ibaPDA-Request-S7-DP/PN/ibaNet-E

Applications of the Request data interface for SIMATIC S7-300/S7-400

Manual part 2

Issue 4.0

Measurement Systems for Industry and Energy

www.iba-ag.com

Manufacturer

iba AG
Koenigswarterstrasse 44
90762 Fuerth
Germany

Contacts

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

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	Author	Version SW
4.0	04-2024	GUI new, ibaNet-E added, S7 routing via TIA Portal	st, mm	8.7.0

Contents

1 About this documentation

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

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

Other documentation

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

1.1 Target group and previous knowledge

This documentation is aimed at qualified professionals who are familiar with handling electrical and electronic modules as well as communication and measurement technology. A person is regarded as professional if he/she is capable of assessing safety and recognizing possible consequences and 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 *ibaPDA-Request-S7-DP/PN/ibaNet-E* 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 *Logic diagram*
Calling the menu command	*Step 1 – Step 2 – Step 3 – Step x* Example: Select the menu *Logic diagram – Add – New function block*.
Keys	<Key name> Example: <Alt>; <F1>
Press the keys simultaneously	<Key name> + <Key name> Example: <Alt> + <Ctrl>
Buttons	<Key name> Example: <OK>; <Cancel>
Filenames, paths	`Filename`, `Path` Example: `Test.docx`

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.

1.4 Documentation structure

The documentation for the *ibaPDA-Request-S7-DP/PN/ibaNet-E* interface (PDF edition) is divided into two separate parts. Each part has its own chapter and page numbering beginning at 1 and is independently updated.

Part	Title	Content
Part 1	Applications of the Request data interface for SIMATIC S7 TIA Portal/S7-1500	Features, settings, and Request blocks for the Request interface in conjunction with SIMATIC TIA Portal and S7-1500 S7 controller
Part 2	Applications of the Request data interface for SIMATIC S7-300/S7-400	Features, settings, and Request blocks for the Request interface in conjunction with SIMATIC STEP 7 and S7-300 and S7-400 S7 controllers

2 Request S7 variants

This chapter describes the Request S7 variants for applications with S7-300/400 controllers. For applications with S7-1500 controllers, refer to the manual Part 1.

S7 family	Firmware	Addressing type	iba block family	iba data paths
S7-300	Unlimited	Operands	ibaREQ	DP, PN, UDP
S7-400	Unlimited	Operands	ibaREQ	DP, PN, UDP

2.1 Request-S7 for ibaBM-PN

This section describes the Request-S7 variant for the PROFINET bus module *ibaBM-PN*.

2.1.1 General information

You can configure Request-S7 for *ibaBM-PN* 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

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

Depending on the hardware and software, you can select different access points for the Request:

- TCP/IP: The connection to the SIMATIC S7 is established using an integrated PN interface of the S7 CPU or the respective CP modules in the PLC and the standard network interface of the computer. 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, for the PC/CP group the SIMATIC communication software (and corresponding licenses) must be installed on the computer with the corresponding functions enabled.
  - MPI, PROFIBUS: The connection to the SIMATIC S7 is established using the MPI or PROFIBUS interface of the computer; e.g. with the CP5611 PCI card or the MPI adapter for USB interface or serial PC interface.
  - TCP/IP, ISO: Here, either the standard network interface of the computer or a suitable interface card is used for the connection to the S7.

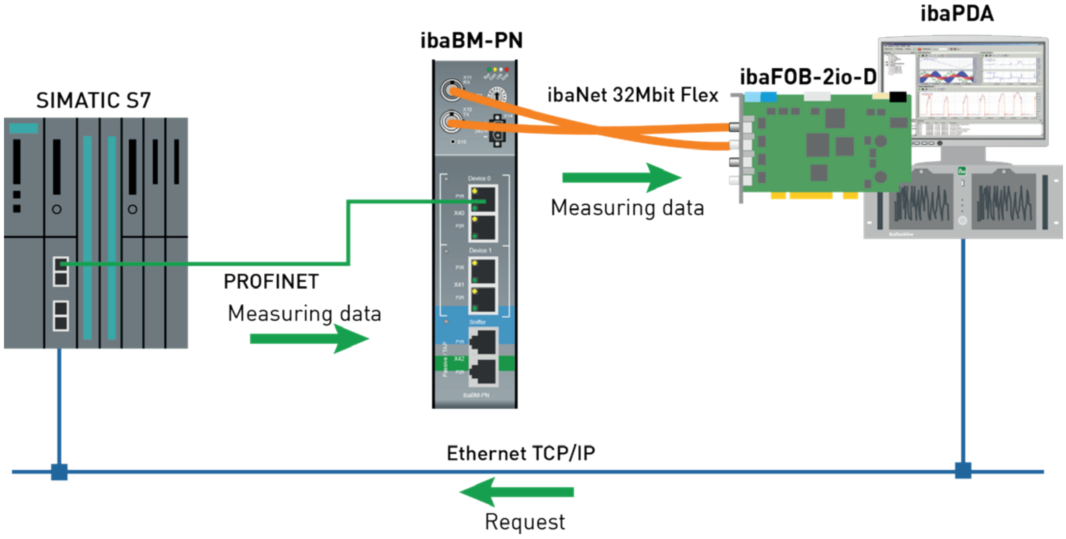
System integration with ibaBM-PN

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.



Other documentation

Detailed information about *ibaBM-PN* can be found in the device manual.

For information about application examples, see chapter ➚ *Application examples*, page 94.

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

This section describes the configuration and engineering on the SIMATIC S7 side with SIMATIC Manager (STEP 7 Version ≤ V5).

On the SIMATIC side, carry out the following configuration steps:

- 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

2.1.2.1 Hardware configuration

Configure a separate PROFINET device for each PROFINET device in use.

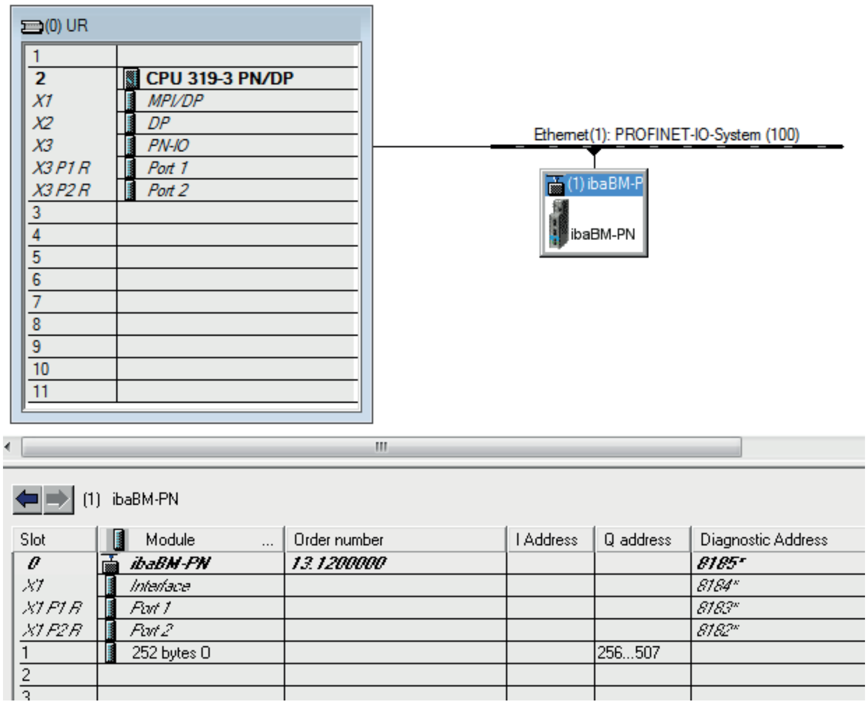
Use the GSDML file `GSDML-Vx.yy-ibaBM-PN-yyyymmdd.xml`.

Note

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

`\02_iba_Hardware\ibaBM-PN\01_GSD_Files\`

S7 CPUs 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.



(0) UR

1	
2	CPU 319-3 PN/DP
X1	MPI/DP
X2	DP
X3	PN-IO
X3 P1 R	Port 1
X3 P2 R	Port 2
3	
4	
5	
6	
7	
8	
9	
10	
11	

(1) ibaBM-PN

Slot	Module ...	Order number	I Address	Q address	Diagnostic Address
0	ibaBM-PN	13.1200000			8185*
X1	Interface				8184*
X1 P1 R	Port 1				8183*
X1 P2 R	Port 2				8182*
1	252 bytes O			256...507	
2					
3					

2.1.2.2 Configuration in STEP 7

This section describes 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

Only use Request blocks from the latest iba S7 library!

Request blocks in application examples can be outdated and, thus, cause errors.

- Copy the following blocks from the iba S7 library to the blocks folder of your STEP 7 project, see ↗ *iba S7 library*, page 91. One set of Request blocks has to be called for each Request module (connection) in *ibaPDA*.
  - ibaREQ_M (FB140), see ↗ *ibaREQ_M (FB140)*, page 69
  - ibaREQ_PN (FB141), see ↗ *ibaREQ_PN (FB141)*, page 70
  - ibaREQ_PNdev (FB150), see ↗ *ibaREQ_PNdev (FB150)*, page 72
  - ibaREQ_DB (DB15)

Note

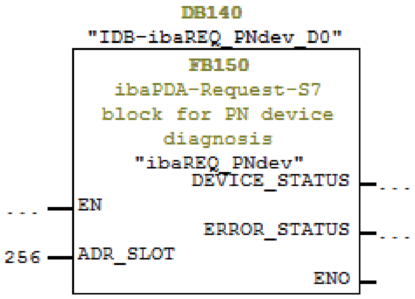
If the block numbers are already used in your project, assign other free numbers to the blocks from the iba S7 library when copying.

Note

The request blocks do not support multi-instance calls.

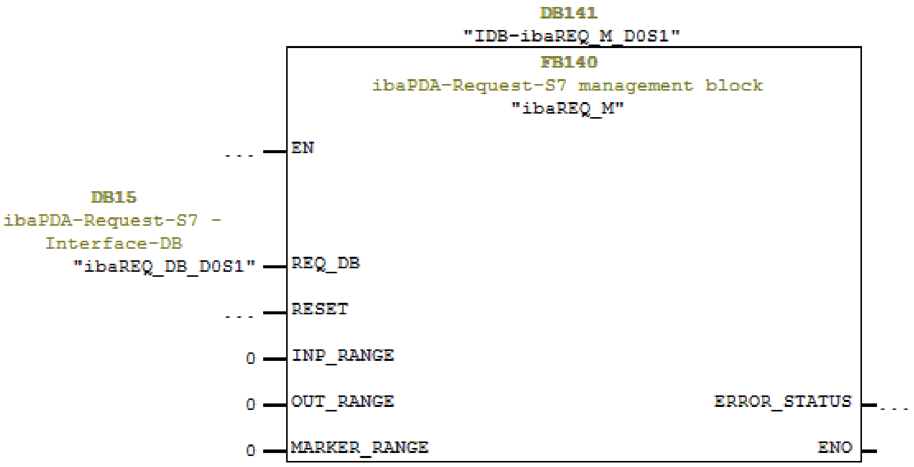
For each used PROFINET device of an ibaBM-PN

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

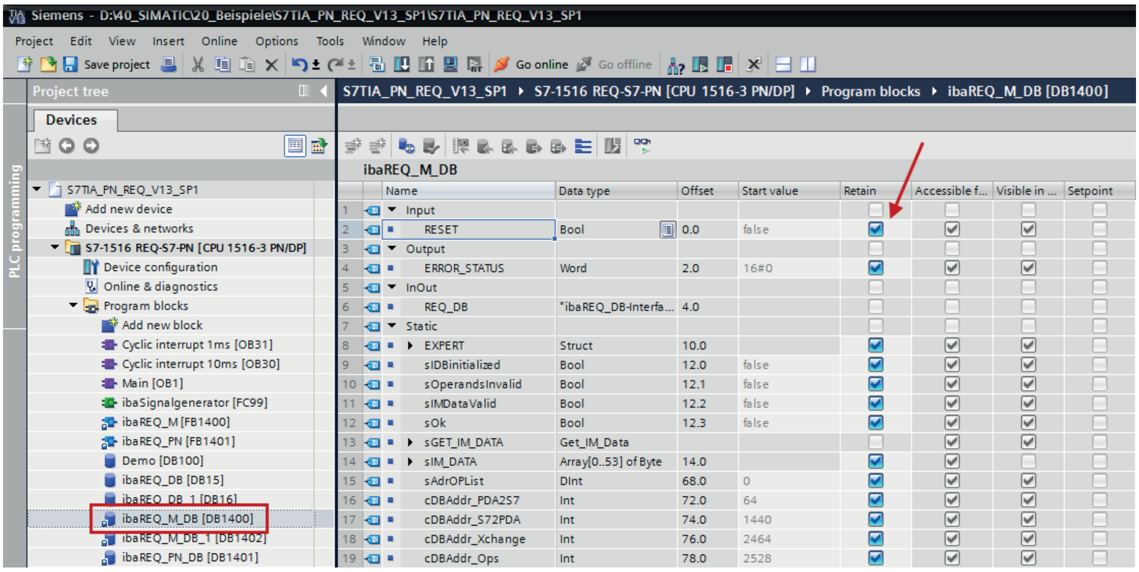


For each Request module

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



2. Enable the option *Retain* for the entire instance data block you have just created.



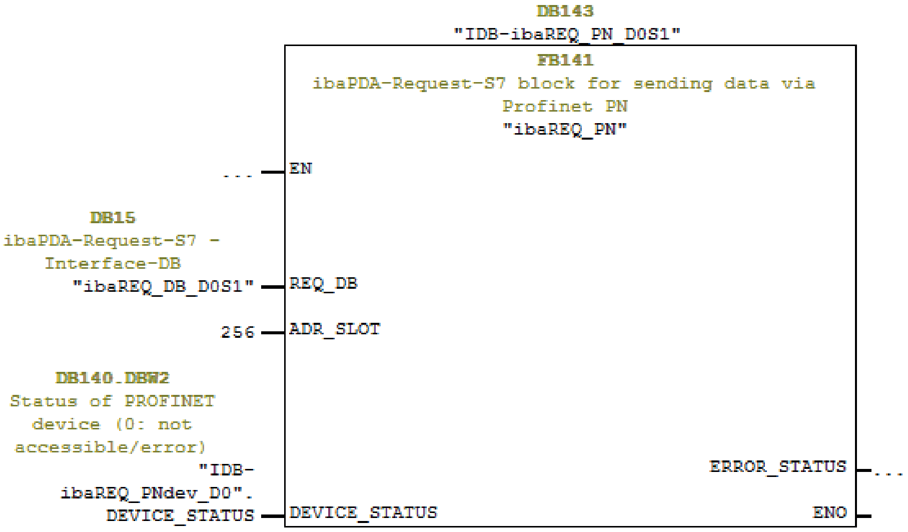
S7TIA_PN_REQ_V13_SP1 ▸ S7-1516 REQ-S7-PN [CPU 1516-3 PN/DP] ▸ Program blocks ▸ ibaREQ_M_DB [DB1400]

- S7TIA_PN_REQ_V13_SP1
  - Add new device
  - Devices & networks
  - S7-1516 REQ-S7-PN [CPU 1516-3 PN/DP]
    - Device configuration
    - Online & diagnostics
    - Program blocks
      - Add new block
      - Cyclic interrupt 1ms [OB31]
      - Cyclic interrupt 10ms [OB30]
      - Main [OB1]
      - ibaSignalgenerator [FC99]
      - ibaREQ_M [FB1400]
      - ibaREQ_PN [FB1401]
      - Demo [DB100]
      - ibaREQ_DB [DB15]
      - ibaREQ_DB_1 [DB16]
      - ibaREQ_M_DB [DB1400]
      - ibaREQ_M_DB_1 [DB1402]
      - ibaREQ_PN_DB [DB1401]

ibaREQ_M_DB

	Name	Data type	Offset	Start value	Retain	Accessible f...	Visible in ...	Setpoint
1	▼ Input				□	□	□	□
2	RESET	Bool	0.0	false	☑	☑	☑	□
3	▼ Output				□	□	□	□
4	ERROR_STATUS	Word	2.0	16#0	☑	☑	☑	□
5	▼ InOut				□	□	□	□
6	REQ_DB	"ibaREQ_DB-Interfa...	4.0		□	□	□	□
7	▼ Static				□	□	□	□
8	▸ EXPERT	Struct	10.0		☑	☑	☑	□
9	sIDBinitialized	Bool	12.0	false	☑	☑	☑	□
10	sOperandsInvalid	Bool	12.1	false	☑	☑	☑	□
11	sIMDataValid	Bool	12.2	false	☑	☑	☑	□
12	sOk	Bool	12.3	false	☑	☑	☑	□
13	▸ sGET_IM_DATA	Get_IM_Data			□	☑	☑	□
14	▸ sIM_DATA	Array[0..53] of Byte	14.0		☑	☑	□	□
15	sAdrOPList	DInt	68.0	0	☑	☑	☑	□
16	cDBAddr_PDA2S7	Int	72.0	64	☑	☑	☑	□
17	cDBAddr_S72PDA	Int	74.0	1440	☑	☑	☑	□
18	cDBAddr_Xchange	Int	76.0	2464	☑	☑	☑	□
19	cDBAddr_Ops	Int	78.0	2528	☑	☑	☑	□

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) data block 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 with the new DB number (input REQ_DB) for each Request module.
- Within the OB1 or a cyclic interrupt OB (OB3x), the ibaREQ_PN (FB141) has to be called once more with the new DB number (input REQ_DB) for each Request module.
- Make sure that all instance data blocks are unique and that unique values are assigned for the ADR_SLOT.
- It is sufficient to call the ibaREQ_PNdev just once per used PROFINET device and not for each Request module.

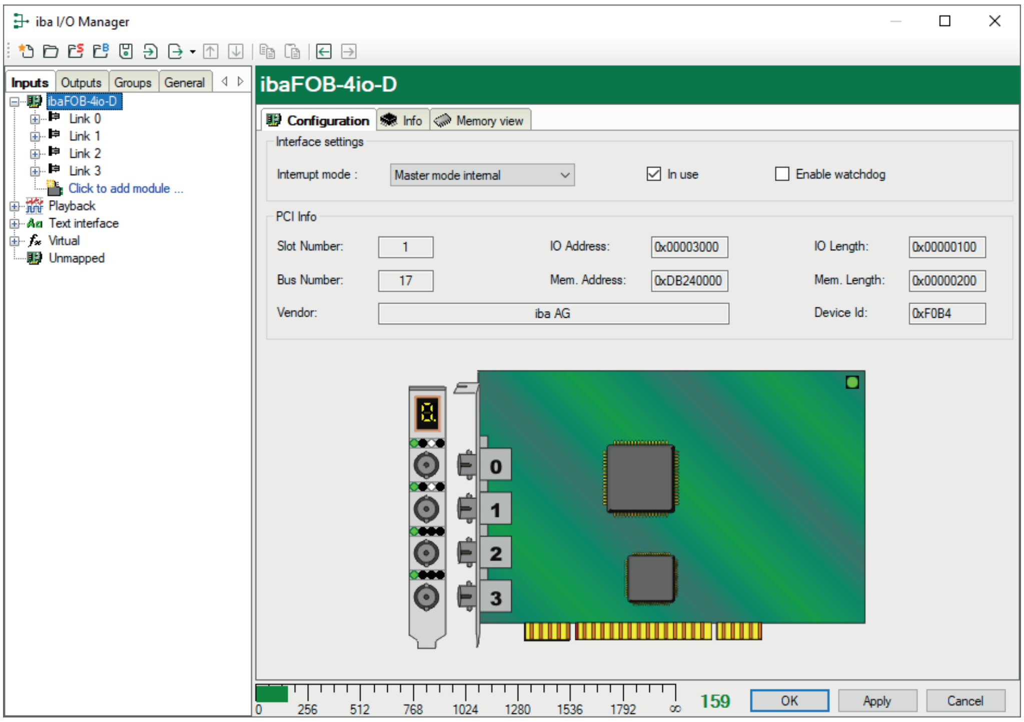
Final

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

2.1.3 Configuration and engineering ibaPDA

2.1.3.1 General interface settings

If an *ibaFOB-D* card is installed in the *ibaPDA* computer, *ibaPDA* displays the interface of the *ibaFOB-D* card in the interface tree of the I/O Manager.



Interface settings

Interrupt mode : Master mode internal ☑ In use □ Enable watchdog

PCI Info

Slot Number: 1 IO Address: 0x00003000 IO Length: 0x00000100

Bus Number: 17 Mem. Address: 0xDB240000 Mem. Length: 0x00000200

Vendor: iba AG Device Id: 0xF0B4

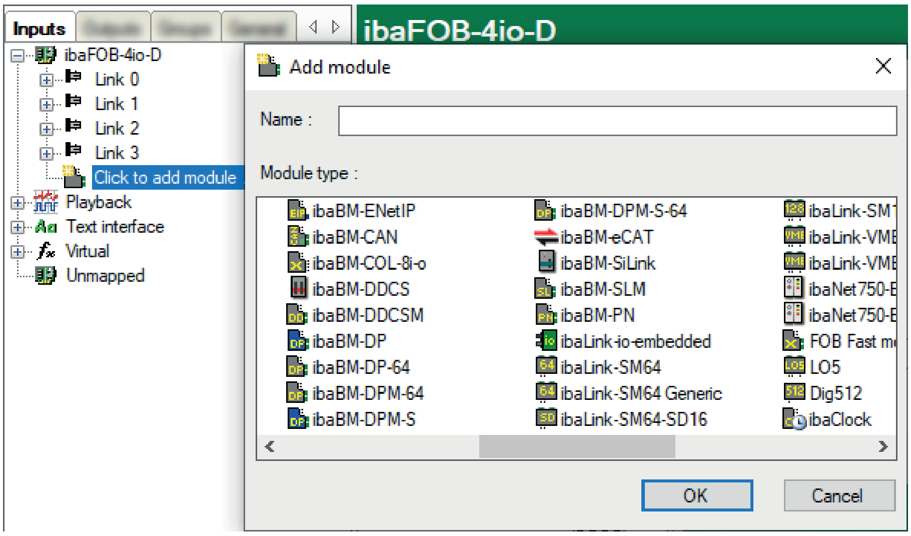
Other documentation

Detailed information about the *ibaFOB-D* card can be found in the respective manual.

Configuring ibaBM-PN

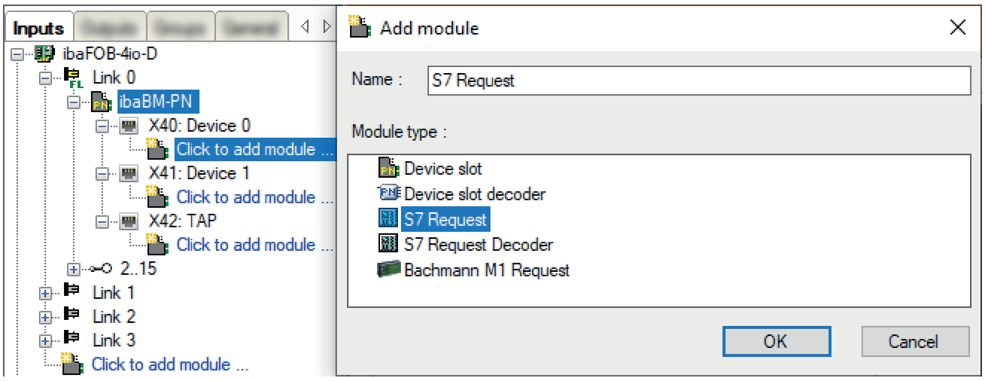
1. Start the *ibaPDA* client and open the I/O Manager.
2. In the interface tree (left), select the link of the *ibaFOB* card to which *ibaBM-PN* is connected.

   Click on the blue command *Click to add module*. Select the *ibaBM-PN* module in the dialog box and assign a name via the input field if required. Confirm the selection with <OK>.



3. Add a Request module (or several modules, in case you need more connections to one or to different S7-CPUs) 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)



Name : S7 Request

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*.

2.1.3.2 General module settings

You can find the description of all settings that are identical for all Request-S7 modules in the general chapter *General module settings* in part 1 of the manual.

