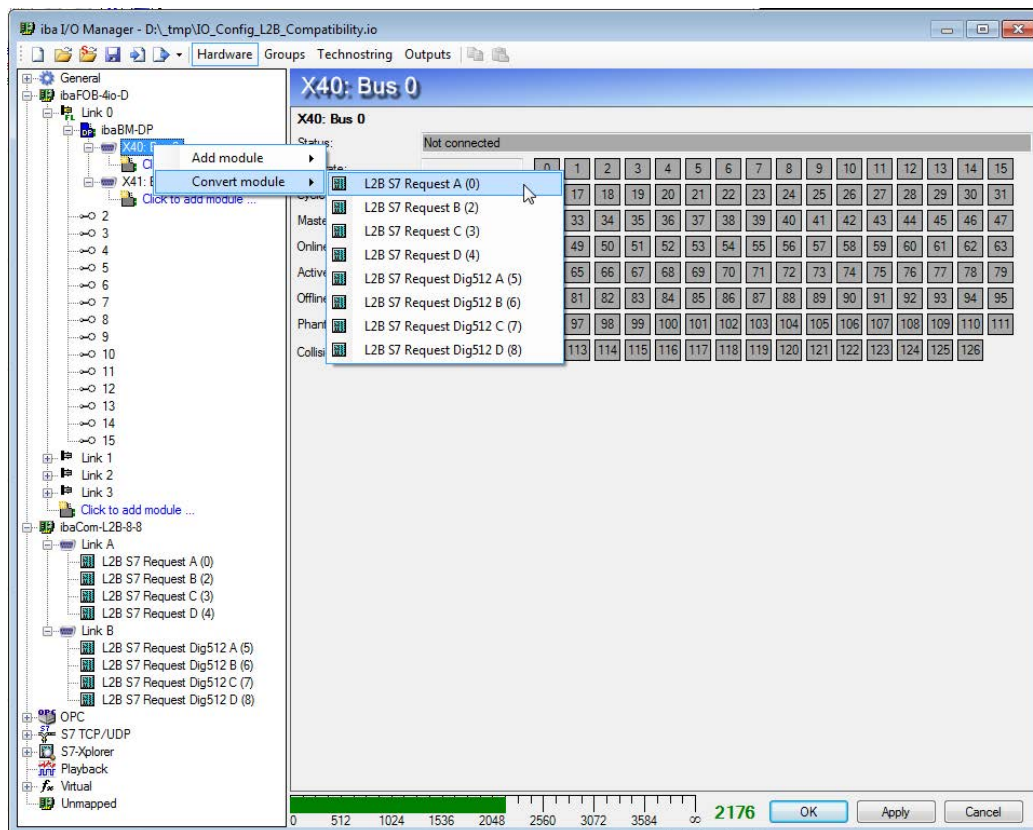
This task can be solved in an easy way by using the *ibaCom-L2B* compatible modules of *ibaBM-DP*.

No modifications in the program and the hardware configuration of the S7-CPU are required!