You can find information on connection settings from the *Connection* tab in the general chapter *General module settings* in part 1 of the manual.

The *ibaBM-PN* modules have the following specific setting options in the *General* tab:

PROFINET

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

Slot
Number of the slot to which the module is assigned

S2 Redundancy
FALSE: Redundancy mode disabled. For further information on the redundancy mode, see ↗ *Request-S7 for ibaBM-PN in redundancy mode*, page 20.

2.1.3.3 Module S7 Request

With the *S7 Request* module, you can acquire data according to the size of the PROFINET slots used. A maximum of 254 bytes is possible.

Configure a separate PROFINET slot and Request block call for each module.

For more information on the module settings, see chapter ↗ *General module settings*, page 18 and the general chapter *General module settings* in part 1 of this manual.

2.1.3.4 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.

Configure a separate PROFINET slot and Request block call for each module.

General tab

For more information on the module settings, see chapter ↗ *General module settings*, page 18 and the general chapter *General module settings* in part 1 of the manual.

Module specific settings

Module layout – No. of decoders
Define the number of configurable decoders in the digital signal table. The default value is 32. The maximum value is 126. The signal tables are adjusted accordingly.

Connection configuration

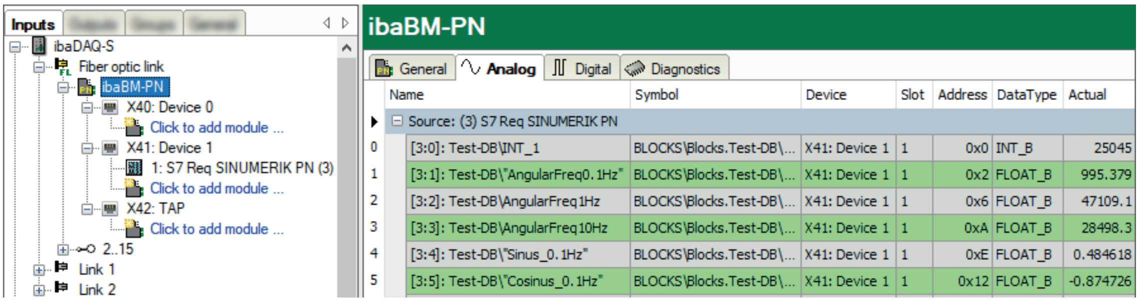
Configure the connection of the *S7 Request Decoder* module in the same way as the connection for an S7-Request module, see the general chapter *Connection settings* in part 1 of the manual.

Digital tab

Configure the signals in the same way as for an *ibaBM-DP* device, see chapter *Module S7 Request Decoder* for *ibaBM-DP* in part 1 of the manual.

2.1.4 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.



	Name	Symbol	Device	Slot	Address	DataType	Actual
	Source: (3) S7 Req SINUMERIK PN						
0	[3:0]: Test-DB\INT_1	BLOCKS\Blocks.Test-DB\...	X41: Device 1	1	0x0	INT_B	25045
1	[3:1]: Test-DB\"AngularFreq0.1Hz"	BLOCKS\Blocks.Test-DB\...	X41: Device 1	1	0x2	FLOAT_B	995.379
2	[3:2]: Test-DB\AngularFreq1Hz	BLOCKS\Blocks.Test-DB\...	X41: Device 1	1	0x6	FLOAT_B	47109.1
3	[3:3]: Test-DB\AngularFreq10Hz	BLOCKS\Blocks.Test-DB\...	X41: Device 1	1	0xA	FLOAT_B	28498.3
4	[3:4]: Test-DB\"Sinus_0.1Hz"	BLOCKS\Blocks.Test-DB\...	X41: Device 1	1	0xE	FLOAT_B	0.484618
5	[3:5]: Test-DB\"Cosinus_0.1Hz"	BLOCKS\Blocks.Test-DB\...	X41: Device 1	1	0x12	FLOAT_B	-0.874726

Other documentation

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

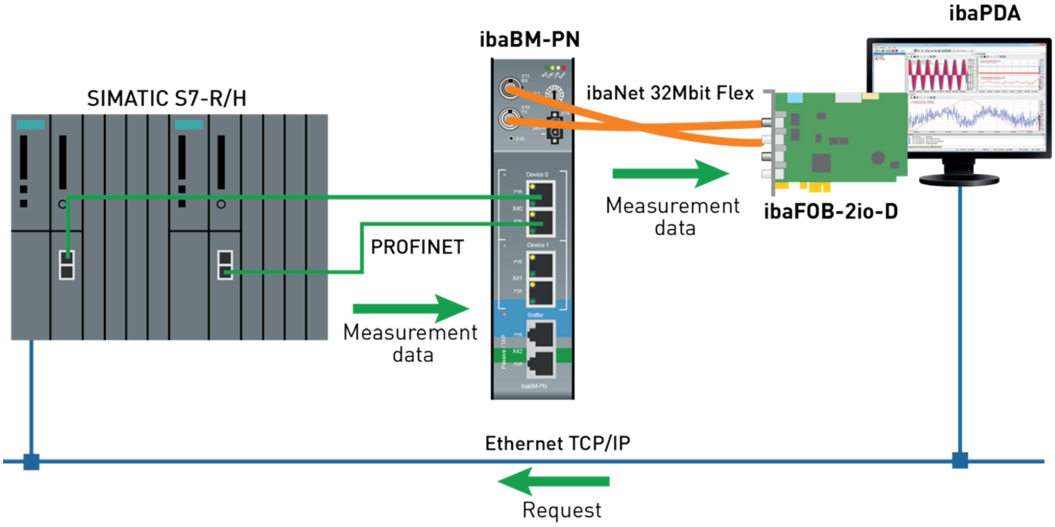
2.2 Request-S7 for ibaBM-PN in redundancy mode

This section describes the Request-S7 variant for the PROFINET bus module *ibaBM-PN* in S2 redundancy mode.

2.2.1 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*. Contact the iba Support.

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



Other documentation

Detailed information about the S2 redundancy mode of the *ibaBM-PN* can be found 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 11). The differences and extensions are described in the following.

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

2.2.2.1 Hardware configuration

Configure a separate PROFINET device for each PROFINET device in use.

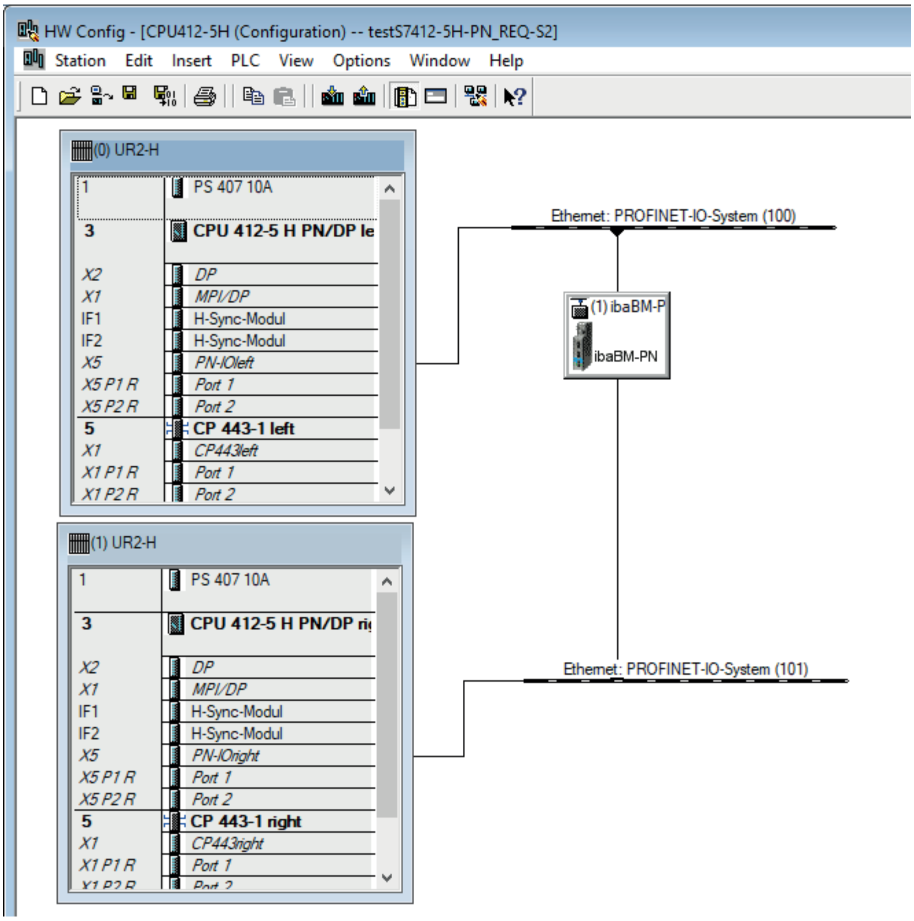
Use the same GSDML file as for normal operation (from version V2.35-20200101).

Note

You find the latest version of the GSDML file on the data storage medium "iba Software & Manuals" 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.


HW Config - [CPU412-5H (Configuration) -- testS7412-5H-PN_REQ-S2]
Station
Edit
Insert
PLC
View
Options
Window
Help
Ethernet: PROFINET-IO-System (100)
(1) ibaBM-P
ibaBM-PN
Ethernet: PROFINET-IO-System (101)

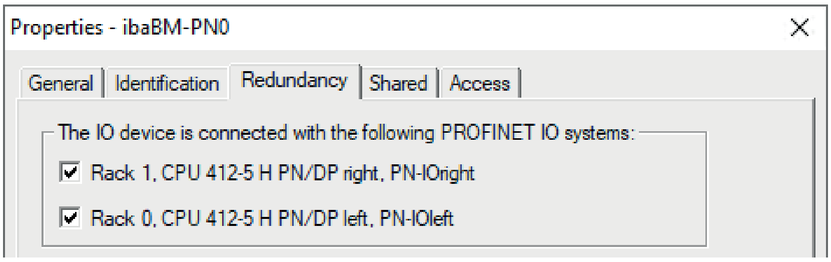
(0) UR2-H

1	PS 407 10A
3	CPU 412-5 H PN/DP le
X2	DP
X1	MPI/DP
IF1	H-Sync-Modul
IF2	H-Sync-Modul
X5	PN-IOleft
X5 P1 R	Port 1
X5 P2 R	Port 2
5	CP 443-1 left
X1	CP443left
X1 P1 R	Port 1
X1 P2 R	Port 2

(1) UR2-H

1	PS 407 10A
3	CPU 412-5 H PN/DP ri
X2	DP
X1	MPI/DP
IF1	H-Sync-Modul
IF2	H-Sync-Modul
X5	PN-IOright
X5 P1 R	Port 1
X5 P2 R	Port 2
5	CP 443-1 right
X1	CP443right
X1 P1 R	Port 1
X1 P2 R	Port 2

You can configure the connection in the *Redundancy* tab of Properties.



The IO device is connected with the following PROFINET IO systems:

☑ Rack 1, CPU 412-5 H PN/DP right, PN-IOright

☑ Rack 0, CPU 412-5 H PN/DP left, PN-IOleft

Apply this hardware configuration to operate *ibaBM-PN* in the S2 redundancy mode.

2.2.2.2 Configuration in STEP 7

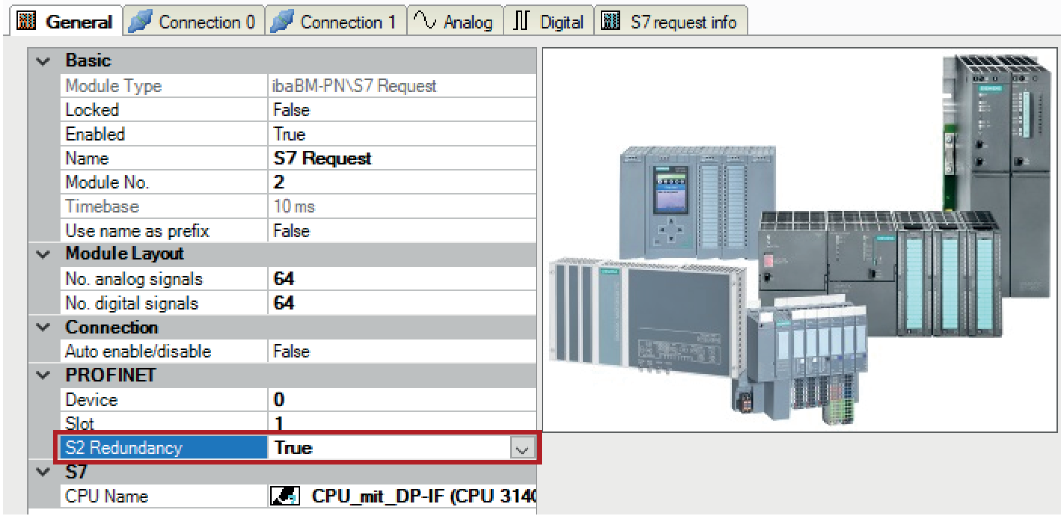
The configuration of the request blocks corresponds to the configuration in normal operation (see chapter ↗ *Configuration in STEP 7*, page 13).

2.2.3 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 orange colored icon of the Request module indicates the redundancy mode.



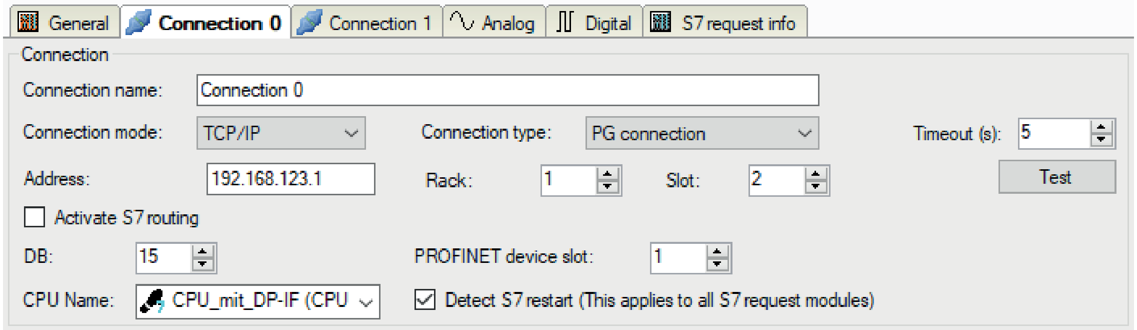
Basic	
Module Type	ibaBM-PN\S7 Request
Locked	False
Enabled	True
Name	**S7 Request**
Module No.	**2**
Timebase	10 ms
Use name as prefix	False
Module Layout	
No. analog signals	**64**
No. digital signals	**64**
Connection	
Auto enable/disable	False
PROFINET	
Device	**0**
Slot	**1**
S2 Redundancy	**True**
S7	
CPU Name	**CPU_mit_DP-IF (CPU 314**

The configuration of the modules otherwise corresponds to the configuration in standard mode, see chapter *General module settings* in part 1.

Connection settings

For transmitting the operand data to both CPUs of the SIMATIC S7-R/H, set up two different connections. *ibaPDA* switches between these connections depending on availability and requirements.

Configure both connections on the tabs *Connection 0* and *Connection 1*. You can change the name of the tabs in the *Connection name* fields.



Connection

Connection name: Connection 0

Connection mode: TCP/IP

Connection type: PG connection

Timeout (s): 5

Address: 192.168.123.1

Rack: 1

Slot: 2

□ Activate S7 routing

DB: 15

PROFINET device slot: 1

CPU Name: CPU_mit_DP-IF (CPU

☑ Detect S7 restart (This applies to all S7 request modules)

The other settings are identical to the settings for operation without redundancy mode, see chapter *Connection settings* in part 1.

Note these special features:

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

2.3 Request-S7 for ibaBM-DP

This section describes the Request-S7 variant for the PROFIBUS bus module *ibaBM-DP*.

2.3.1 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	-
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 using an integrated PN interface of the S7 CPU or the respective CP modules in the PLC and the standard network interface of the computer. 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, for the PC/CP group the SIMATIC communication software (and corresponding licenses) must be installed on the computer with the corresponding functions enabled.
  - MPI, PROFIBUS: The connection to the SIMATIC S7 is established using the MPI or PROFIBUS interface of the computer; e.g. with the CP5611 PCI card or the MPI adapter for USB interface or serial PC interface.
  - TCP/IP, ISO: Here, either the standard network interface of the computer 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 using an integrated PN interface of the S7 CPU or respective CP modules in the PLC and the standard network interface of the computer. No additional Siemens software is required for the connection.

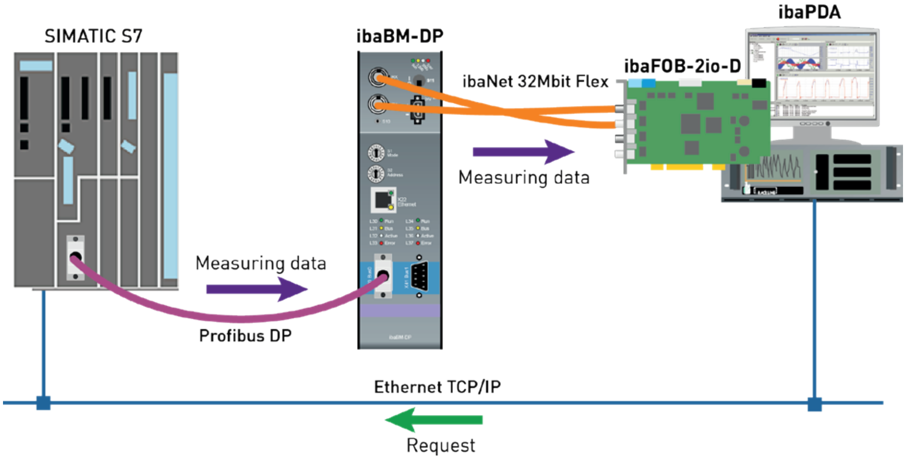
System integration with ibaBM-DP

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 ➚ *Request-S7 for ibaBM-DPM-S*, page 44.

In the standard version, you can configure a maximum of 8 connections per device, i.e. 8 PROFIBUS slaves. Up to 244 Byte data can be transmitted per slave.



Note

The following description refers to the Request blocks V4.0 or higher. If you require information regarding older versions, contact the iba Support.

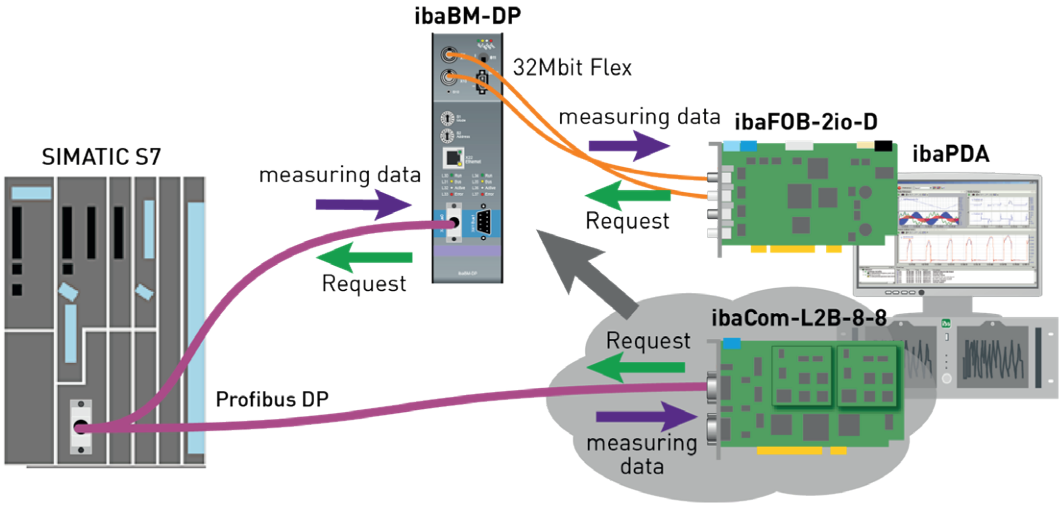
Other documentation

Detailed information about *ibaBM-DP* can be found in the device manual.

For information about application examples, see chapter ➚ *Application examples*, page 94.

2.3.1.1 ibaCom-L2B compatibility mode

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



For this purpose, you can use the following modules:

- 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, see ↗ *Request-S7 for ibaCom-L2B*, page 53 and ↗ *Replacing Request-S7 on ibaCom-L2B by ibaBM-DP*, page 108.

Note

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

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

This section describes 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, see ↗ *Configuration and engineering SIMATIC S7-300, S7-400 and WinAC*, page 53.

2.3.2.1 Hardware configuration

Specify an iba-PROFIBUS slave for each module.

Use the GSD file `ibaDPMSi.gsd` (version V2.2 and higher).

Note

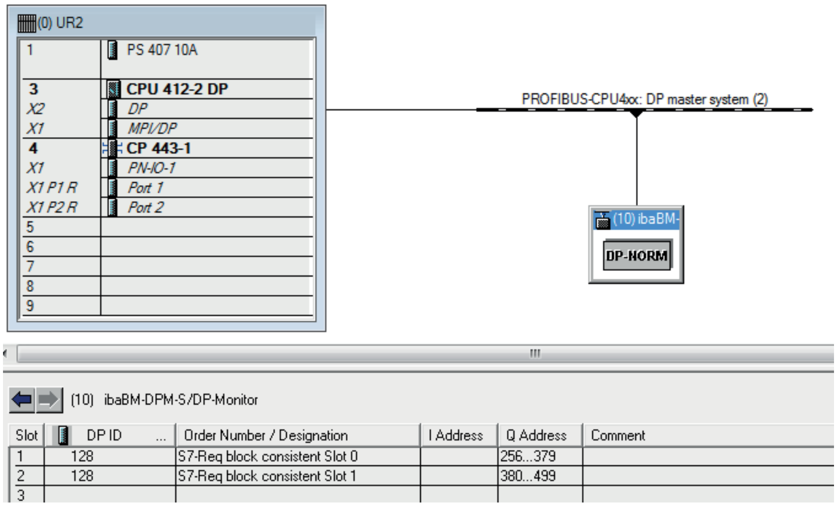
You find the GSD file `ibaDPMSi.gsd` on the data storage medium "iba Software & Manuals" in the following directory:

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

Note

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

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



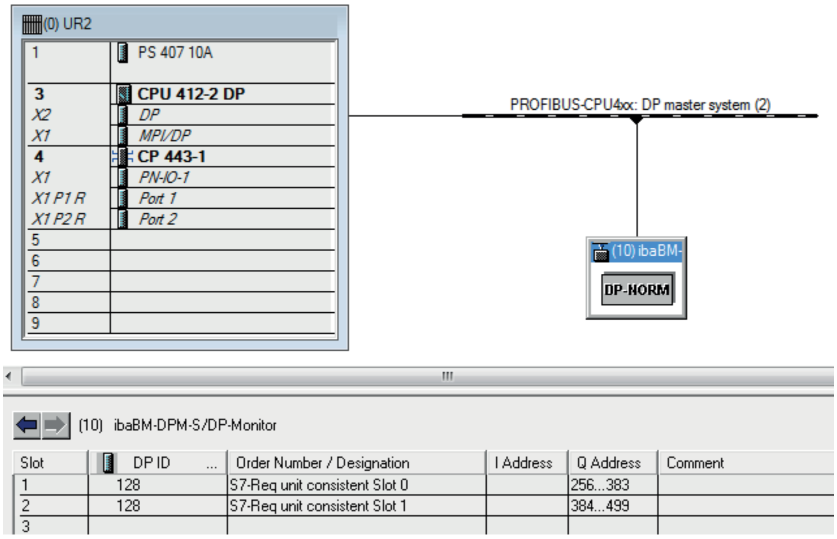
1	PS 407 10A
3	CPU 412-2 DP
X2	DP
X1	MPI/DP
4	CP 443-1
X1	PN-IO-1
X1 P1 R	Port 1
X1 P2 R	Port 2
5	
6	
7	
8	
9	

(10) ibaBM-DPM-S/DP-Monitor

Slot	DP ID	Order Number / Designation	I Address	Q Address	Comment
1	128	S7-Req block consistent Slot 0		256...379	
2	128	S7-Req block consistent Slot 1		380...499	
3					

[1)] 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

For **S7-CPUs and for CP443-5 of older types**, you cannot use long consistent slots.
Use the elements "S7-Req **unit consistent** Slot 0 / Slot 1".



(0) UR2

1	PS 407 10A
3	CPU 412-2 DP
X2	DP
X1	MPI/DP
4	CP 443-1
X1	PN-IO-1
X1 P1 R	Port 1
X1 P2 R	Port 2
5	
6	
7	
8	
9	

(10) ibaBM-DPM-S/DP-Monitor

Slot	DP ID	...	Order Number / Designation	I Address	Q Address	Comment
1	128		S7-Req unit consistent Slot 0		256...383	
2	128		S7-Req unit consistent Slot 1		384...499	
3						

2.3.2.2 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

1. Copy the following blocks from the iba S7 library to the blocks folder of your STEP 7 project, see ↗ *iba S7 library*, page 91. The ibaDP_Req has to be called once for each slave in the cyclic program.
   - FC122 (ibaDP_Req), see ↗ *Request-FC ibaDP_Req (FC122)*, page 73
   - DB10 (ibaDP_DB_PDA)
   - DB25 (ibaDP_DB_work)

Note

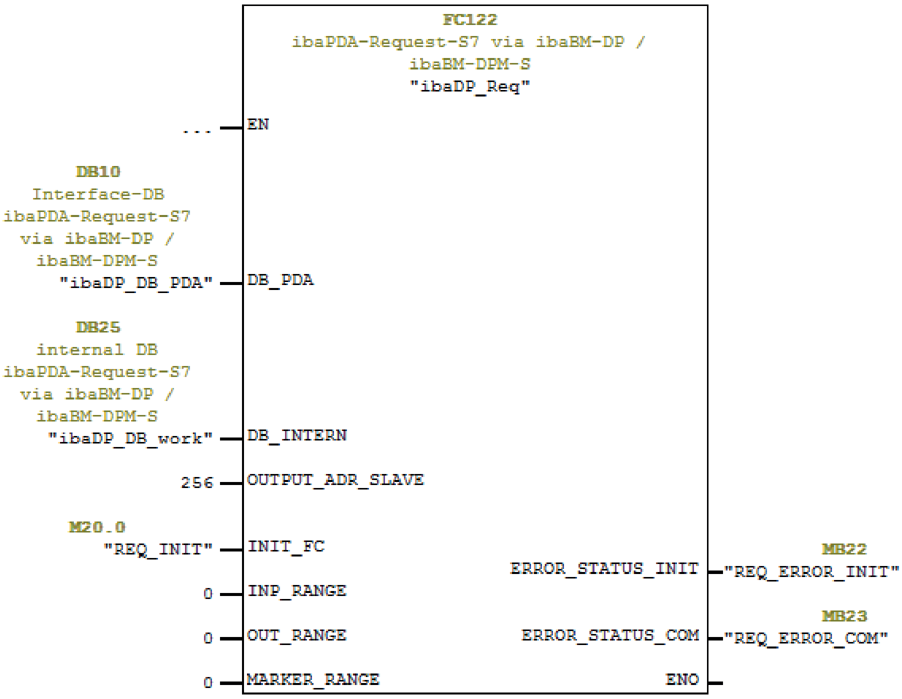
Only use Request blocks from the latest iba S7 library!

Request blocks in application examples can be outdated and, thus, cause errors.

Note

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

2. Create the error OBs (OB82, OB85, OB86, OB87, OB122) in order to prevent CPU stops in case of an error.
3. 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) has to be called once more 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.

Final

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

2.3.2.3 Configuration in STEP 7 (CFC)

For Request-S7 with one PROFIBUS slave

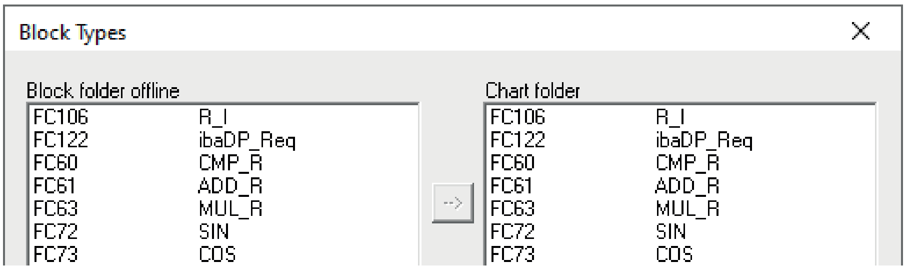
1. Copy the following blocks from the iba S7 library to the blocks folder of your STEP 7 project, see ↗ *iba S7 library*, page 91. The ibaDP_Req has to be called once for each slave in the cyclic program.
    - FC122 (ibaDP_Req), see ↗ *Request-FC ibaDP_Req (FC122)*, page 73
    - DB10 (ibaDP_DB_PDA)
    - DB25 (ibaDP_DB_work)

Note

If the block numbers FC122, DB10 and DB25 are already used in your project, assign other free 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*.

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



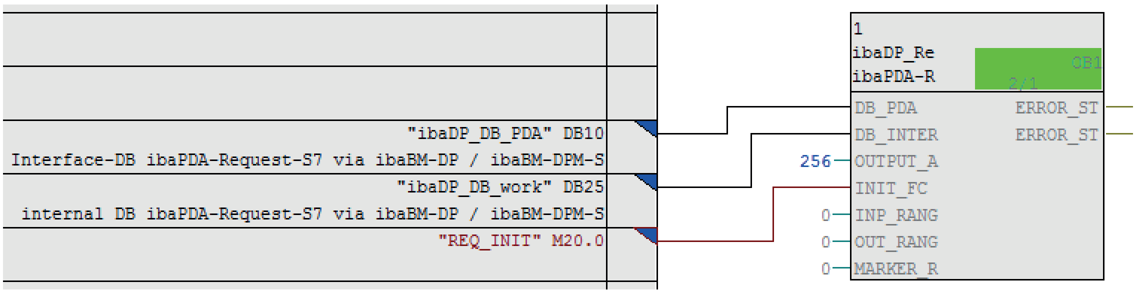
Block folder offline

FC106	R_I
FC122	ibaDP_Req
FC60	CMP_R
FC61	ADD_R
FC63	MUL_R
FC72	SIN
FC73	COS

Chart folder

FC106	R_I
FC122	ibaDP_Req
FC60	CMP_R
FC61	ADD_R
FC63	MUL_R
FC72	SIN
FC73	COS

3. 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 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 are 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.

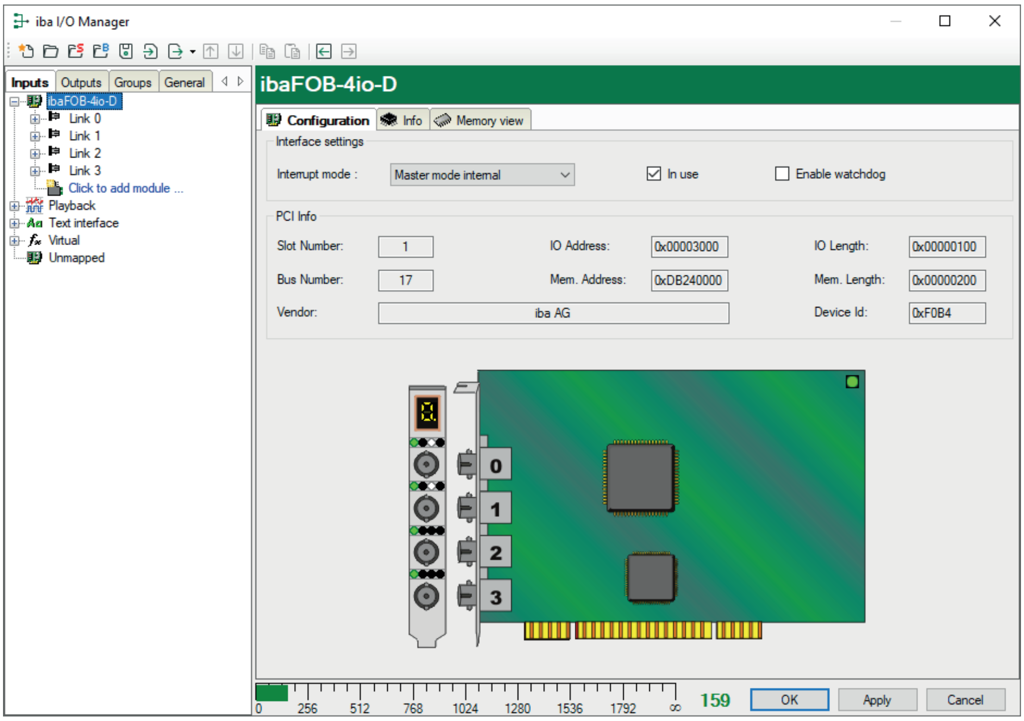
After each program modification

1. Compile the complete program:
   a.) Activate the option *Entire program*.
   b.) Activate the option *Generate SCL source* (for SIMATIC CFC version 6.1 or higher).
   c.) Deactivate the option *Delete empty runtime groups* if available.
2. Load the program and restart the S7-CPU.

2.3.3 Configuration and engineering ibaPDA

2.3.3.1 General interface settings

If an *ibaFOB-D* card is installed in the *ibaPDA* computer, *ibaPDA* displays the interface of the *ibaFOB-D* card in the interface tree of the I/O Manager.



ibaFOB-4io-D

Interface settings

Interrupt mode : Master mode internal

☑ In use

□ Enable watchdog

PCI Info

Slot Number:	1	IO Address:	0x00003000	IO Length:	0x00000100
Bus Number:	17	Mem. Address:	0xDB240000	Mem. Length:	0x00000200
Vendor:	iba AG			Device Id:	0xF0B4

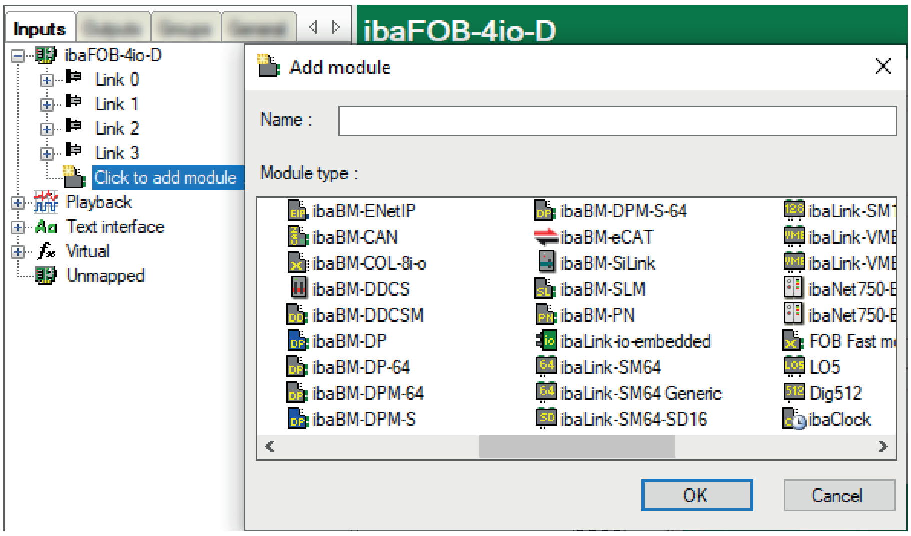
Other documentation

Detailed information about the *ibaFOB-D* card can be found in the respective manual.

Configuring ibaBM-DP

1. Start the *ibaPDA* client and open the I/O Manager.
2. In the interface tree (left), select the link of the *ibaFOB* card to which *ibaBM-DP* is connected.

   Click on the blue command *Click to add module*. Select the *ibaBM-DP* module in the dialog box and assign a name via the input field if required. Confirm the selection with <OK>.



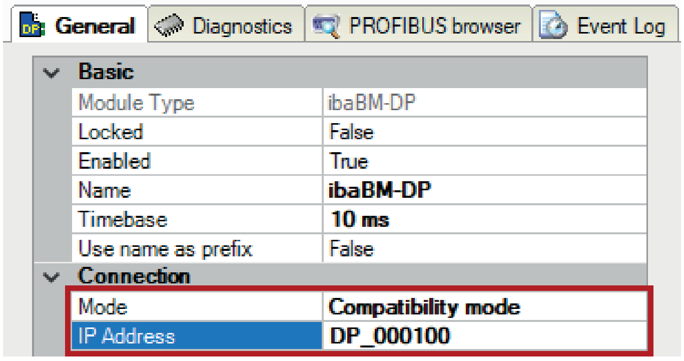
Name :

Module type :

ibaBM-ENetIP
ibaBM-CAN
ibaBM-COL-8i-o
ibaBM-DDCS
ibaBM-DDCSM
ibaBM-DP
ibaBM-DP-64
ibaBM-DPM-64
ibaBM-DPM-S
ibaBM-DPM-S-64
ibaBM-eCAT
ibaBM-SiLink
ibaBM-SLM
ibaBM-PN
ibaLink-io-embedded
ibaLink-SM64
ibaLink-SM64 Generic
ibaLink-SM64-SD16
ibaLink-SM1
ibaLink-VME
ibaLink-VME
ibaNet750-E
ibaNet750-E
FOB Fast m
LO5
Dig512
ibaClock

3. If you operate the device in Flex mode, *ibaPDA* sets the IP address of the device on the *General* tab under *Connection* automatically.

   If you operate the device in compatibility mode, enter the IP address of the device on the *General* tab under *Connection*: either as name e.g. "DP_000100" or as IP address e.g. "192.168.81.123".



Basic	
Module Type	ibaBM-DP
Locked	False
Enabled	True
Name	**ibaBM-DP**
Timebase	**10 ms**
Use name as prefix	False
Connection	
Mode	**Compatibility mode**
IP Address	**DP_000100**

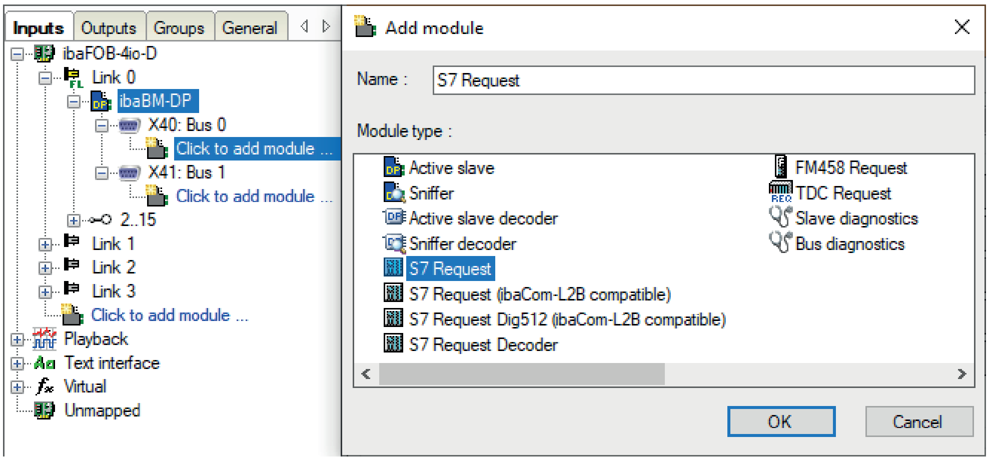
Note

An online connection to the device via Ethernet or USB is mandatory in compatibility mode. The manual of the *ibaBM-DP* device explains how to establish and check this connection.

The device manual also describes differences between Flex mode and compatibility mode in detail.

4. Add a request module (or several modules, in case you need more connections to one or to different S7-CPUs) 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)



Name : S7 Request

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*.

2.3.3.2 General module settings

You can find the description of all settings that are identical for all Request-S7 modules in the general chapter *General module settings* in part 1 of the manual.

You can find information on connection settings from the *Connection* tab in the general chapter *General module settings* in part 1 of the manual.

The *ibaBM-DP* modules have the following specific setting options in the *General* tab:

PROFIBUS

Bus number
0 = connector X40 left, 1 = connector X41 right

Slave number
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. If successful, the acquisition is restarted with this module enabled.
If FALSE, the acquisition is not started, if a connection to the configured S7-CPU cannot be established.

2.3.3.3 Module S7 Request

With the *S7 request* module, you can acquire up to 64 analog and 64 digital signals.

Configure a separate PROFIBUS slave and Request block call for each module.

For a description of the module settings, see the general chapter *General module settings* and *General module settings* for *ibaBM-DP* in part 1 of the manual.

2.3.3.4 Module 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.

Configure a separate PROFIBUS slave and Request block call for each module.

General tab

For more information on the module settings, see ➚ *General module settings*, page 33, and the general chapter *General module settings* in part 1 of the manual.

Connection configuration

Configure the connection of the *S7 Request Decoder* module in the same way as the connection for an S7-Request module, see the general chapter *Connection settings* in part 1 of the manual.

Digital tab

The declaration of the digital signals is done in two steps.

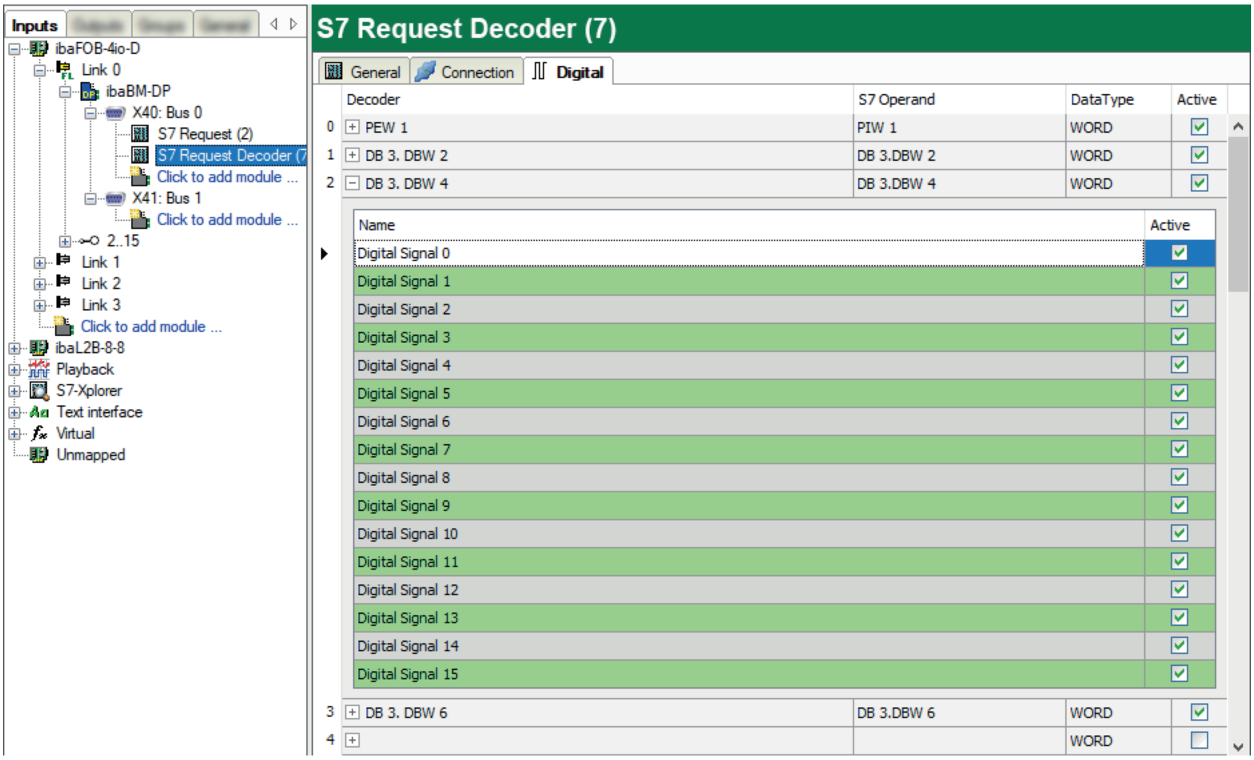
- First define the words (source signals), which are broken down for the digital signals (bits).

  You can directly enter the words as basic signals for decoding via the absolute S7 operands. Only word operands (e.g. PIW, MW, DBW) are allowed.

  You can also use S7 symbols by generating address books. For detailed information, see chapter *Selection via the symbolic operand addresses* in part 1 of the manual. 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.

- You can open each word (source signal) via the <+> button to display the list of associated digital signals.

  Then define the individual digital signals (bits) of the source signal.



	Decoder	S7 Operand	DataType	Active
0	⊞ PEW 1	PIW 1	WORD	☑
1	⊞ DB 3. DBW 2	DB 3.DBW 2	WORD	☑
2	⊟ DB 3. DBW 4	DB 3.DBW 4	WORD	☑

Name	Active
Digital Signal 0	☑
Digital Signal 1	☑
Digital Signal 2	☑
Digital Signal 3	☑
Digital Signal 4	☑
Digital Signal 5	☑
Digital Signal 6	☑
Digital Signal 7	☑
Digital Signal 8	☑
Digital Signal 9	☑
Digital Signal 10	☑
Digital Signal 11	☑
Digital Signal 12	☑
Digital Signal 13	☑
Digital Signal 14	☑
Digital Signal 15	☑

	Decoder	S7 Operand	DataType	Active
3	⊞ DB 3. DBW 6	DB 3.DBW 6	WORD	☑
4	⊞		WORD	□

The individual columns of the signal table have the following meanings:

Source signal

Decoder

Enter a name for the source signal.

S7 Operand/S7 Symbol

Enter the S7 operand to which the signal is assigned and, if applicable, the S7 symbol.

DataType

Enter the data type of the signal. The data type also determines the number of digital signals. *ibaPDA* automatically derives the possible data type from the S7 operand or S7 symbol.

Active
If you activate the source signal, it is acquired with all digital signals. You can deactivate individual digital signals.

Individual digital signals (bits)

Name
Enter a name for the individual digital signals.

Active
If you activate the signal, the signal is acquired and is also considered when checking the number of licensed signals.

Note

ibaPDA only takes the activated digital signals into account for the number of licensed signals, i.e. no additional signal for the source signal.

2.3.3.5 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➚ *Configuration and engineering ibaPDA*, page 62.

Configure the SIMATIC S7 side according to chapter ➚ *Configuration and engineering SIMATIC S7-300, S7-400 and WinAC*, page 53.

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 108.

Note

The module *S7 Request (ibaCom-L2B compatible)* is not released for the connection to controllers of S7-1500 type.

2.3.3.6 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 ➚ *Configuration and engineering ibaPDA*, page 62.

Configure the SIMATIC S7 side according to ➚ *Configuration and engineering SIMATIC S7-300, S7-400 and WinAC*, page 53.

Note

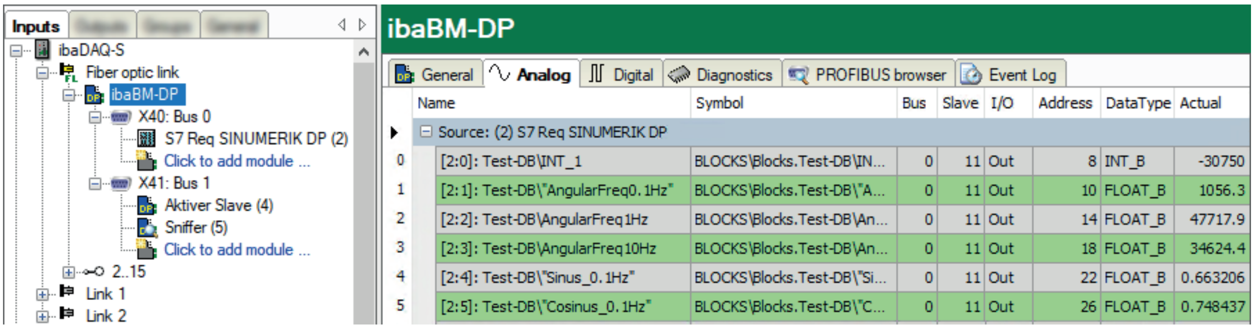
This module type usually is used to replace an existing Request-S7 solution based on an *ibaCom-L2B* card by an *ibaBM-DP* device. For more information about this application case, see chapter ➚ *Replacing Request-S7 on ibaCom-L2B by ibaBM-DP*, page 108.

Note

The module *S7 Request Dig512 (ibaCom-L2B compatible)* is not released for the connection to controllers of S7-1500 type.