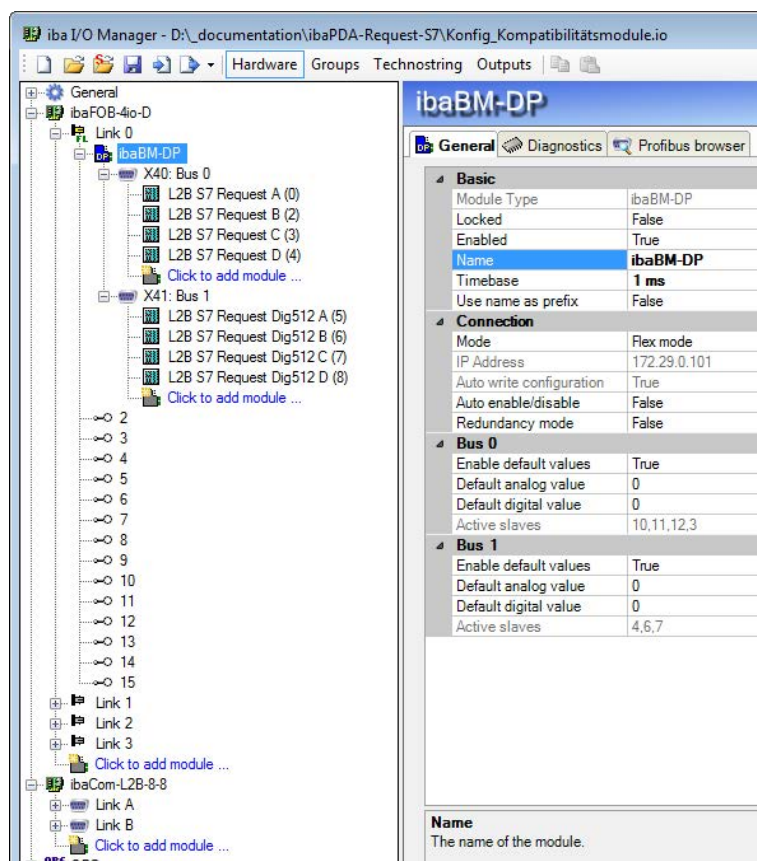
As for the *ibaCom-L2B* card also for the compatible solution with *ibaBM-DP* no network connection between *ibaPDA*-PC and the S7-CPU is required. The measurement values (Request handshake) are requested via FO and are sent via the PROFIBUS IO range.

The following steps have to be performed for the replacement:

1. If required, install a new *ibaFOB-D* card in the *ibaPDA*-PC. A card with input and output is needed (*ibaFOB-io-D*, *ibaFOB-2io-D* or *ibaFOB-4io-D*) as the compatible modules can only be used in the bidirectional 32Mbit Flex mode.
2. Connect the *ibaBM-DP* to the *ibaFOB-D* card via the bidirectional FO cable.
3. Connect the PROFIBUS connector of the *ibaCom-L2B* card to the PROFIBUS plugs of the *ibaBM-DP*.
4. In the *ibaPDA* I/O Manager, a newly installed *ibaFOB-D* card is displayed. Configure an *ibaBM-DP* device on the respective link.
5. Click with the right mouse button on the PROFIBUS links and select in the context menu "Convert module". All available L2B Request-S7 modules are displayed. Select here the modules you want to convert to a *ibaCom-L2B* compatible module (usually all modules):



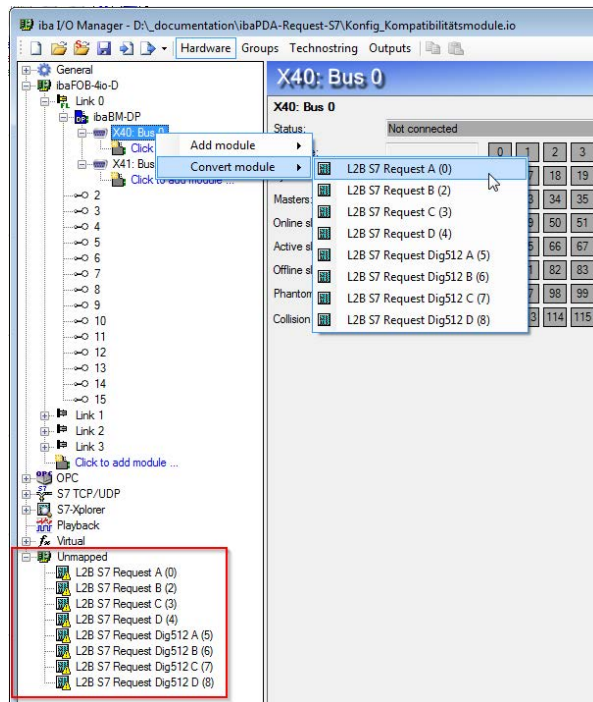
6. The selected modules are deleted on the *ibaCom-L2B* card and converted into compatible modules on the *ibaBM-DP* device.