2.3.4 Diagnostics

When you select the bus module node in the signal tree and select 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.



	Name	Symbol	Bus	Slave	I/O	Address	DataType	Actual
	Source: (2) S7 Req SINUMERIK DP							
0	[2:0]: Test-DB\INT_1	BLOCKS\Blocks.Test-DB\IN...	0	11	Out	8	INT_B	-30750
1	[2:1]: Test-DB\"AngularFreq0.1Hz"	BLOCKS\Blocks.Test-DB\"A...	0	11	Out	10	FLOAT_B	1056.3
2	[2:2]: Test-DB\AngularFreq1Hz	BLOCKS\Blocks.Test-DB\An...	0	11	Out	14	FLOAT_B	47717.9
3	[2:3]: Test-DB\AngularFreq10Hz	BLOCKS\Blocks.Test-DB\An...	0	11	Out	18	FLOAT_B	34624.4
4	[2:4]: Test-DB\"Sinus_0.1Hz"	BLOCKS\Blocks.Test-DB\"Si...	0	11	Out	22	FLOAT_B	0.663206
5	[2:5]: Test-DB\"Cosinus_0.1Hz"	BLOCKS\Blocks.Test-DB\"C...	0	11	Out	26	FLOAT_B	0.748437

Other documentation

Detailed information about the device-specific diagnostic options of *ibaBM-DP* can be found in the device manual.

2.4 Request-S7 for ibaBM-DP in redundancy mode

This section describes the Request-S7 version for the PROFIBUS bus module *ibaBM-DP* in redundancy mode is described.

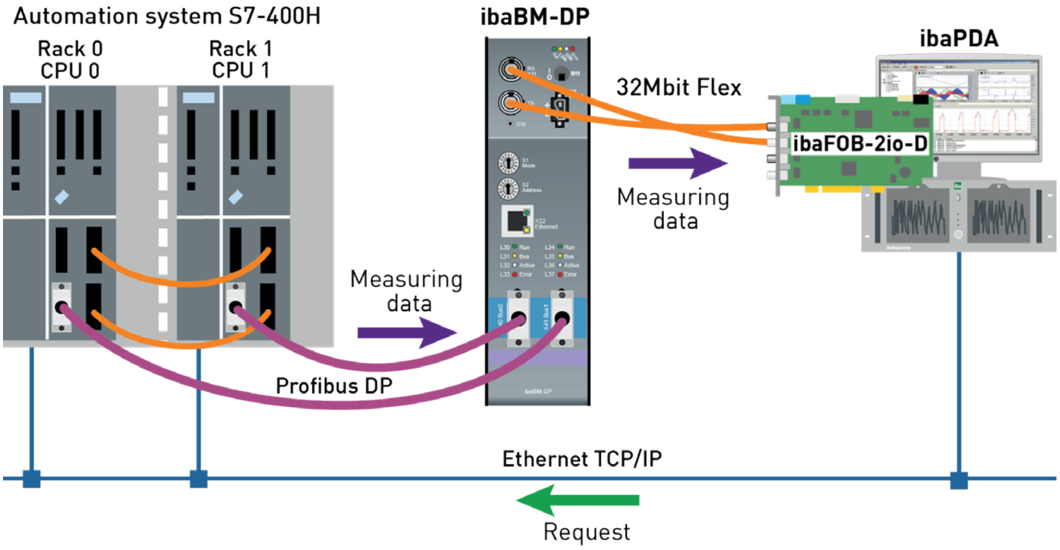
2.4.1 General information

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*. Contact the iba Support. 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 an exemplary integration of an *ibaBM-DP* in redundancy mode.



Note

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

Other documentation

Detailed information about the redundancy mode of the *ibaBM-DP* can be found 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 23). The differences and extensions are described in the following.

For information about application examples, see chapter ➚ *Application examples*, page 94.

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

2.4.2.1 Hardware configuration

Specify a separate ibaBM-PROFIBUS slave for each PROFIBUS master.

Use the GSD file `ibaDPMSi.gsd` (version V2.2 and higher).

Note

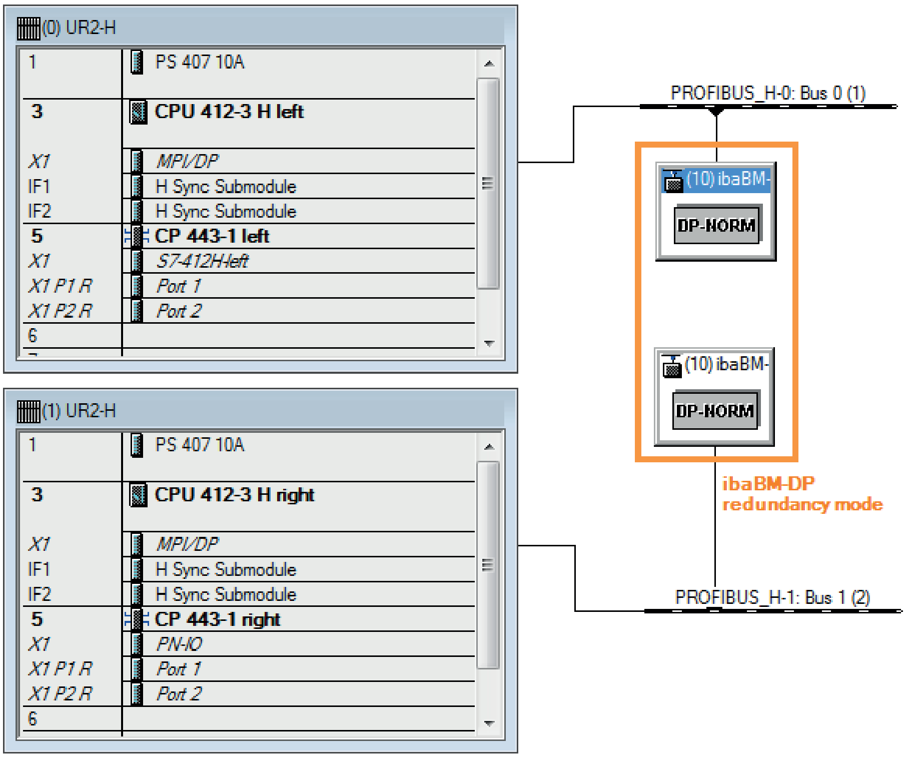
You find the GSD file `ibaDPMSi.gsd` on the data storage medium "iba Software & Manuals" in the following directory:

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

Note

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

The following figure shows an exemplary hardware configuration for a redundant system.



(0) UR2-H

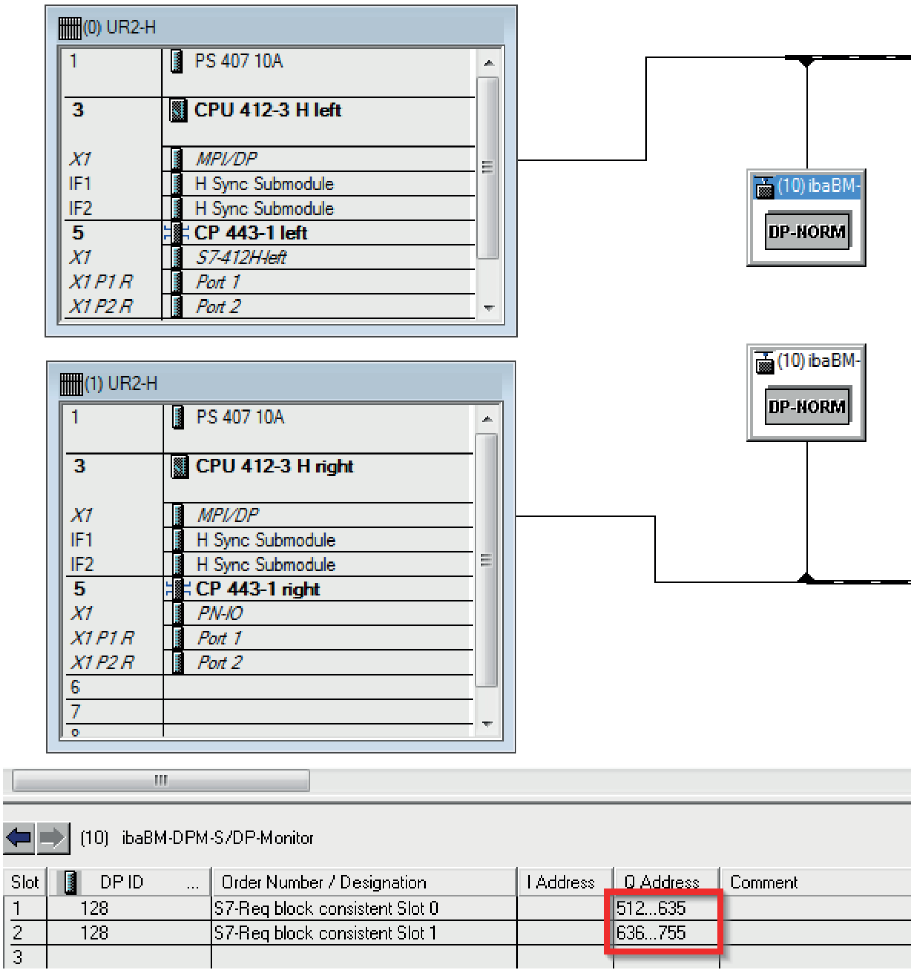
1	PS 407 10A
3	CPU 412-3 H left
X1	*MPI/DP*
IF1	H Sync Submodule
IF2	H Sync Submodule
5	CP 443-1 left
X1	*S7-412H-left*
X1 P1 R	*Port 1*
X1 P2 R	*Port 2*
6	

(1) UR2-H

1	PS 407 10A
3	CPU 412-3 H right
X1	*MPI/DP*
IF1	H Sync Submodule
IF2	H Sync Submodule
5	CP 443-1 right
X1	*PN-IO*
X1 P1 R	*Port 1*
X1 P2 R	*Port 2*
6	

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".

The following figure shows the slave I/O addresses for Bus 0.



(0) UR2-H

1	PS 407 10A
3	**CPU 412-3 H left**
X1	*MPI/DP*
IF1	H Sync Submodule
IF2	H Sync Submodule
5	**CP 443-1 left**
X1	*S7-412H-left*
X1 P1 R	*Port 1*
X1 P2 R	*Port 2*

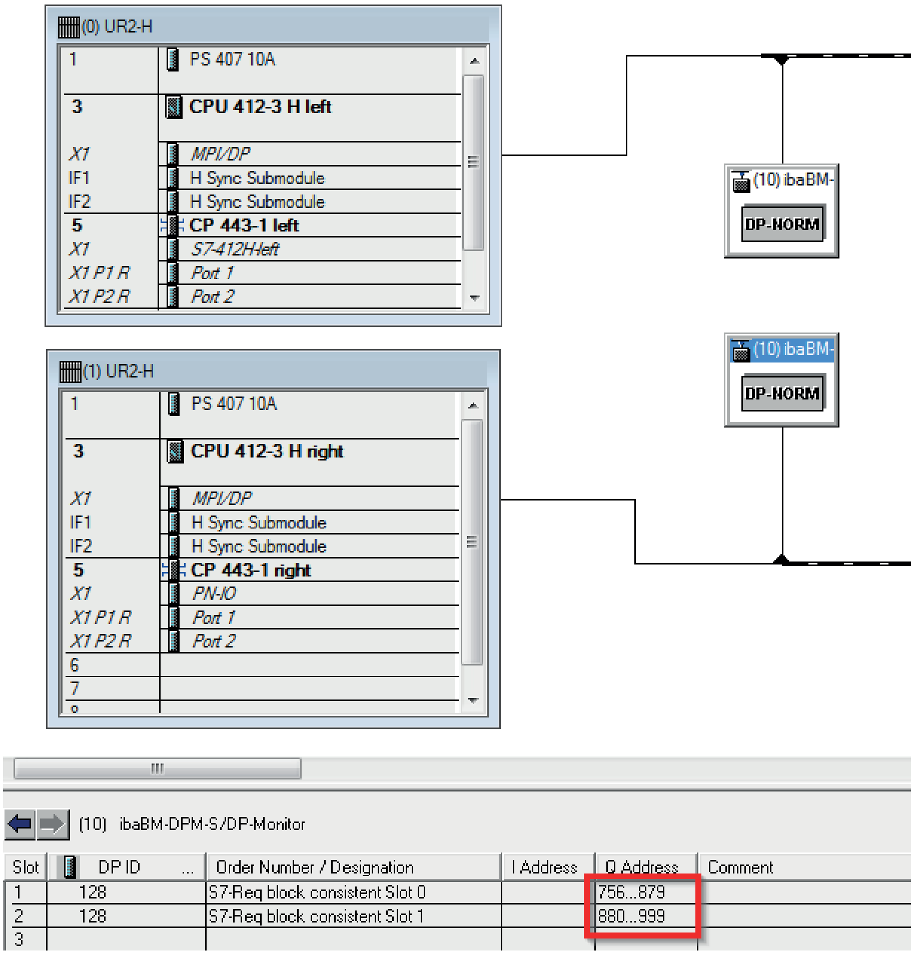
(1) UR2-H

1	PS 407 10A
3	**CPU 412-3 H right**
X1	*MPI/DP*
IF1	H Sync Submodule
IF2	H Sync Submodule
5	**CP 443-1 right**
X1	*PN-IO*
X1 P1 R	*Port 1*
X1 P2 R	*Port 2*
6	
7	

(10) ibaBM-DPM-S/DP-Monitor

Slot	DP ID	Order Number / Designation	I Address	Q Address	Comment
1	128	S7-Req block consistent Slot 0		512...635	
2	128	S7-Req block consistent Slot 1		636...755	
3					

The following figure shows the slave I/O addresses for Bus 1.



(0) UR2-H

1	PS 407 10A
3	**CPU 412-3 H left**
X1	*MPI/DP*
IF1	H Sync Submodule
IF2	H Sync Submodule
5	**CP 443-1 left**
X1	*S7-412H-left*
X1 P1 R	*Port 1*
X1 P2 R	*Port 2*

(1) UR2-H

1	PS 407 10A
3	**CPU 412-3 H right**
X1	*MPI/DP*
IF1	H Sync Submodule
IF2	H Sync Submodule
5	**CP 443-1 right**
X1	*PN-IO*
X1 P1 R	*Port 1*
X1 P2 R	*Port 2*
6	
7	

(10) ibaBM-DPM-S/DP-Monitor

Slot	DP ID	Order Number / Designation	I Address	Q Address	Comment
1	128	S7-Req block consistent Slot 0		756...879	
2	128	S7-Req block consistent Slot 1		880...999	
3					

2.4.2.2 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

1. Copy the following blocks from the iba S7 library to the blocks folder of your STEP 7 project, see ↗ *iba S7 library*, page 91. The ibaDP_Req_H function has to be called once per slave pair in the cyclic program.
   - FC123 (ibaDP_Req_H) , see ↗ *Request-FC ibaDP_Req_H (FC123)*, page 76
   - DB10 (ibaDP_DB_PDA)
   - DB25 (ibaDP_DB_work)

Note

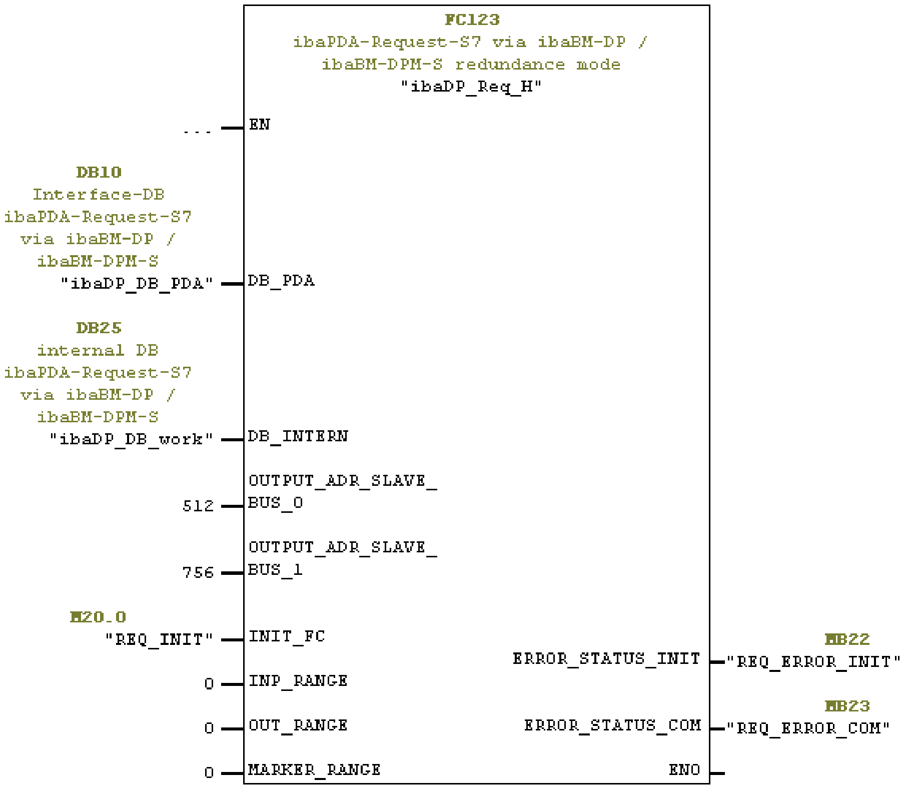
Only use Request blocks from the latest iba S7 library!

Request blocks in application examples can be outdated and, thus, cause errors.

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.

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



FC123
ibaPDA-Request-S7 via ibaBM-DP /
ibaBM-DPM-S redundance mode
"ibaDP_Req_H"

... — EN

DB10
Interface-DB
ibaPDA-Request-S7
via ibaBM-DP /
ibaBM-DPM-S
"ibaDP_DB_PDA" — DB_PDA

DB25
internal DB
ibaPDA-Request-S7
via ibaBM-DP /
ibaBM-DPM-S
"ibaDP_DB_work" — DB_INTERN

512 — OUTPUT_ADR_SLAVE_BUS_0

756 — OUTPUT_ADR_SLAVE_BUS_1

M20.0
"REQ_INIT" — INIT_FC

ERROR_STATUS_INIT — MB22 "REQ_ERROR_INIT"

0 — INP_RANGE

0 — OUT_RANGE

ERROR_STATUS_COM — MB23 "REQ_ERROR_COM"

0 — MARKER_RANGE

ENO

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.

Final

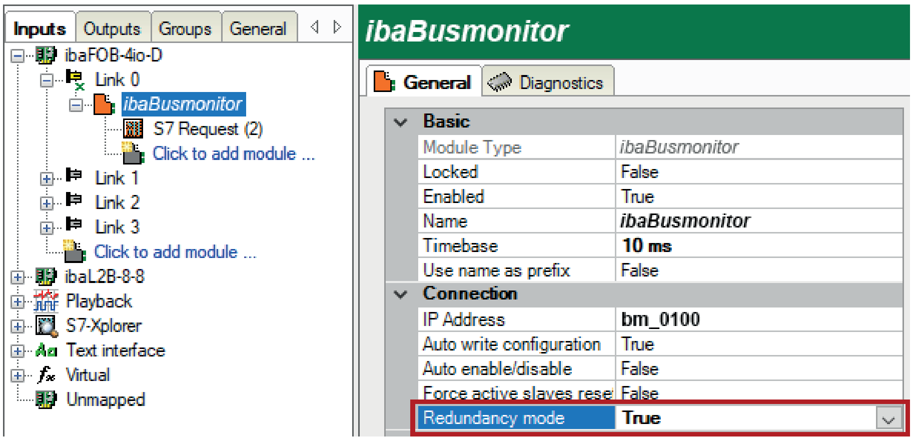
- Load all blocks into the S7 CPU and restart the S7 CPU.

2.4.3 Configuration and engineering ibaPDA

First 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.

In the *General* tab, set the parameter *Redundancy mode* to "True". The orange colored icon of the device module indicates the redundancy mode.

You can operate the device in the 32Mbit Flex mode as well as in compatibility mode with 32Mbit.



Basic	
Module Type	*ibaBusmonitor*
Locked	False
Enabled	True
Name	***ibaBusmonitor***
Timebase	**10 ms**
Use name as prefix	False
Connection	
IP Address	**bm_0100**
Auto write configuration	True
Auto enable/disable	False
Force active slaves rese	False
Redundancy mode	**True**

As on the redundant PROFIBUS both bus systems 0 and 1 are operated in parallel, there are only settings for *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)

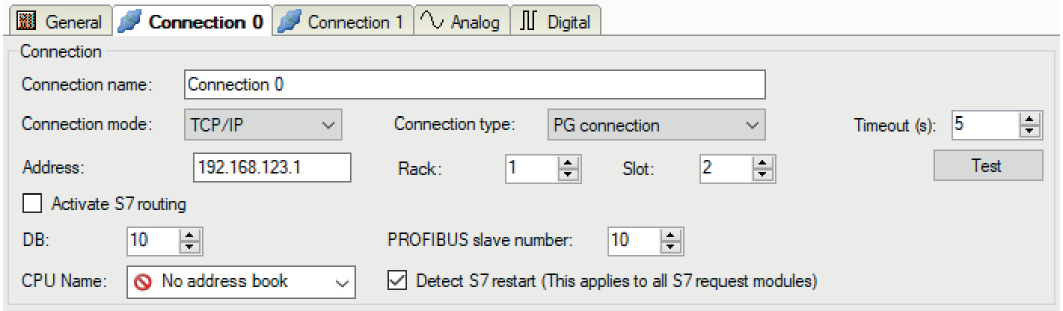
These modules are deactivated automatically and cannot be activated manually, either.

The configuration of the modules corresponds to the configuration in standard mode, see ➚ *General module settings*, page 33.

Connection settings

For transmitting the operand data to both CPUs of the SIMATIC S7-400H, set up two different connections. *ibaPDA* switches between these connections depending on availability and requirements.

Configure both connections on the tabs *Connection 0* and *Connection 1*. You can change the name of the tabs in the *Connection name* fields.



Connection

Connection name: Connection 0

Connection mode: TCP/IP

Connection type: PG connection

Timeout (s): 5

Address: 192.168.123.1

Rack: 1

Slot: 2

□ Activate S7 routing

DB: 10

PROFIBUS slave number: 10

CPU Name: No address book

☑ Detect S7 restart (This applies to all S7 request modules)

The other settings are identical to the settings for operation without redundancy mode, see general chapter *Connection settings* in part 1 of the manual.

Note these special features:

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

2.5 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.

2.5.1 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 using an integrated PN interface of the S7-CPU or the respective CP modules in the PLC and the standard network interface of the computer. 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, for the PC/CP group the SIMATIC communication software (and the corresponding licenses) must be installed on the computer with the corresponding functions enabled.
    - MPI, PROFIBUS: The connection to the SIMATIC S7 is established using the MPI or PROFIBUS interface of the computer; e.g. with the CP5611 PCI card or the MPI adapter for USB interface or serial PC interface.
    - TCP/IP, ISO: Here, either the standard network interface of the computer or a suitable interface card is used for the connection to the S7.

System integration with ibaBM-DPM-S

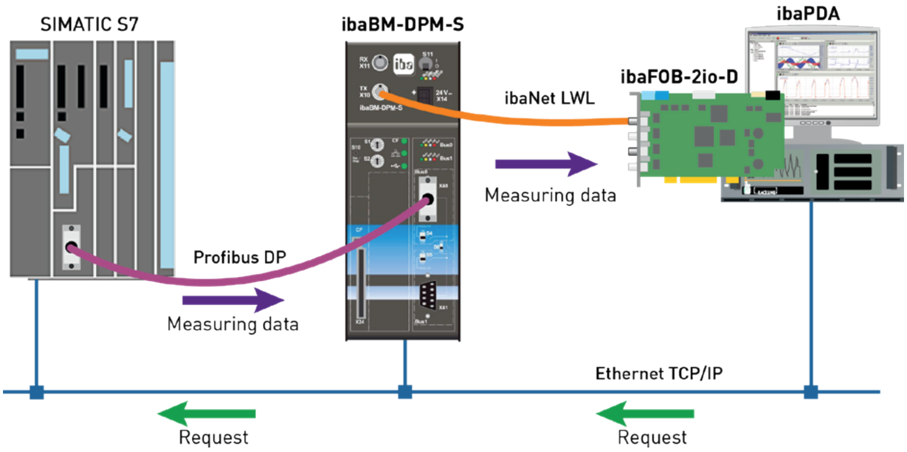
The measured data is transmitted to the *ibaBM-DPM-S* device over PROFIBUS DP.

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-io-D* and *ibaBM-DPM-S*
- PROFIBUS connection between *ibaBM-DPM-S* and S7 PROFIBUS master

In the standard version, you can configure a maximum of 8 connections per device, i.e. 8 PROFIBUS slaves. 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.



Note

The following description refers to the Request blocks V4.0 or higher. In case you require information regarding older versions, contact the iba Support.

Other documentation

Detailed information about *ibaBM-DPM-S* can be found in the device manual.

For information about application examples, see chapter ↗ *Application examples*, page 94.

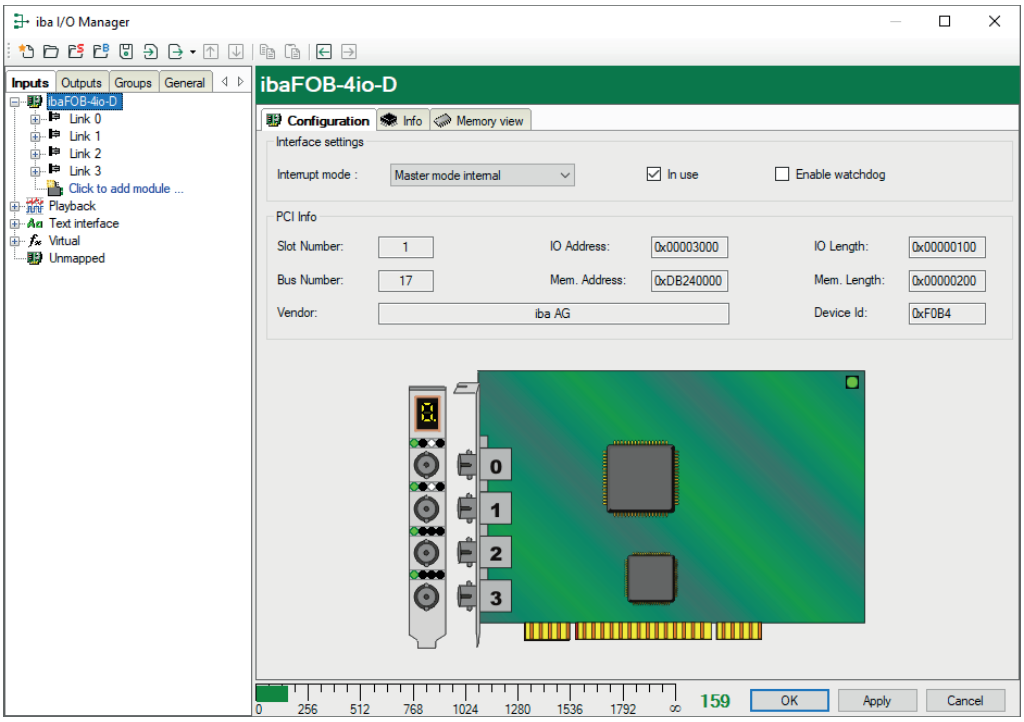
2.5.2 Configuration and engineering SIMATIC S7-300, S7-400 and WinAC

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 25).

2.5.3 Configuration and engineering ibaPDA

2.5.3.1 General interface settings

If an *ibaFOB-D* card is installed in the *ibaPDA* computer, *ibaPDA* displays the interface of the *ibaFOB-D* card in the interface tree of the I/O Manager.



Interface settings

Interrupt mode : Master mode internal ☑ In use ☐ Enable watchdog

PCI Info

Slot Number:	1	IO Address:	0x00003000	IO Length:	0x00000100
Bus Number:	17	Mem. Address:	0xDB240000	Mem. Length:	0x00000200
Vendor:	iba AG			Device Id:	0xF0B4

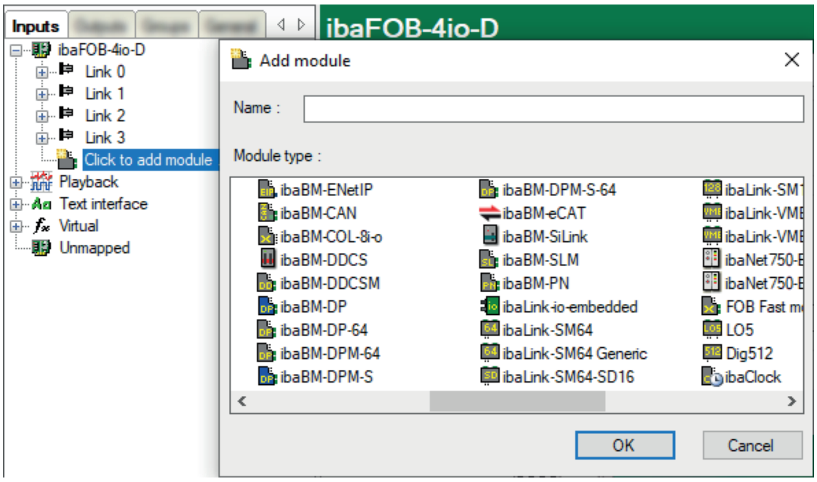
Other documentation

Detailed information about the *ibaFOB-D* card can be found in the respective manual.

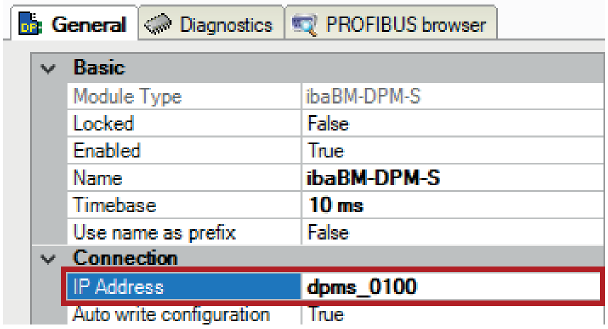
Configuring ibaBM-DPM-S

1. Start the *ibaPDA* client and open the I/O Manager.
2. In the interface tree (left), select the link of the *ibaFOB* card to which *ibaBM-DPM-S* is connected.

   Click on the blue command *Click to add module*. Select the *ibaBM-DPM-S* module in the dialog box and assign a name via the input field if required. Confirm the selection with <OK>.



3. Enter the IP address of the device on the *General* tab under *Connection*: either as name e.g. "dpms_000100" or as IP address e.g. "192.168.81.123".

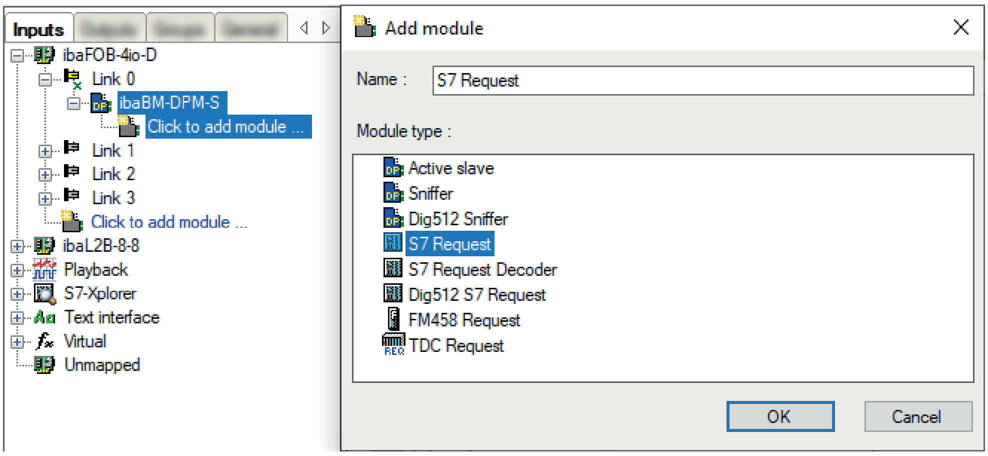


Basic	
Module Type	ibaBM-DPM-S
Locked	False
Enabled	True
Name	**ibaBM-DPM-S**
Timebase	**10 ms**
Use name as prefix	False
Connection	
IP Address	**dpms_0100**
Auto write configuration	True

Note

An online connection to the device via Ethernet or USB is mandatory. The manual of the *ibaBM-DPMS* device explains how to 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-CPUs) to the *ibaBM-DPM-S* module. 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 Dig512 (for acquiring up to 512 digital signals)



Name : S7 Request

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*.

2.5.3.2 General module settings

You find the description of all settings that are identical for all Request-S7 modules in part 1 of the manual.

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

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.

2.5.3.3 Connection settings

The connection settings are identical to those in the general chapter *Connection settings* in part 1.

PROFIBUS slave number
PROFIBUS slave address assigned to the module. If you change the entry in the *Connection* tab, the entry in the *General* tab changes accordingly and vice versa.

2.5.3.4 Module S7 Request

With this module type, up to 64 analog and 64 digital signals can be acquired.

Configure a separate PROFIBUS slave and Request block call for each module.

For a description of the module settings, see chapter ↗ *General module settings*, page 48 and in the general chapter *General module settings* in part 1 of this manual.

2.5.3.5 Module 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. 512 directly addressable digital values of the *ibaBM-DPM-S* are not sufficient.

For a detailed description of the *S7 Request Decoder* module, see the module description of the *ibaBM-DP* device in part 1 of the manual.

2.5.3.6 Module Dig512 S7 Request

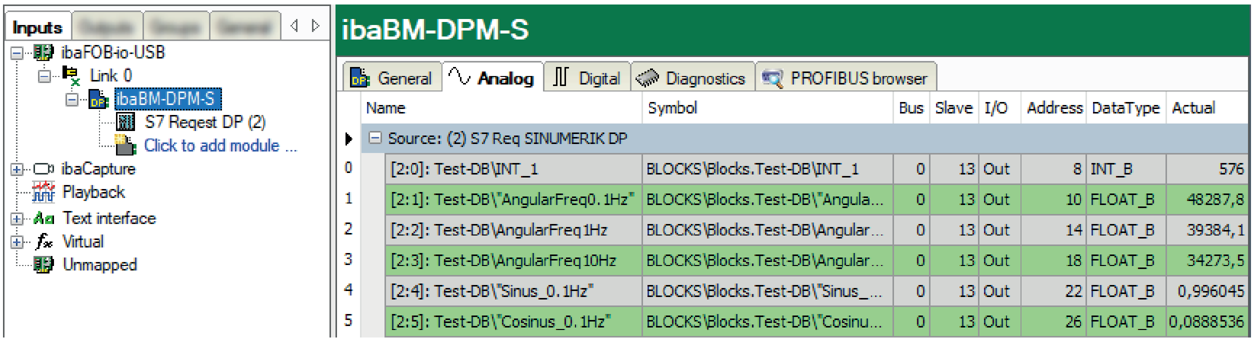
For the *Dig512 S7 request* module the same applies as for the module *S7 Request Decoder*. But you can only acquire 32 words (512 digital values).

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 ↗ *Modules S7 Request Decoder*, page 49.

2.5.4 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.



Source: (2) S7 Req SINUMERIK DP

	Name	Symbol	Bus	Slave	I/O	Address	DataType	Actual
0	[2:0]: Test-DB\INT_1	BLOCKS\Blocks.Test-DB\INT_1	0	13	Out	8	INT_B	576
1	[2:1]: Test-DB\"AngularFreq0.1Hz"	BLOCKS\Blocks.Test-DB\"Angula...	0	13	Out	10	FLOAT_B	48287,8
2	[2:2]: Test-DB\AngularFreq1Hz	BLOCKS\Blocks.Test-DB\Angular...	0	13	Out	14	FLOAT_B	39384,1
3	[2:3]: Test-DB\AngularFreq10Hz	BLOCKS\Blocks.Test-DB\Angular...	0	13	Out	18	FLOAT_B	34273,5
4	[2:4]: Test-DB\"Sinus_0.1Hz"	BLOCKS\Blocks.Test-DB\"Sinus_...	0	13	Out	22	FLOAT_B	0,996045
5	[2:5]: Test-DB\"Cosinus_0.1Hz"	BLOCKS\Blocks.Test-DB\"Cosinu...	0	13	Out	26	FLOAT_B	0,0888536

Other documentation

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

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

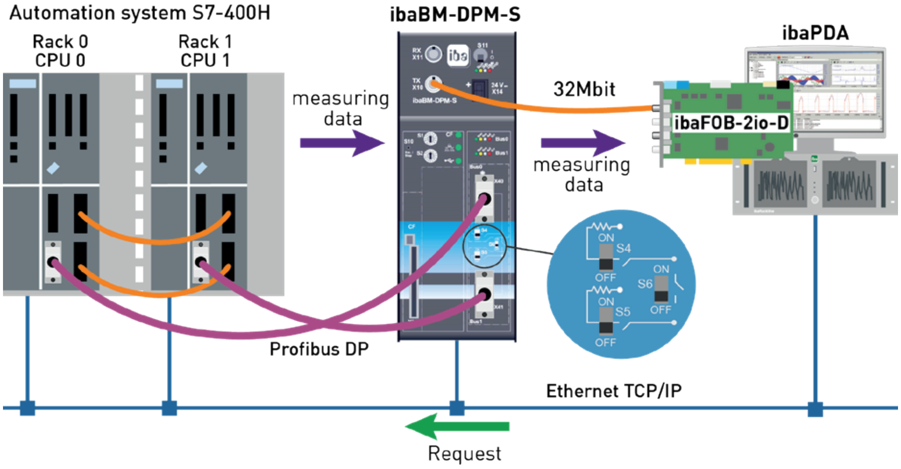
In the following, the Request-S7 version for the PROFIBUS bus module *ibaBM-DPM-S* in redundancy mode is described. The solution "Request-S7 for *ibaBM-DPM-S* in redundancy mode" is replaced by "Request-S7 for *ibaBM-DP* in redundancy mode" with full functional compatibility.

2.6.1 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*. Contact the iba Support. 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 an exemplary integration of an *ibaBM-DPM-S* in redundancy mode.



Note

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

Other documentation

Detailed information about the redundancy mode of *ibaBM-DPM-S* can be found in 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 44). The differences and extensions are described in the following.

For information about application examples, see chapter ➚ *Application examples*, page 94.

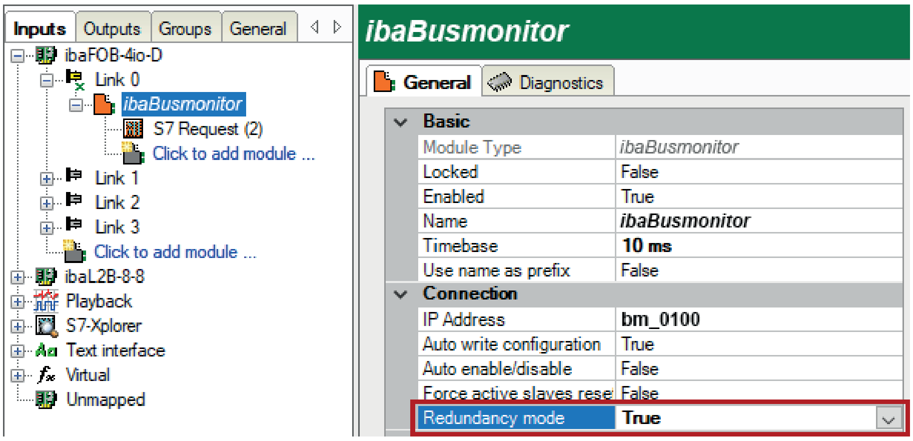
2.6.2 Configuration and engineering SIMATIC S7-300, S7-400 and WinAC

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 38).

2.6.3 Configuration and engineering ibaPDA

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

In the *General* tab, set the parameter *Redundancy mode* to "True". The orange colored icon of the Request module indicates the redundancy mode.



Basic	
Module Type	ibaBusmonitor
Locked	False
Enabled	True
Name	ibaBusmonitor
Timebase	10 ms
Use name as prefix	False
Connection	
IP Address	bm_0100
Auto write configuration	True
Auto enable/disable	False
Force active slaves rese	False
Redundancy mode	True

As on the redundant PROFIBUS, both bus systems 0 and 1 are operated in parallel, there are only settings for *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, see ↗ *General module settings*, page 48.

Connection settings

For transmitting the operand data to both CPUs of the SIMATIC S7-400H, set up two different connections. *ibaPDA* switches between these connections 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 38).

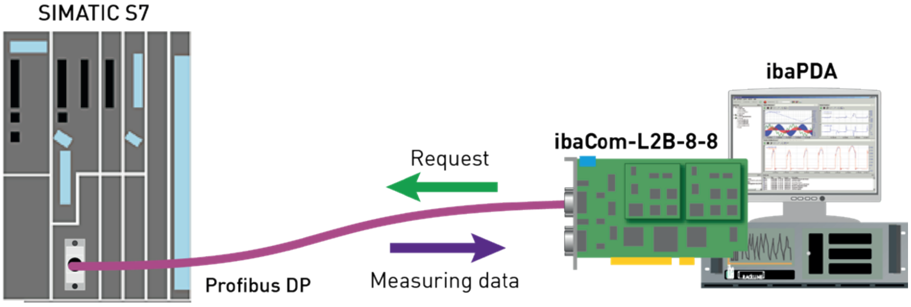
2.7 Request-S7 for ibaCom-L2B

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 108.

2.7.1 General information



With the L2B Request, the measurement request (Request Handshake) 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. In case you require information regarding older versions, contact the iba Support.

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

For information about application examples, see chapter ↗ *Application examples*, page 94.

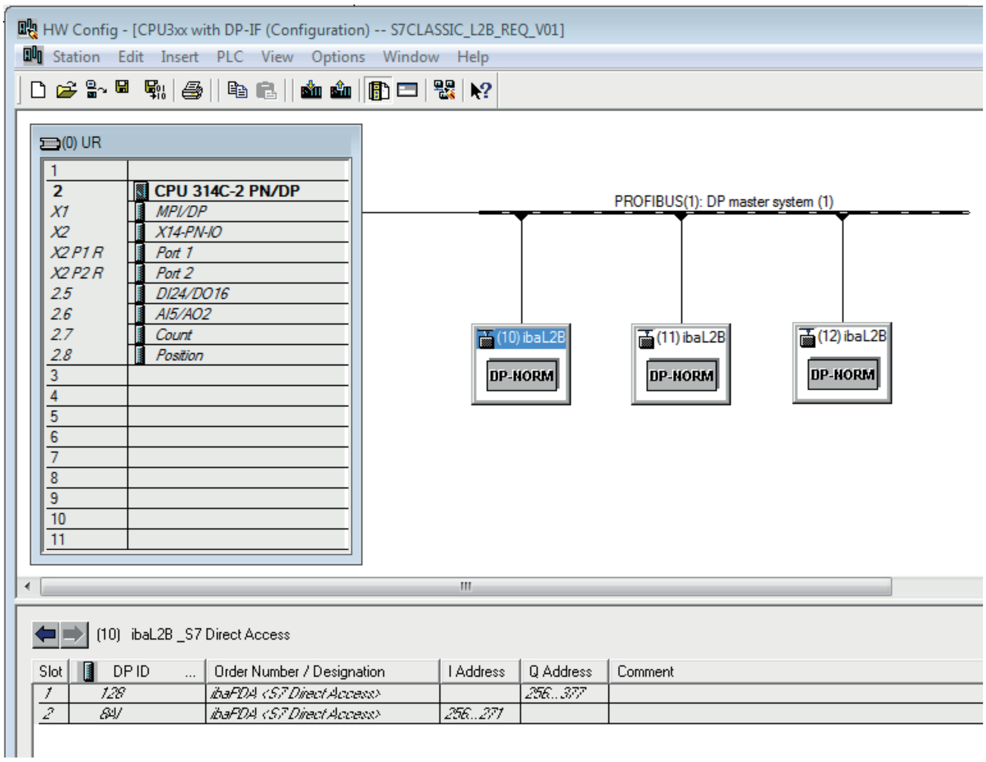
2.7.2 Configuration and engineering SIMATIC S7-300, S7-400 and WinAC

2.7.2.1 Hardware configuration

Perform the following steps:

1. Link the DP interface of the CPU or CP as DP master to a PROFIBUS network.
2. Install the GSD file `IBA_0F05.GSD` via the menu *Options – Installing GSD-files*.
3. After the installation finished, the iba-GSD-file is in the HW catalog under:
   Profibus-DP – Additional Field Devices – General – ibaL2B <S7 Direct Access>
4. 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.

5. Load the HW config/system data to the S7-CPU.


HW Config - [CPU3xx with DP-IF (Configuration) -- S7CLASSIC_L2B_REQ_V01]
Station Edit Insert PLC View Options Window Help
(0) UR
PROFIBUS(1): DP master system (1)
(10) ibaL2B
DP-NORM
(11) ibaL2B
DP-NORM
(12) ibaL2B
DP-NORM

1	
2	CPU 314C-2 PN/DP
X1	MPI/DP
X2	X14-PN-IO
X2 P1 R	Port 1
X2 P2 R	Port 2
2.5	DI24/DO16
2.6	AI5/AO2
2.7	Count
2.8	Position
3	
4	
5	
6	
7	
8	
9	
10	
11	

(10) ibaL2B _S7 Direct Access

Slot	DP ID	Order Number / Designation	I Address	Q Address	Comment
1	128	ibaPDA <S7 Direct Access>		256...377	
2	84I	ibaPDA <S7 Direct Access>	256...271		

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

2.7.2.2.1 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.

If you use the external DP interface CP342-5 of a CPU S7-300, continue in chapter ↗ *External DP interface CP342-5*, page 56.

For Request-S7 with one PROFIBUS slave

1. Copy the following blocks from the iba S7 library to the blocks folder of your STEP 7 project, see ↗ *iba S7 library*, page 91. The functions have to be called once for each activated PROFIBUS slave in the program.
    - FC111 (ibaL2B_Init) and FC112 (ibaL2B_Req), see ↗ *Initialization-FC ibaL2B_Init (FC111)*, page 79 and ↗ *Communication-FC ibaL2B_Req (FC112)*, page 84
    - DB22 (ibaL2B_DB_work) and UDT22 (ibaL2B_DB_Struct)

Note

Only use Request blocks from the latest iba S7 library!

Request blocks in application examples can be outdated and, thus, cause errors.

Note

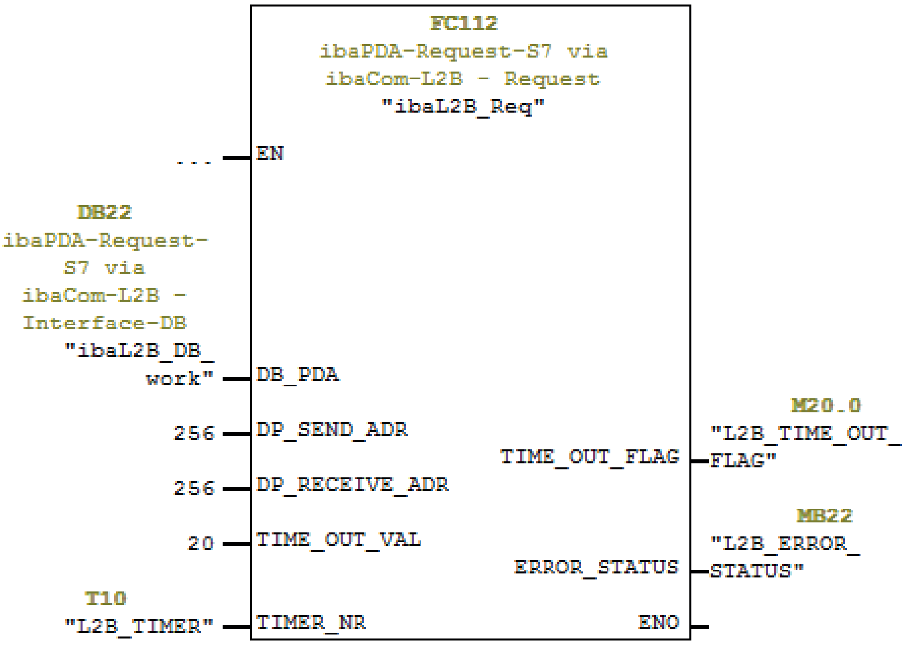
In case the block numbers FC111, FC112, DB22 and UDT22 are already used in your project, assign other free numbers to the blocks from the iba S7 library when copying.