7. The conversion is finished. Apply the new configuration with <OK>.

Note

L2B Request-S7 modules can also be converted if they are filed under the "un-mapped" modules. This is e.g. the case if the existing I/O configuration incl. L2B Request S7 modules is loaded on a new *ibaPDA* computer, that does not have an *ibaCom-L2B* anymore, but only the combination of *ibaFOB-D* and *ibaBM-DP*.



6.7 Error codes of Request blocks

The Request blocks deliver the following possible error codes.

FB140/141/...

Value ERROR_STATUS	Description
1	datablock ibaREQ_DB is write protected
2	datablock ibaREQ_DB invalid (DB =0 or > limit of cpu)
3	datablock ibaREQ_DB does not exist
4	datablock ibaREQ_DB undefined error
5	datablock ibaREQ_DB too short
6	datablock ibaREQ_DB too short for ibaREQ_UDP
9	internal error (RD_SINFO)
10	no access to datablock ibaREQ_DB (read)
11	no access to datablock ibaREQ_DB (write)
20	initialization not finished

Value ERROR_STATUS	Description
21	insufficient memory for SZL
22	wrong SZL_ID
23	wrong or invalid index of SZL
24	error while reading I&M data from cpu
25	error while reading plc data
31	initialization canceled with error
32	initialization not completed
41	too many pointers (ibaREQ_DB too small)
42	too many pointers in one command (>128)
44	invalid command id
45	operand invalid (not defined)
46	operand invalid (datatype)
47	operand invalid (memory area)
200	no connection to PN device / DP slave
300	version of ibaREQ_UDPack does not match with ibaREQ_M (ID)
301	version of ibaREQ_UDPack does not match with ibaREQ_M (FB)
302	version of ibaREQ_UDPack does not match with ibaREQ_M (DB)
303	type of transmit agent does not match with configured request type in ibaPDA
305	PROFIBUS DP slave hardware configuration is invalid
306	configured peripheral address is invalid
310	no access to datablock ibaREQ_DB (read)
311	no access to datablock ibaREQ_DB (write)
315	error while masking of synchronous faults
316	error while demasking of synchronous faults
320	operand invalid (datatype)
321	operand invalid (pointer)
401	ADR_SLOT / ADR_SLOT_0 invalid hw-id
402	ADR_SLOT / ADR_SLOT_0 invalid hw-id, no IO-Device or DP-Slave
403	ADR_SLOT / ADR_SLOT_0 invalid hw-id, is no PROFIBUS or PROFINET
406	ADR_SLOT / ADR_SLOT_0 invalid configuration slot (0)
407	ADR_SLOT / ADR_SLOT_0 invalid configuration slot (0)
409	ADR_SLOT_1 invalid configuration slot 1
410	no connection to PN device / DP slave or error
411	ADR_SLOT_1 invalid hw-id
412	ADR_SLOT_1 invalid hw-id, no IO-Device or DP-Slave
413	ADR_SLOT_1 invalid hw-id, is no PROFIBUS
416	ADR_SLOT_1 invalid configuration slot 1
0x8yyy	errorcode of inner TUSEND / AG_SEND / AG_LSEND

Table 14: Error codes of Request blocks FB140/141/...

FC122 (PROFIBUS)

Value ERROR_STATUS	Description
1	DB_PDA is write protected
2	DB_PDA = 0 or > limit of cpu
3	DP_PDA does not exist
5	DB_PDA too short
11	DB_INTERN is write protected
12	DB_INTERN = 0 or > limit of CPU
13	DB_INTERN does not exist
15	DB_INTERN too short
16	error while reading identification data of CPU
19	initialization not completed
21	insufficient memory for system status list
22	wrong or unknown system status list
23	wrong or invalid index of system status list
30	invalid OUTPUT_ADR_SLAVE
31	OUTPUT_ADR_SLAVE no PROFIBUS DP slave
100	bit number not 0
101	bit number not 0-7
103	operand invalid (memory area)
104	operand invalid (datatype)
105	operand invalid (datablock 0)
106	datablock number > limit of cpu
107	datablock does not exist
109	datablock too short
110	address does not exist
111	initialization canceled with error
112	initialization not completed
150	request fragmentation not supported
151	wrong number of requested values
152	only <64 digital signals are supported
153	only <64 analog signals are supported
200	no connection to DP slave