2. 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.

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



FC112
ibaPDA-Request-S7 via ibaCom-L2B - Request
"ibaL2B_Req"

... — EN

DB22 ibaPDA-Request-S7 via ibaCom-L2B - Interface-DB "ibaL2B_DB_work" — DB_PDA

256 — DP_SEND_ADR

256 — DP_RECEIVE_ADR

20 — TIME_OUT_VAL

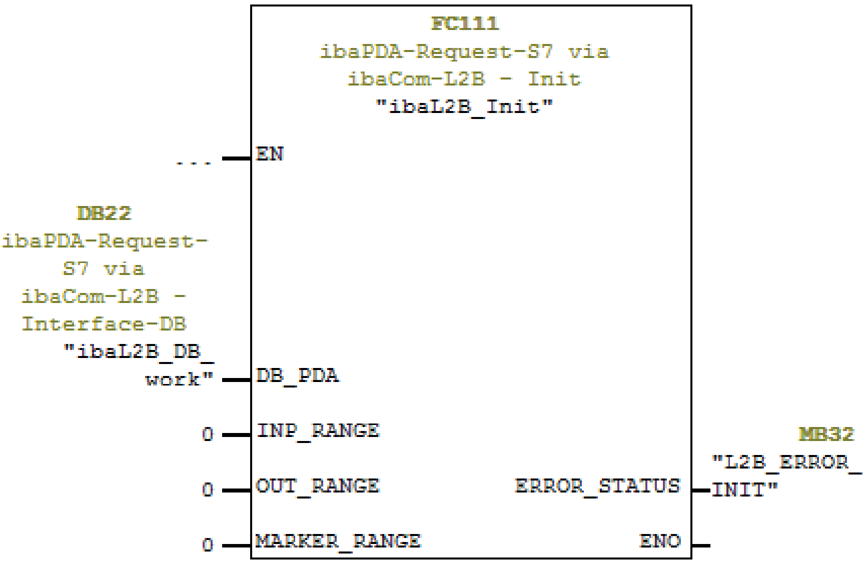
T10 "L2B_TIMER" — TIMER_NR

TIME_OUT_FLAG — M20.0 "L2B_TIME_OUT_FLAG"

ERROR_STATUS — MB22 "L2B_ERROR_STATUS"

ENO —

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



FC111
ibaPDA-Request-S7 via ibaCom-L2B - Init
"ibaL2B_Init"

... — EN

DB22 ibaPDA-Request-S7 via ibaCom-L2B - Interface-DB "ibaL2B_DB_work" — DB_PDA

0 — INP_RANGE

0 — OUT_RANGE

0 — MARKER_RANGE

ERROR_STATUS — MB32 "L2B_ERROR_INIT"

ENO —

For Request-S7 with multiple PROFIBUS slaves

- In the blocks folder, a data block (DB) that is referenced by both FCs has to be available for each Request slave. 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.

Final

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

2.7.2.2.2 External DP interface CP342-5

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

For Request-S7 with one PROFIBUS slave

1. Copy the following blocks from the iba S7 library to the blocks folder of your STEP 7 project, see ➚ *iba S7 library*, page 91. The functions have to be called once for each activated PROFIBUS slave in the program.
   - FC113 (ibaL2B_Req_CP) and FC111 (ibaL2B_Init), see ➚ *Communication-FC ibaL2B_Req_CP (FC113) for CP342-5*, page 82 and ➚ *Initialization-FC ibaL2B_Init (FC111)*, page 79
   - DB10 (ibaL2B_CP_SNDRCV), DB22 (ibaL2B_DB_work) and UDT22 (ibaL2B_DB_Struct)

Note

Only use Request blocks from the latest iba S7 library!

Request blocks in application examples can be outdated and, thus, cause errors.

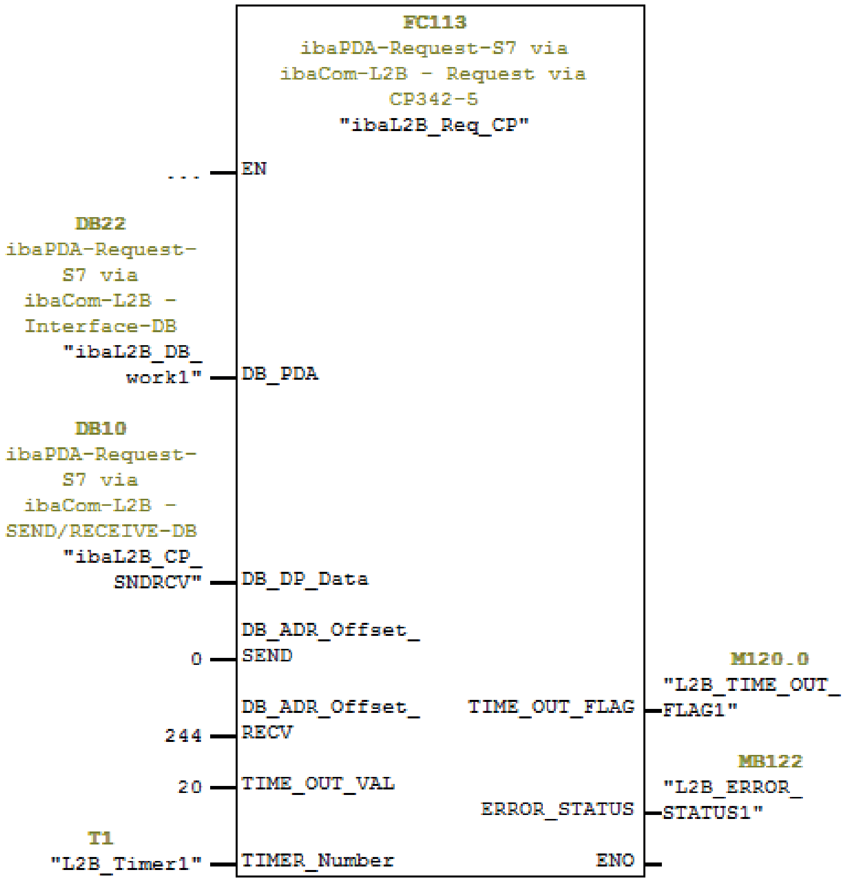
Note

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

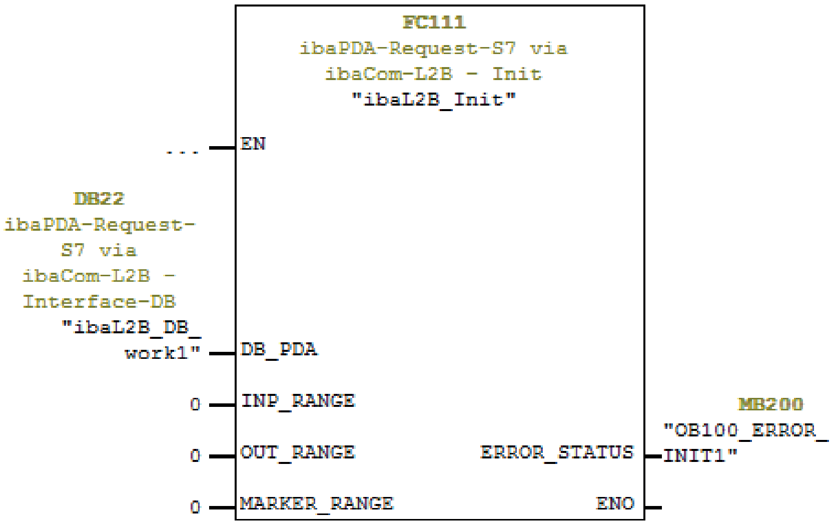
2. Create the error OBs (OB82, OB85, OB86, OB87, OB122) in order to prevent CPU stops in case of an error.

3. Call and parameterize the ibaL2B_Req_CP (FC113) in the cyclic program.

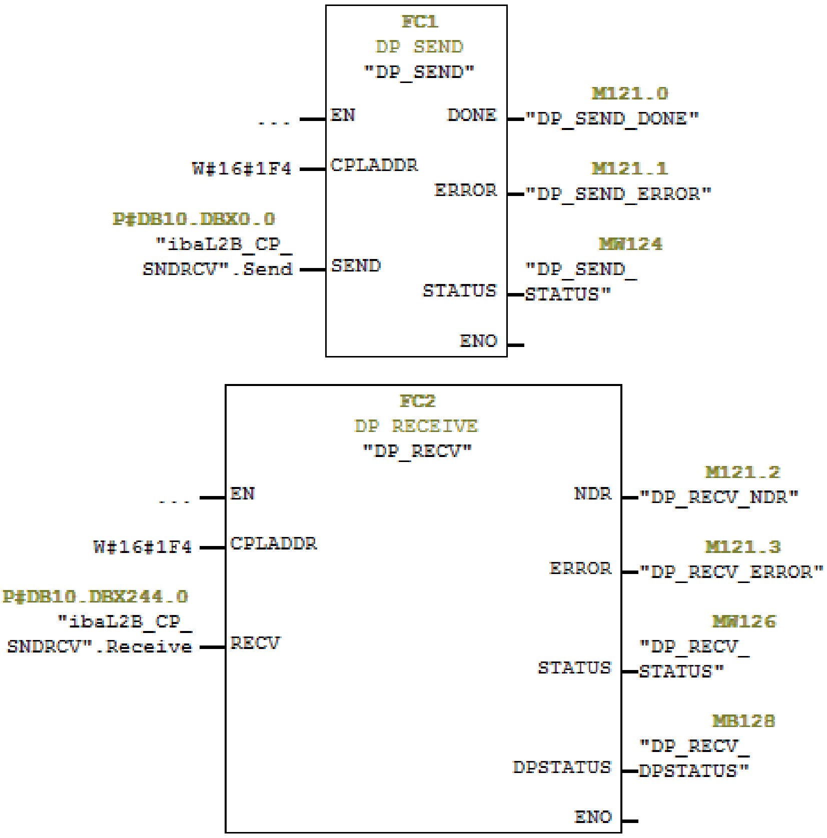
   As offset addresses for send and receive data, own ranges within the DB "ibaPDA_SENDRECV" (DB10) have to be defined for each slave and handed over as parameters
   "DB_ADR_Offset_SEND" or "DB_ADR_Offset_RECV".



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



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



For Request-S7 with multiple PROFIBUS slaves

- In the blocks folder, a data block (DB) that is referenced by both FCs has to be available for each Request slave. 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.

Final

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

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, you have to extend the arrays in DB10 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!

2.7.2.3 Configuration in STEP 7 (CFC)

2.7.2.3.1 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.

If you use the external DP interface CP342-5 of a CPU S7-300, continue in chapter ↗ *External DP interface CP342-5*, page 56.

For Request-S7 with one PROFIBUS slave

1. Copy the following blocks from the iba S7 library to the blocks folder of your STEP 7 project, see ↗ *iba S7 library*, page 91. The functions have to be called once for each activated PROFIBUS slave in the program.
   - FC111 (ibaL2B_Init) and FC112 (ibaL2B_Req), see ↗ *Initialization-FC ibaL2B_Init (FC111)*, page 79 and ↗ *Communication-FC ibaL2B_Req (FC112)*, page 84
   - DB22 (ibaL2B_DB_work) and UDT22 (ibaL2B_DB_Struct)

Note

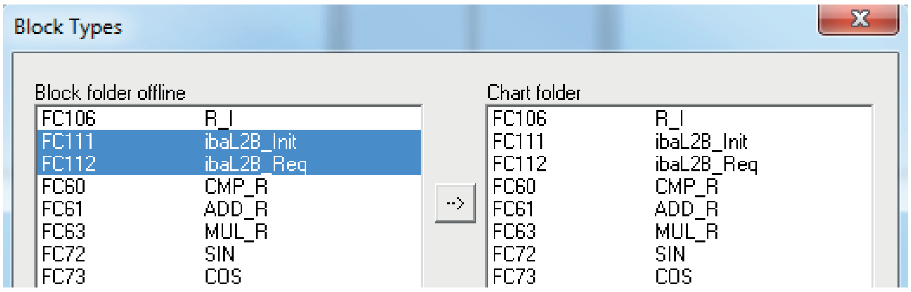
Only use Request blocks from the latest iba S7 library!

Request blocks in application examples can be outdated and, thus, cause errors.

Note

If 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.

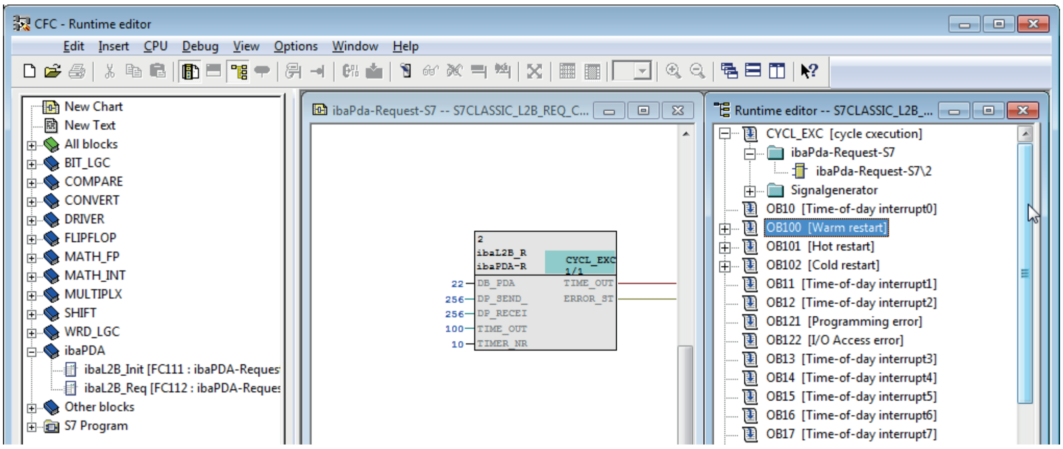
2. Start the CFC editor and import these blocks (under *Options – Block Types*).
   The FCs are stored in the group "ibaPDA" in the function block library.



Block Types

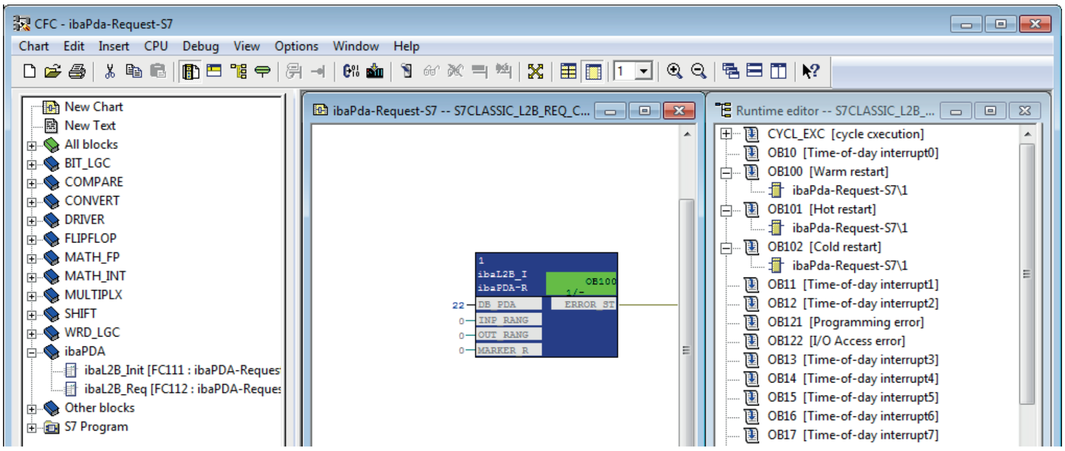
Block folder offline		Chart folder	
FC106	R_I	FC106	R_I
FC111	ibaL2B_Init	FC111	ibaL2B_Init
FC112	ibaL2B_Req	FC112	ibaL2B_Req
FC60	CMP_R	FC60	CMP_R
FC61	ADD_R	FC61	ADD_R
FC63	MUL_R	FC63	MUL_R
FC72	SIN	FC72	SIN
FC73	COS	FC73	COS

3. 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).



CFC - Runtime editor
Edit Insert CPU Debug View Options Window Help
New Chart
New Text
All blocks
BIT_LGC
COMPARE
CONVERT
DRIVER
FLIPFLOP
MATH_FP
MATH_INT
MULTIPLX
SHIFT
WRD_LGC
ibaPDA
ibaL2B_Init [FC111 : ibaPDA-Reques
ibaL2B_Req [FC112 : ibaPDA-Reques
Other blocks
S7 Program
ibaPda-Request-S7 -- S7CLASSIC_L2B_REQ_C...
2
ibaL2B_R
ibaPDA-R
CYCL_EXC
1/1
22 DB_PDA
256 DP_SEND_
256 DP_RECEI
100 TIME_OUT
10 TIMER_NR
TIME_OUT
ERROR_ST
Runtime editor -- S7CLASSIC_L2B_...
CYCL_EXC [cycle execution]
ibaPda-Request-S7
ibaPda-Request-S7\2
Signalgenerator
OB10 [Time-of-day interrupt0]
OB100 [Warm restart]
OB101 [Hot restart]
OB102 [Cold restart]
OB11 [Time-of-day interrupt1]
OB12 [Time-of-day interrupt2]
OB121 [Programming error]
OB122 [I/O Access error]
OB13 [Time-of-day interrupt3]
OB14 [Time-of-day interrupt4]
OB15 [Time-of-day interrupt5]
OB16 [Time-of-day interrupt6]
OB17 [Time-of-day interrupt7]

4. 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).

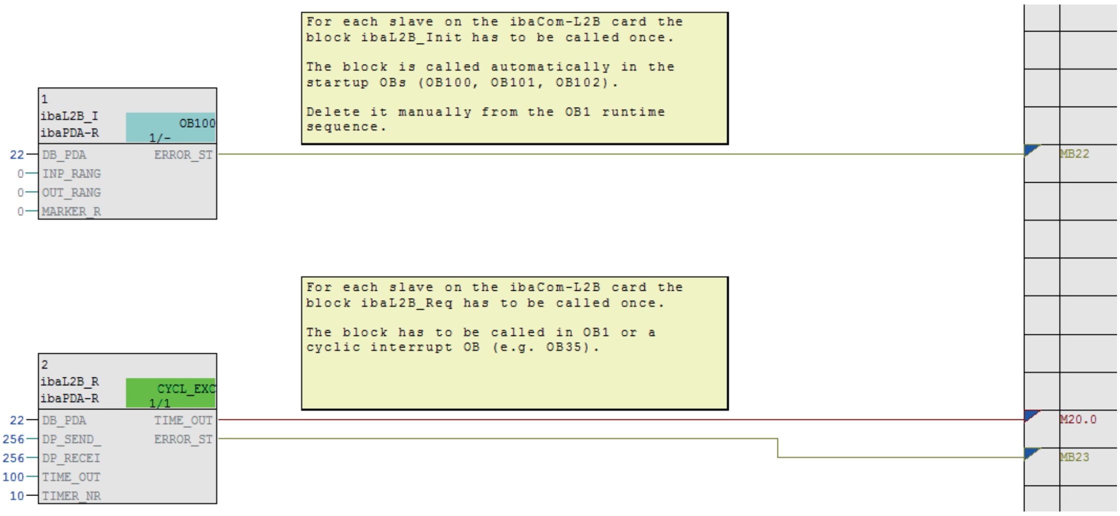


CFC - ibaPda-Request-S7
Chart Edit Insert CPU Debug View Options Window Help
New Chart
New Text
All blocks
BIT_LGC
COMPARE
CONVERT
DRIVER
FLIPFLOP
MATH_FP
MATH_INT
MULTIPLX
SHIFT
WRD_LGC
ibaPDA
ibaL2B_Init [FC111 : ibaPDA-Reques
ibaL2B_Req [FC112 : ibaPDA-Reques
Other blocks
S7 Program
ibaPda-Request-S7 -- S7CLASSIC_L2B_REQ_C...
1
ibaL2B_I
ibaPDA-R
OB100
1/-
22 DB_PDA
0 INP_RANG
0 OUT_RANG
0 MARKER_R
ERROR_ST
Runtime editor -- S7CLASSIC_L2B_...
CYCL_EXC [cycle execution]
OB10 [Time-of-day interrupt0]
OB100 [Warm restart]
ibaPda-Request-S7\1
OB101 [Hot restart]
ibaPda-Request-S7\1
OB102 [Cold restart]
ibaPda-Request-S7\1
OB11 [Time-of-day interrupt1]
OB12 [Time-of-day interrupt2]
OB121 [Programming error]
OB122 [I/O Access error]
OB13 [Time-of-day interrupt3]
OB14 [Time-of-day interrupt4]
OB15 [Time-of-day interrupt5]
OB16 [Time-of-day interrupt6]
OB17 [Time-of-day interrupt7]

5. Connect the blocks:

   a.) Enter the DB number of the Interface-DB (DB22) on both FCs on the input DB_NUMBER.

   b.) 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.



```
For each slave on the ibaCom-L2B card the
block ibaL2B_Init has to be called once.

The block is called automatically in the
startup OBs (OB100, OB101, OB102).

Delete it manually from the OB1 runtime
sequence.
```

```
For each slave on the ibaCom-L2B card the
block ibaL2B_Req has to be called once.

The block has to be called in OB1 or a
cyclic interrupt OB (e.g. OB35).
```

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 are 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

1. Compile the complete program:

   a.) Activate the option *Entire program*.

   b.) Activate the option *Generate SCL source* (for SIMATIC CFC version 6.1 or higher).

   c.) Deactivate the option *Delete empty runtime groups* if available.

2. Load the program and restart the S7-CPU.

Note

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

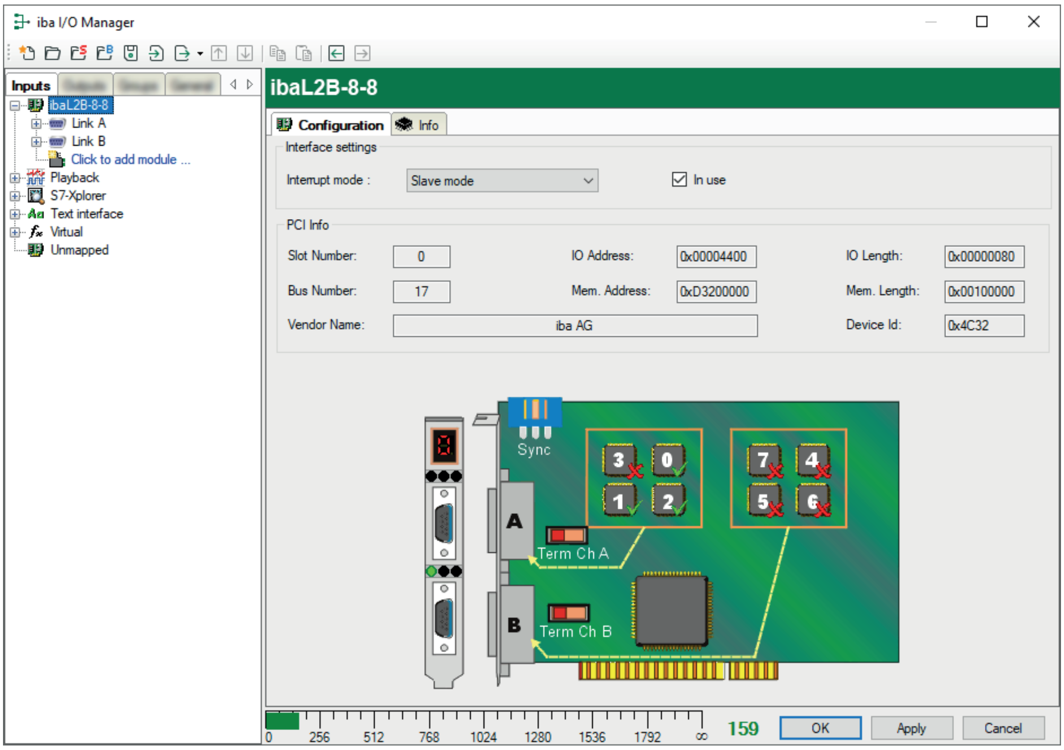
- "W: OB101/ OB102 is not supported by this CPU"
- "W: OB101/OB102 could not be downloaded, because it is not supported by the connected online CPU".