Table 15: Error codes of Request block FC122

FC123

Value ERROR_STATUS	Description
1	DB_PDA is write protected
2	DB_PDA = 0 or > limit of cpu
3	DP_PDA does not exist
5	DB_PDA too short
11	DB_INTERN is write protected
12	DB_INTERN = 0 or > limit of CPU
13	DB_INTERN does not exist
15	DB_INTERN too short
16	error while reading identification data of CPU
19	initialization not completed
21	insufficient memory for system status list
22	wrong or unknown system status list
23	wrong or invalid index of system status list
30	invalid OUTPUT_ADR_SLAVE
31	OUTPUT_ADR_SLAVE no PROFIBUS DP slave
32	RM: the parameterized "OUTPUT_ADR_SLAVE_BUS_0" ist wrong.
33	RM: the parameterized "OUTPUT_ADR_SLAVE_BUS_0" is not assigned to a PROFIBUS DP slave
34	RM: the parameterized "OUTPUT_ADR_SLAVE_BUS_1" ist wrong
35	RM: the parameterized "OUTPUT_ADR_SLAVE_BUS_1" is not assigned to a PROFIBUS DP slave
36	RM: SLAVE BUS0 and SLAVE BUS1 do not have the same DP address
100	bit number not 0
101	bit number not 0-7
103	operand invalid (memory area)
104	operand invalid (datatype)
105	operand invalid (datablock 0)
106	datablock number > limit of cpu
107	datablock does not exist
109	datablock too short
110	address does not exist
111	initialization canceled with error
112	initialization not completed
150	request fragmentation not supported
151	wrong number of requested values
152	only <64 digital signals are supported
153	only <64 analog signals are supported

Value ERROR_STATUS	Description
200	no connection to DP slave
201	RM: slave bus 0 has failed
202	RM: slave bus 1 has failed
203	RM: slaves bus 0 + 1 have failed
210	output modules of the slaves bus 0 and 1 are configured differently

Table 16: Error codes of Request block FC123

6.8 Use of MPI/DP-TCP-adapters

Siemens S7 CPUs which do not have an Ethernet interface can be connected via MPI/DP-TCP-adapters on their MPI interface.

Adapters, which convert the S7 communication from TCP/IP to MPI/PROFIBUS DP, are offered by different manufacturers.

Basically, these adapters can be used together with *ibaPDA*. Therefore, the connection on the *baPDA*-side must be configured as a TCP-connection.

The adapter can be accessed via an IP address that is set using the software supplied by the manufacturer.

In the connection configuration, enter "0" as frame and the MPI- or DP-address of the S7 CPU as slot.

For more information please refer to the corresponding device documentation.

7 Support and contact

Support

Phone: +49 911 97282-14
Fax: +49 911 97282-33
Email: support@iba-ag.com

Note



If you need support for software products, please state the license number or the CodeMeter container number (WIBU dongle). For hardware products, please have the serial number of the device ready.

Contact

Headquarters

iba AG
Koenigswarterstrasse 44
90762 Fuerth
Germany

Phone: +49 911 97282-0
Fax: +49 911 97282-33
Email: iba@iba-ag.com

Mailing address

iba AG
Postbox 1828
D-90708 Fuerth, Germany

Delivery address

iba AG
Gebhardtstrasse 10
90762 Fuerth, Germany

Regional and Worldwide

For contact data of your regional iba office or representative please refer to our web site

www.iba-ag.com.