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. You can ignore the warnings.

2.7.3 Configuration and engineering ibaPDA

2.7.3.1 General interface settings

If an *ibaCom-L2B* card is installed in the *ibaPDA* computer, *ibaPDA* displays the interface of the *ibaCom-L2B* card in the interface tree of the I/O Manager.



Interface settings

Interrupt mode : Slave mode ☑ In use

PCI Info

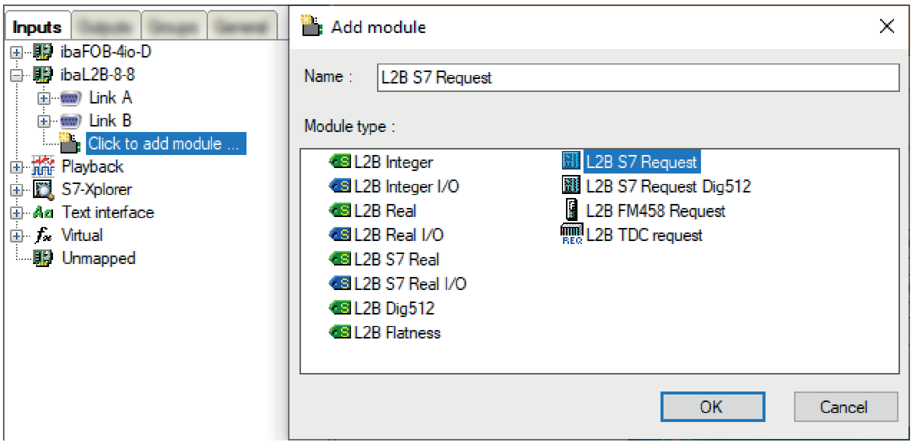
Slot Number:	0	IO Address:	0x00004400	IO Length:	0x00000080
Bus Number:	17	Mem. Address:	0xD3200000	Mem. Length:	0x00100000
Vendor Name:	iba AG			Device Id:	0x4C32

Other documentation

Detailed information about the *ibaCom-L2B* card can be found in the respective manual.

2.7.3.2 General module settings

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



Name : L2B S7 Request

You can find the description of all settings that are identical for all Request-S7 modules in part 1 of the manual.

The L2B modules have the following common specific setting options:

PROFIBUS

Slave number
PROFIBUS slave address assigned to the module.

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

Connection settings

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 I/O peripheral area.

2.7.3.3 Module L2B S7 Request

You can acquire up to 32 analog and 32 digital signals with the *L2B S7 Request* module.

Configure a separate PROFIBUS slave and Request block call for each module.

For a description of the module settings, see chapter ➚ *General module settings*, page 63 and part 1 of this manual.

2.7.3.4 Module L2B S7 Request Dig512

With the *L2B S7 Request Dig512* module, you can acquire up to 512 digital signals 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.

Configure a separate PROFIBUS slave and Request block call for each module.

General tab

For further information on module settings, see ↗ *General module settings*, page 63.

Digital tab

The declaration of the digital signals is done in two steps.

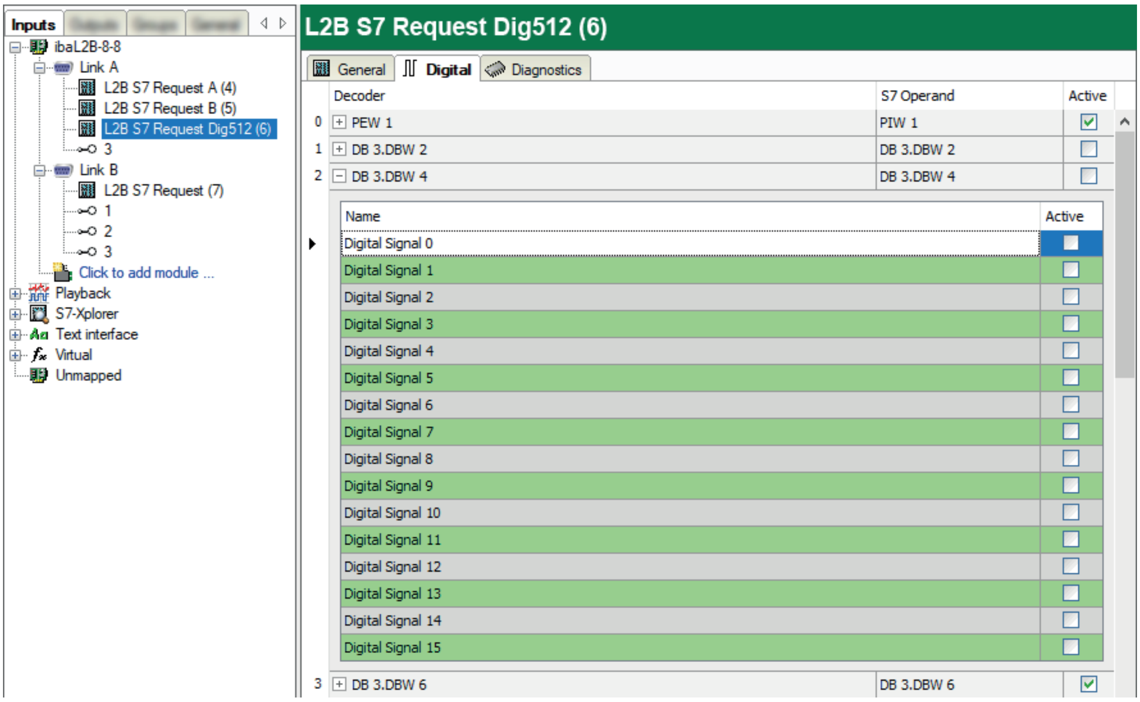
- First define the words (source signals), which are broken down for the digital signals (bits).

  You can directly enter the words as basic signals for decoding via the 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 part 1 of this manual, section *Selection via the symbolic operand addresses*. The signals selected in S7 CFC and symbol browser are applied and the columns *Name*, *S7 symbol* and *S7 Operand* are filled in automatically.

- You can open each word (source signal) via the <+> button to display the list of associated digital signals.

  Then define the individual digital signals (bits) of the source signal.



	Decoder	S7 Operand	Active
0	⊞ PEW 1	PIW 1	☑
1	⊞ DB 3.DBW 2	DB 3.DBW 2	□
2	⊟ DB 3.DBW 4	DB 3.DBW 4	□

Name	Active
Digital Signal 0	□
Digital Signal 1	□
Digital Signal 2	□
Digital Signal 3	□
Digital Signal 4	□
Digital Signal 5	□
Digital Signal 6	□
Digital Signal 7	□
Digital Signal 8	□
Digital Signal 9	□
Digital Signal 10	□
Digital Signal 11	□
Digital Signal 12	□
Digital Signal 13	□
Digital Signal 14	□
Digital Signal 15	□

	Decoder	S7 Operand	Active
3	⊞ DB 3.DBW 6	DB 3.DBW 6	☑

The individual columns of the signal table have the following meanings:

Source signal

Decoder

Enter a name for the source signal.

S7 Operand/S7 Symbol

Enter the S7 operand to which the signal is assigned and, if applicable, the S7 symbol.

Active

If you activate the source signal, it is acquired with all digital signals. You can deactivate individual digital signals.

Individual digital signals (bits)

Name

Enter a name for the individual digital signals.

Active

If you activate the signal, the signal is acquired and is also considered when checking the number of licensed signals.

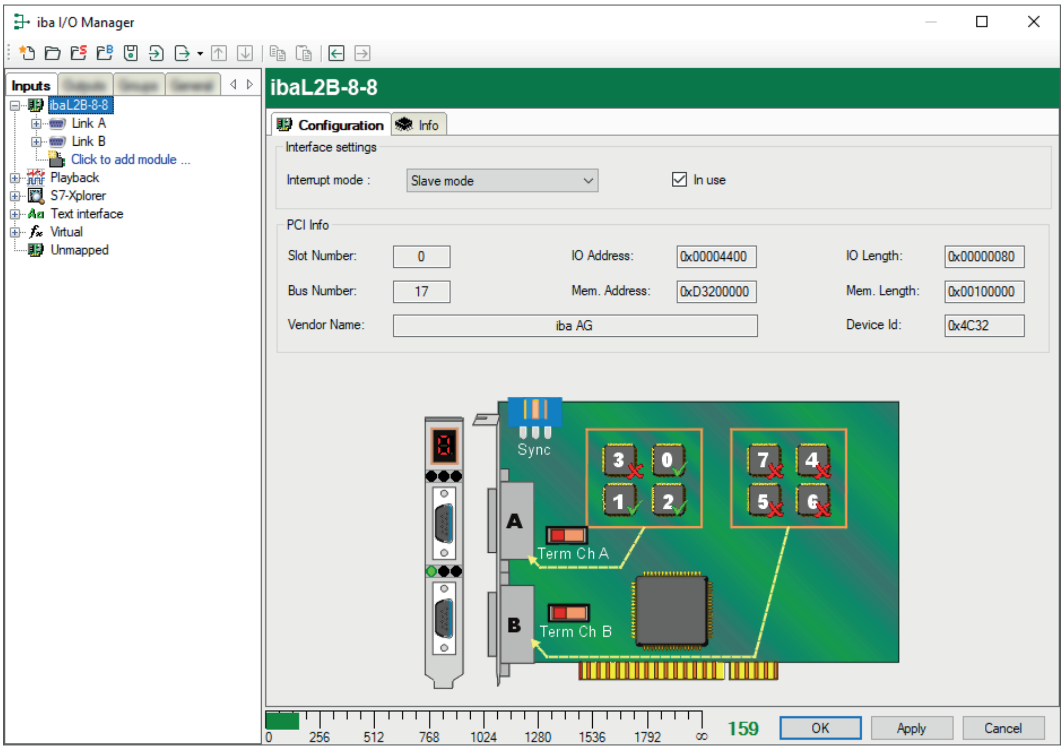
Note

ibaPDA only takes the activated digital signals into account for the number of licensed signals, i.e. no additional signal for the source signal.

2.7.4 Diagnostics

2.7.4.1 ibaCom-L2B card

When you select an L2B card node in the signal tree, *ibaPDA* displays a status overview of the card. The green checkmark on the slave number indicates an active connection. In the figure slaves 0 to 2 are active.



ibaL2B-8-8

Interface settings

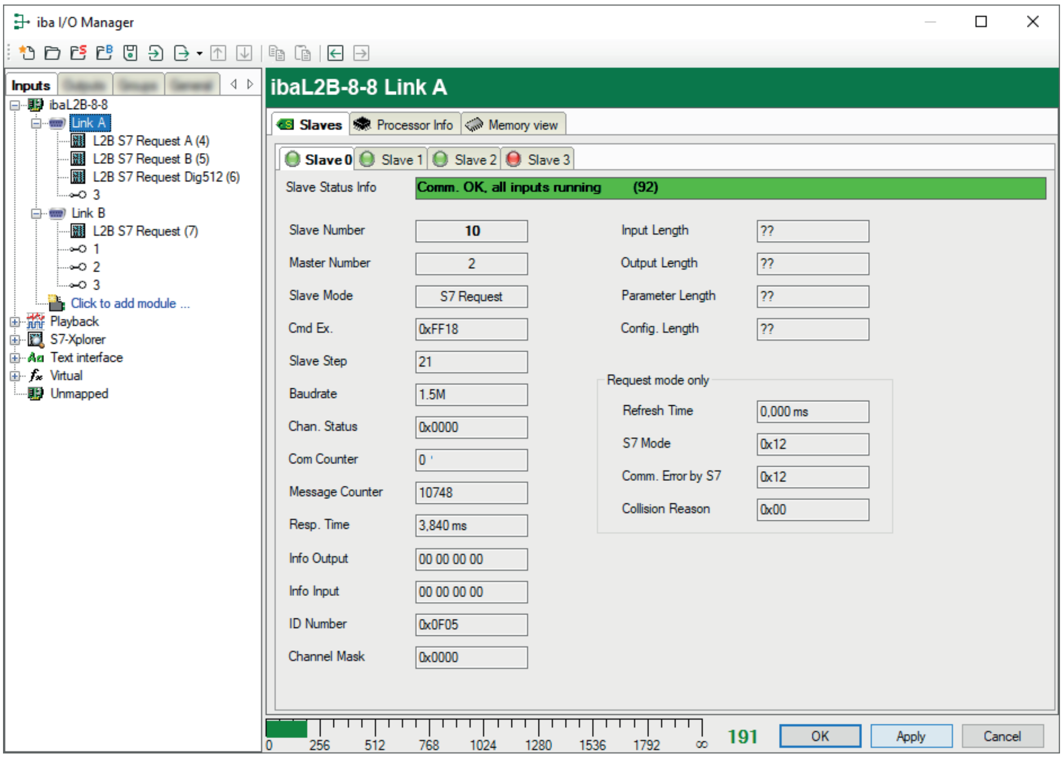
Interrupt mode : Slave mode

☑ In use

PCI Info

Slot Number:	0	IO Address:	0x00004400	IO Length:	0x00000080
Bus Number:	17	Mem. Address:	0xD3200000	Mem. Length:	0x00100000
Vendor Name:	iba AG			Device Id:	0x4C32

When you select the PROFIBUS link, you can see detailed information regarding the status of each slave. This information is mainly relevant for support purposes.



ibaL2B-8-8 Link A

Slave Status Info: Comm. OK, all inputs running (92)

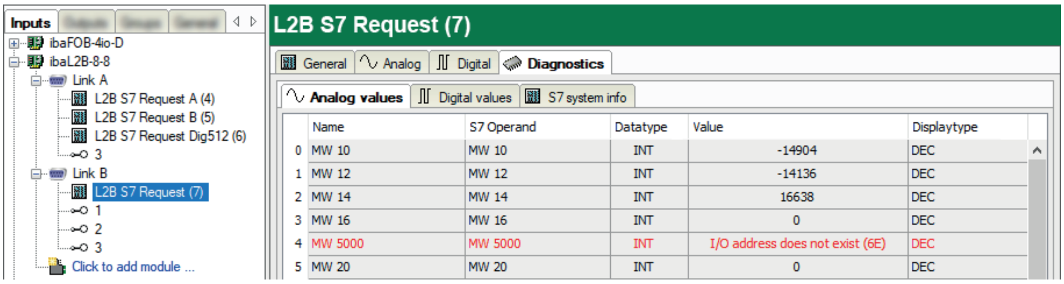
Slave Number	10	Input Length	??
Master Number	2	Output Length	??
Slave Mode	S7 Request	Parameter Length	??
Cmd Ex.	0xFF18	Config. Length	??
Slave Step	21		
Baudrate	1.5M		
Chan. Status	0x0000		
Com Counter	0 '		
Message Counter	10748		
Resp. Time	3,840 ms		
Info Output	00 00 00 00		
Info Input	00 00 00 00		
ID Number	0x0F05		
Channel Mask	0x0000		

Request mode only

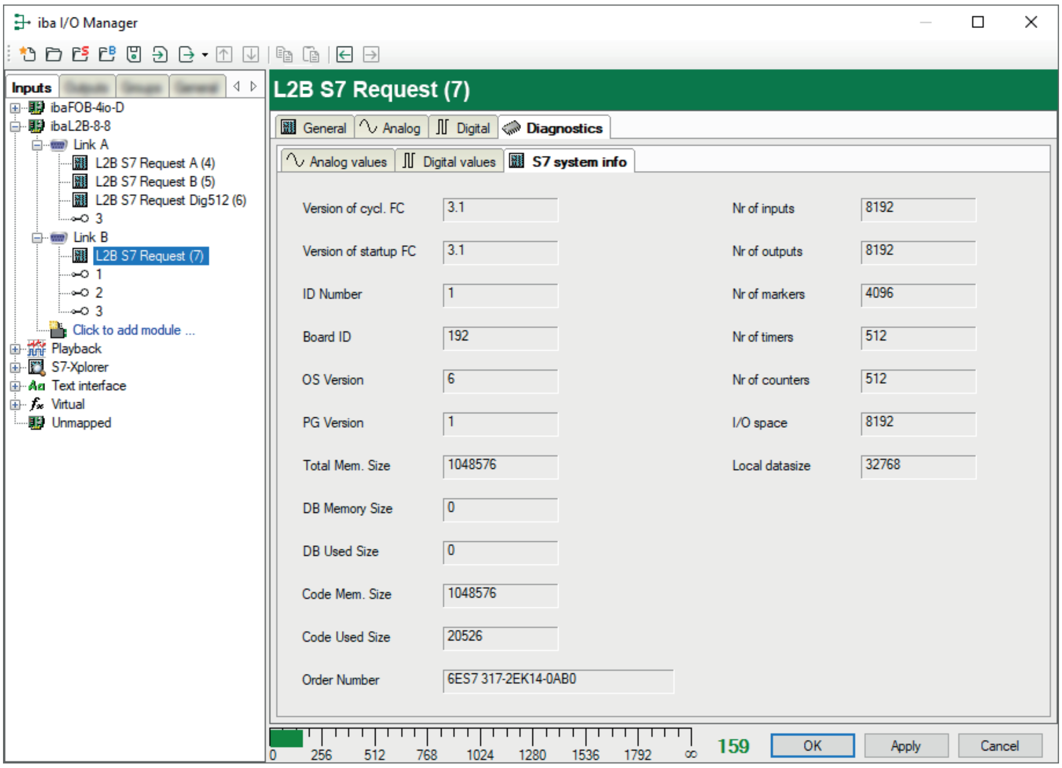
Refresh Time	0,000 ms
S7 Mode	0x12
Comm. Error by S7	0x12
Collision Reason	0x00

2.7.4.2 Module diagnostics

When you select an L2B-Request module node in the signal tree, in the *Diagnostics* tab you can see details of the individual measured values and system information of the S7-CPU in the sub-tab *S7 system info*.



	Name	S7 Operand	Datatype	Value	Displaytype
0	MW 10	MW 10	INT	-14904	DEC
1	MW 12	MW 12	INT	-14136	DEC
2	MW 14	MW 14	INT	16638	DEC
3	MW 16	MW 16	INT	0	DEC
4	MW 5000	MW 5000	INT	I/O address does not exist (6E)	DEC
5	MW 20	MW 20	INT	0	DEC



Version of cycl. FC	3.1	Nr of inputs	8192
Version of startup FC	3.1	Nr of outputs	8192
ID Number	1	Nr of markers	4096
Board ID	192	Nr of timers	512
OS Version	6	Nr of counters	512
PG Version	1	I/O space	8192
Total Mem. Size	1048576	Local datasize	32768
DB Memory Size	0		
DB Used Size	0		
Code Mem. Size	1048576		
Code Used Size	20526		
Order Number	6ES7 317-2EK14-0AB0		

3 Description of Request blocks

3.1 ibaREQ iba block family

These blocks initialize and control communication between *ibaPDA* and the S7 controller.

The ibaREQ iba block family allows access exclusively to non-optimized data blocks. Addressing is done by using the operand address.

One set of Request blocks has to be called for each Request module (connection) in *ibaPDA*. The blocks used are part of the iba S7 library, see ↗ *iba S7 library*, page 91.

Note

Only use Request blocks from the latest iba S7 library!

Request blocks in application examples can be outdated and, thus, cause errors.

Use different Request block combinations depending on the current system configuration:

Request block	S7-300 with integrated DP interface	S7-400(H) with integrated DP interface and CP443-5	S7-300 with integrated PN interface	S7-400 with integrated PN interface and CP443-1 (PROFINET)	Recommended call level
ibaREQ_M (FB140)	X	X	X	X	OB1
ibaREQ_PN (FB141)	X	X	X	X	OB1 or OB3x
ibaREQ_DP (FB142)	X	X	X	X	OB1 or OB3x
ibaREQ_DB (DB15)	X	X	X	X	-
ibaREQ_PNdev (FB150)	-	-	X	X	OB1 or OB3x
ibaREQ_DB-Interface	X	X	X	X	-

- ibaREQ_M (Management)
  The block realizes the communication with *ibaPDA*. Ideally, the block is called in OB1.
- ibaREQ_PN/ibaREQ_DP (provision and sending of current signal values)
  The block provides the current signal values using the transmission cycle. Ideally, the block is called in a cyclic interrupt OB.
- ibaREQ-PNdev (diagnostics for PROFINET device)
  The block performs cyclic diagnostics for availability of the PROFINET device used.
- ibaREQ_DB (interface DB)
  This DB acts as an interface to *ibaPDA* and between the different Request blocks.

3.1.1 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: No reset (default) TRUE: Block reset
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 related Request blocks, configure the identical DB.

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 are evaluated according to the length.

RESET
Used to manually reset the Request blocks. All Request blocks of a combination are automatically reset together. Usually, the parameter 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 is output.

For a list of all possible error codes, see chapter ↗ *Error codes of Request blocks*, page 110.

3.1.2 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 related Request blocks, configure the identical DB.

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 is output.

For a list of all possible error codes, see chapter ↗ *Error codes of Request blocks*, page 110.

3.1.3 ibaREQ_DP (FB142)

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	FALSE: No reset (default) 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.

RESET_CON
Used for manually resetting the communication connection.

REQ_DB
The DB is used for data exchange with *ibaPDA*. For all related Request blocks, configure the identical DB.

ERROR_STATUS
Internal error of the block. If there is no error, the value 0 is output.

For a list of all possible error codes, see chapter ↗ *Error codes of Request blocks*, page 110.

3.1.4 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 is output.

For a list of all possible error codes, see chapter ↗ *Error codes of Request blocks*, page 110.

3.2 Description of Request FCs

These functions initialize and control the communication between the S7 controller (SIMATIC S7-300, S7-400, WinAC only), *ibaPDA*, and the *ibaBM-DP* device or *ibaCom-L2B* board.

The blocks used are part of the iba S7 library, see ↗ *iba S7 library*, page 91.

Note

Only use Request blocks from the latest iba S7 library!

Request blocks in application examples can be outdated and, thus, cause errors.

3.2.1 Request-FC ibaDP_Req (FC122)

This function initializes and controls the communication between the S7 controller, *ibaPDA* and the *ibaBM-DP* device.

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 ↗ *iba S7 library*, page 91.

Note

Only use Request blocks from the latest iba S7 library!

Request blocks in application examples can be outdated and, thus, cause errors.

Description of the formal parameters of the ibaDP_Req (FC122)

Name	Type	Data Type	Description
DB_PDA	IN	BLOCK_DB	DB of the *ibaPDA* 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 handles the 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
- S7 pointers requested by *ibaPDA*
- 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 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_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.

3.2.2 Request-FC ibaDP_Req_H (FC123)

This function initializes and controls the communication between the S7 controller, *ibaPDA* and the *ibaBM-DP* device.

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 ↗ *iba S7 library*, page 91.

Note

Only use Request blocks from the latest iba S7 library!

Request blocks in application examples can be outdated and, thus, cause errors.

Description of the formal parameters of the ibaDP_Req_H

Name	Type	Data Type	Description
DB_PDA	IN	BLOCK_DB	DB of the *ibaPDA* 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
- S7 pointers requested by *ibaPDA*
- 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 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_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: RM[2]: 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: RM[3]: 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.

[2] RM: Redundancy mode
[3] RM: Redundancy mode

3.2.3 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 ibaL2B_DB_work 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); generally the setting "0" (automatic detection) is recommended. Only if the wrong number is displayed in the "S7 System Info" (*ibaPDA*), 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); generally the setting "0" (automatic detection) is recommended. Only if the wrong number is displayed in the "S7 System Info" (*ibaPDA*), 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); to define "0" (automatic detection) is generally recommended. Only if the wrong number is displayed in the "S7 System Info" (*ibaPDA*), the number of marker bytes can be set manually! "0": automatic detection (recommended).

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

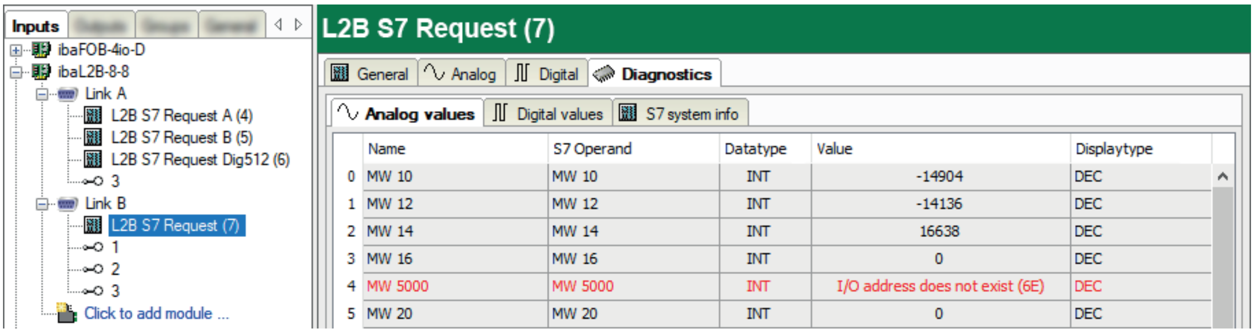
The following SFCs are used internally:

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

Ranges for number of inputs, outputs and markers

The ibaL2B_Init function can reliably evaluate the number of inputs, outputs and markers in the CPU. 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 or modified after consultation with the iba support. These parameters help to avoid configuration errors when selecting signals in *ibaPDA*. If, for example, you select a marker that 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.



	Name	S7 Operand	Datatype	Value	Displaytype
0	MW 10	MW 10	INT	-14904	DEC
1	MW 12	MW 12	INT	-14136	DEC
2	MW 14	MW 14	INT	16638	DEC
3	MW 16	MW 16	INT	0	DEC
4	MW 5000	MW 5000	INT	I/O address does not exist (6E)	DEC
5	MW 20	MW 20	INT	0	DEC

In such case, 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.

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

3.2.5 Communication-FC ibaL2B_Req (FC112)

This function provides the communication between the S7 and the *ibaCom-L2B* card within the *ibaPDA* computer. 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	Parameterization error The following error codes can be displayed: "92" – Invalid command from *ibaPDA* (→ contact iba support) "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) "101" – channel no. > 31 (→ contact iba support) "102" – data type is not supported (analog values only) (→ contact iba support) "103" – range identifier missing or cannot be interpreted (→ contact iba support) "104 – this variable type cannot be interpreted. → contact iba support "105" – DB 0 was requested as data source. (→ use other allowed DB) "106" – DB number is higher than max. permissible number of DBs of this CPU. (→ change to allowed DB) "107" – DB with the specified number does not exist. (→ request existing DB data) "109" – DB is too short for the accessed operand (→ copy/rename DB22) "110" – actual operand address (I, O, M, P) does not exist. (→ request available operands) "111" – Initialization aborted with error (→ analyze error state of ibaL2B_Init, otherwise contact iba support) "112" – Initialization not performed (→ call ibaL2B_Init in the start-up OBs)

The following SFCs are used internally:

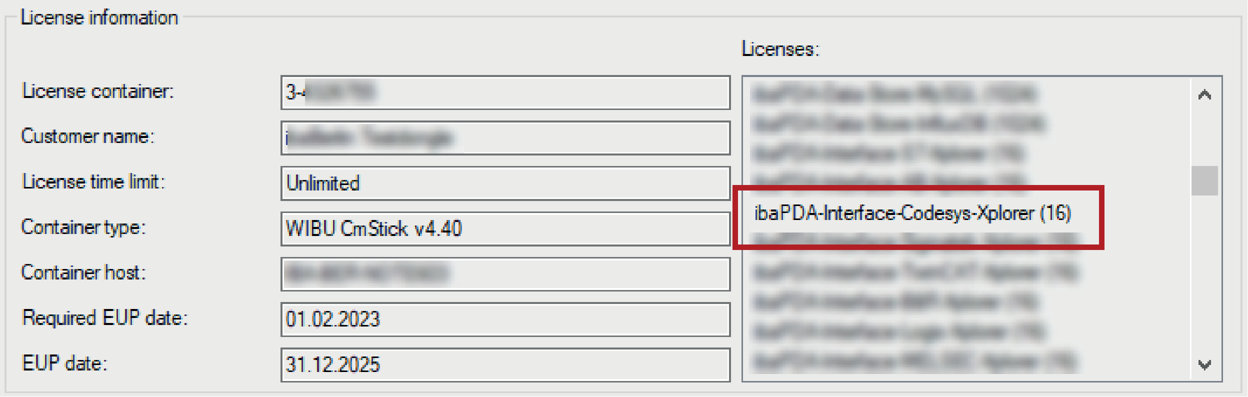
- SFC 20 (BLKMOV)
- SFC 24 (TEST_DB)
- SFC 36 (MSK_FLT)
- SFC 37 (DMSK_FLT)
- SFC 50 (RD_LGADR)

4 Diagnostics

4.1 License

If the interface is not displayed in the signal tree, you can either check in *ibaPDA* in the I/O Manager under *General – Settings* or in the *ibaPDA* service status application whether your license for this interface has been properly recognized. The number of licensed connections is shown in brackets.

The figure below shows the license for the *Codesys Xplorer* interface as an example.



License information

License container:	3-[redacted]
Customer name:	[redacted]
License time limit:	Unlimited
Container type:	WIBU CmStick v4.40
Container host:	[redacted]
Required EUP date:	01.02.2023
EUP date:	31.12.2025

Licenses:

[redacted]

ibaPDA-Interface-Codesys-Xplorer (16)

[redacted]

4.2 Visibility of the interface

If the interface is not visible despite a valid license, it may be hidden.

Check the settings in the *General* tab in the *Interfaces* node.

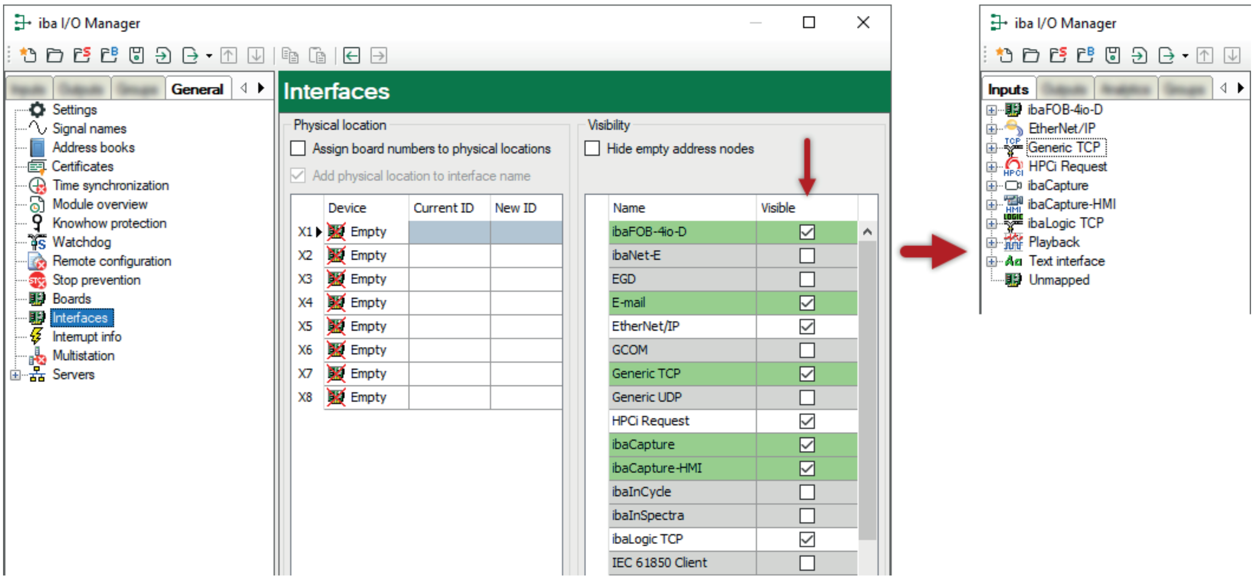
Visibility

The table *Visibility* lists all the interfaces that are available either through licenses or installed cards. These interfaces can also be viewed in the interface tree.

You can hide or display the interfaces not required in the interface tree by using the checkbox in the *Visible* column.

Interfaces with configured modules are highlighted in green and cannot be hidden.

Selected interfaces are visible, the others are hidden:



Physical location

☐ Assign board numbers to physical locations

☑ Add physical location to interface name

Visibility

☐ Hide empty address nodes

	Device	Current ID	New ID
X1	Empty		
X2	Empty		
X3	Empty		
X4	Empty		
X5	Empty		
X6	Empty		
X7	Empty		
X8	Empty		

Name	Visible
ibaFOB-4io-D	☑
ibaNet-E	☐
EGD	☐
E-mail	☑
EtherNet/IP	☑
GCOM	☐
Generic TCP	☑
Generic UDP	☐
HPCi Request	☑
ibaCapture	☑
ibaCapture-HMI	☑
ibaInCycle	☐
ibaInSpectra	☐
ibaLogic TCP	☑
IEC 61850 Client	☐

4.3 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.

You can open the log file 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 can find the log files of the *ibaPDA* server (…\ProgramData\iba\ibaPDA\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.

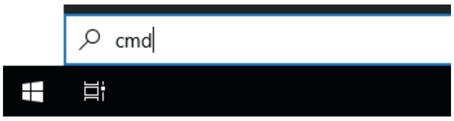
Examples:

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

4.4 Connection diagnostics with PING

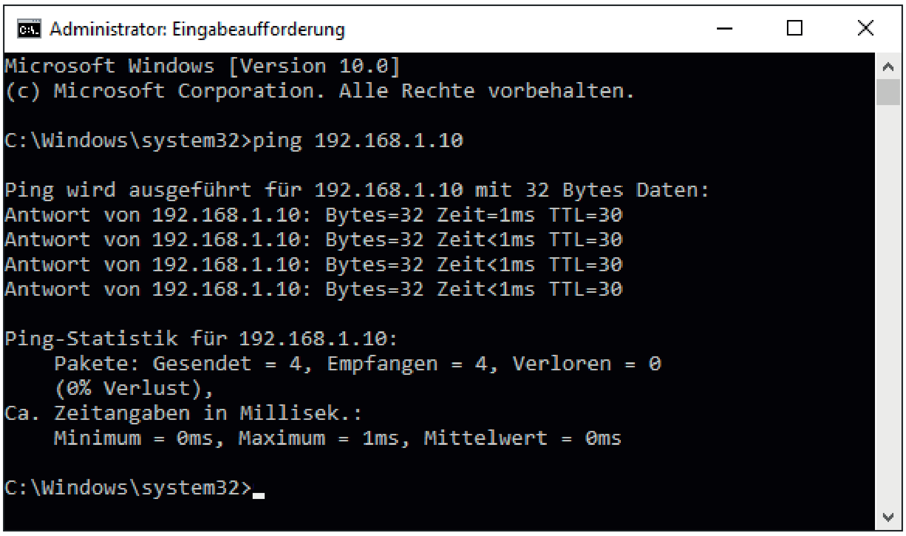
PING is a system command with which you can check if a certain communication partner can be reached in an IP network.

1. Open a Windows command prompt.



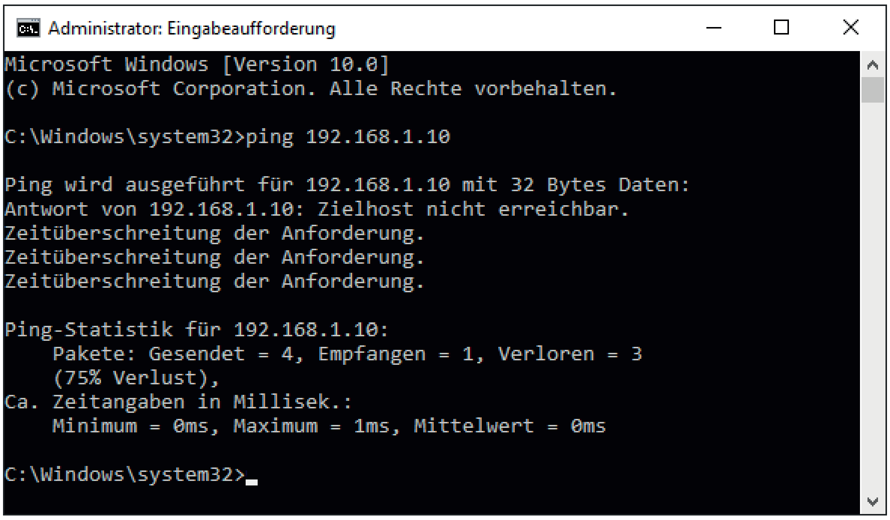
2. Enter the command "ping" followed by the IP address of the communication partner and press <ENTER>.

→ With an existing connection you receive several replies.



```
Microsoft Windows [Version 10.0]
(c) Microsoft Corporation. Alle Rechte vorbehalten.

C:\Windows\system32>ping 192.168.1.10

Ping wird ausgeführt für 192.168.1.10 mit 32 Bytes Daten:
Antwort von 192.168.1.10: Bytes=32 Zeit=1ms TTL=30
Antwort von 192.168.1.10: Bytes=32 Zeit<1ms TTL=30
Antwort von 192.168.1.10: Bytes=32 Zeit<1ms TTL=30
Antwort von 192.168.1.10: Bytes=32 Zeit<1ms TTL=30

Ping-Statistik für 192.168.1.10:
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0
    (0% Verlust),
Ca. Zeitangaben in Millisek.:
    Minimum = 0ms, Maximum = 1ms, Mittelwert = 0ms

C:\Windows\system32>
```

→ With no existing connection you receive error messages.

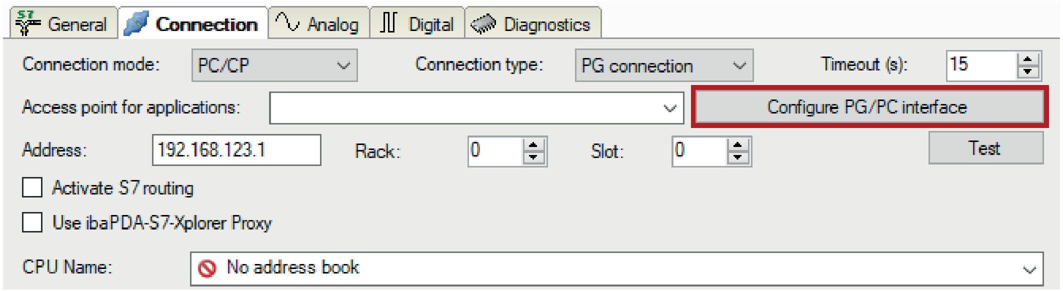


```
Microsoft Windows [Version 10.0]
(c) Microsoft Corporation. Alle Rechte vorbehalten.

C:\Windows\system32>ping 192.168.1.10

Ping wird ausgeführt für 192.168.1.10 mit 32 Bytes Daten:
Antwort von 192.168.1.10: Zielhost nicht erreichbar.
Zeitüberschreitung der Anforderung.
Zeitüberschreitung der Anforderung.
Zeitüberschreitung der Anforderung.

Ping-Statistik für 192.168.1.10:
    Pakete: Gesendet = 4, Empfangen = 1, Verloren = 3
    (75% Verlust),
Ca. Zeitangaben in Millisek.:
    Minimum = 0ms, Maximum = 1ms, Mittelwert = 0ms

C:\Windows\system32>
```

4.5 Connection diagnostics with PG/PC interface

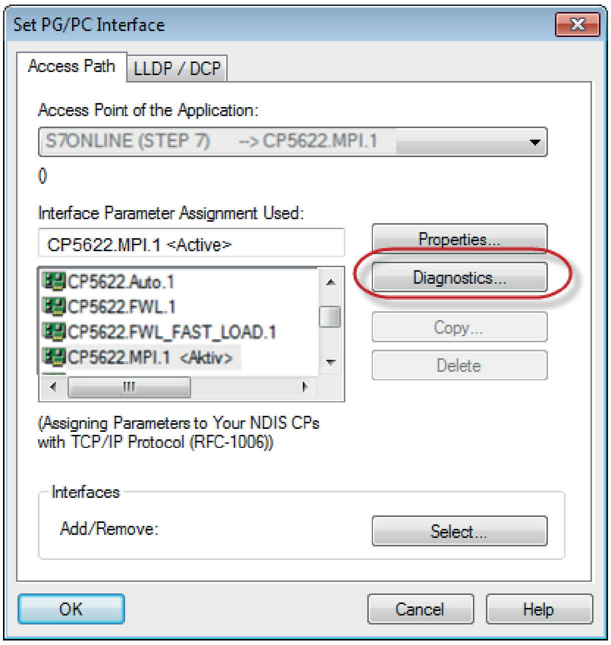
Use the diagnostic function of the PG/PC interface to the functionality and connection configuration.

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



Connection mode: PC/CP
Connection type: PG connection
Timeout (s): 15
Access point for applications:
Address: 192.168.123.1
Rack: 0
Slot: 0
□ Activate S7 routing
□ Use ibaPDA-S7-Xplorer Proxy
CPU Name: No address book

2. Open the diagnostics dialog with the <Diagnostics> button.



Set PG/PC Interface

Access Point of the Application:
S7ONLINE (STEP 7) --> CP5622.MPI.1
()

Interface Parameter Assignment Used:
CP5622.MPI.1 <Active>

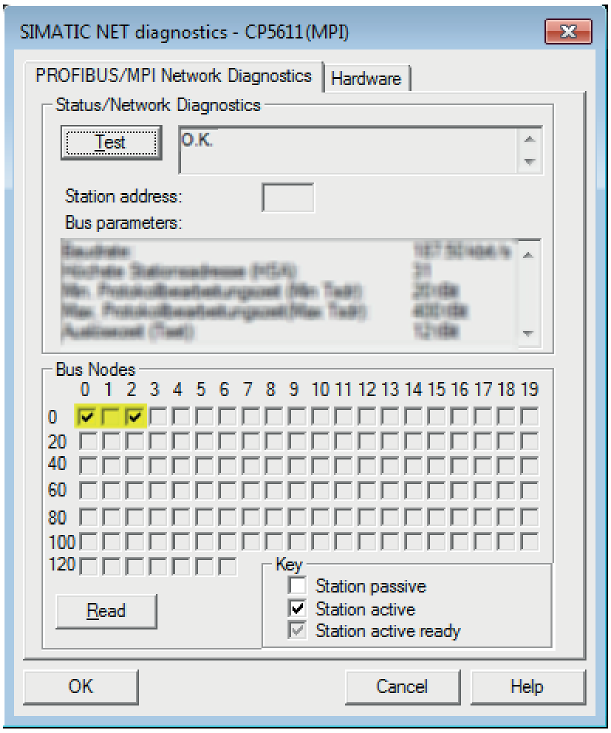
CP5622.Auto.1
CP5622.FWL.1
CP5622.FWL_FAST_LOAD.1
CP5622.MPI.1 <Aktiv>

(Assigning Parameters to Your NDIS CPs with TCP/IP Protocol (RFC-1006))

Interfaces
Add/Remove:

3. Start the network diagnostics with the <Test> button.
Check the availability of the bus devices with the <Read> button.

→ The following figure shows an example of a diagnostics of a SIMATIC Net CP5622 (PROFIBUS).
One active station is on the address 0 and 2 each.


SIMATIC NET diagnostics - CP5611(MPI)
PROFIBUS/MPI Network Diagnostics
Hardware
Status/Network Diagnostics
Test
Bus Nodes
Read
OK
Cancel
Help

Status: O.K.

Station address:

Bus parameters:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	☑	□	☑	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
20	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
40	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
60	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
80	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
100	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
120	□	□	□	□	□	□	□													

Key
□ Station passive
☑ Station active
☑ Station active ready

4.6 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 19

See chapter Request-S7 for ibaBM-DP ↗ *Diagnostics*, page 36

See chapter Request-S7 for ibaBM-DPM-S ↗ *Diagnostics*, page 50

See chapter Request-S7 for ibaCom-L2B ↗ *Diagnostics*, page 66

5 Appendix

5.1 iba S7 library

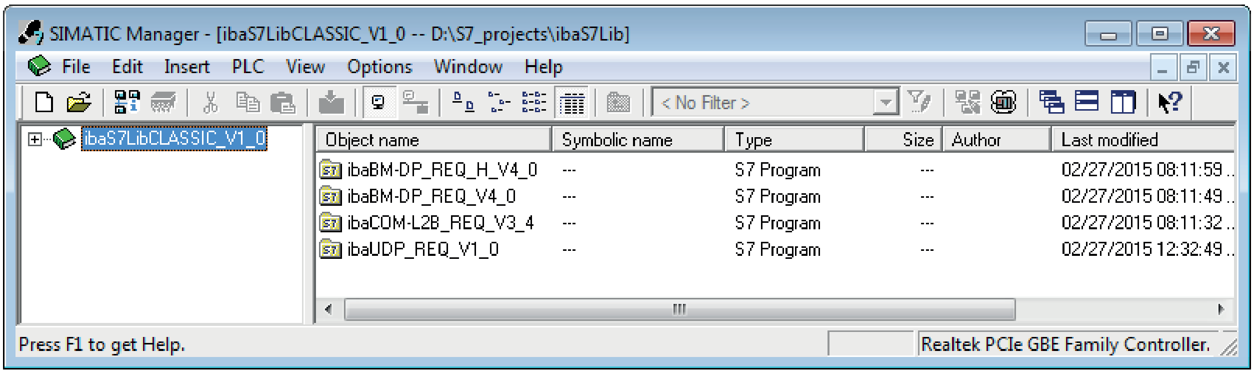
The iba S7 library is available in two versions:

- SIMATIC Manager : STEP 7 ≥ V5.5
- SIMATIC TIA Portal STEP 7 ≥ V16

5.1.1 iba S7 library for SIMATIC Manager

The iba S7 library for SIMATIC Manager ("ibaS7LibCLASSIC_Vx_y") is suitable for 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/ibaNet-E*.

You find the iba S7 library as an archived file on the data storage medium "iba Software & Manuals" in the following directory:
`\04_Libraries_and_Examples\10_Libraries\01_SIMATIC_S7\`


SIMATIC Manager - [ibaS7LibCLASSIC_V1_0 -- D:\S7_projects\ibaS7Lib]

Object name	Symbolic name	Type	Size	Author	Last modified
ibaBM-DP_REQ_H_V4_0	---	S7 Program	---		02/27/2015 08:11:59 ..
ibaBM-DP_REQ_V4_0	---	S7 Program	---		02/27/2015 08:11:49 ..
ibaCOM-L2B_REQ_V3_4	---	S7 Program	---		02/27/2015 08:11:32 ..
ibaUDP_REQ_V1_0	---	S7 Program	---		02/27/2015 12:32:49 ..

Press F1 to get Help.

Realtek PCIe GBE Family Controller.

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	

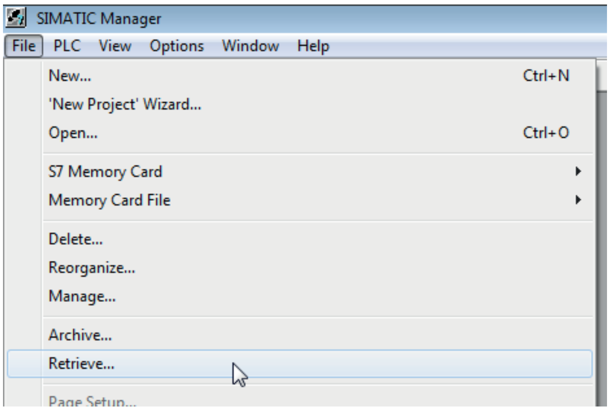
iba connection	Block name	Block no.	Note
ibaCom-L2B	ibaL2B_Init	FC111	formerly FC23 and FC101
	ibaL2B_Req	FC112	formerly FC22 and FC100
	ibaL2B_Req_CP	FC113	formerly FC26 and FC102 only necessary when using a CP342-5 instead of the FC112
	ibaL2B_DB_work	DB22	
	ibaL2B_DB_Struct	UDT22	
	ibaL2B_CP_SNDRCV	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 1: ibaS7LibCLASSIC block overview

5.1.1.1 Integrating the library into SIMATIC Manager

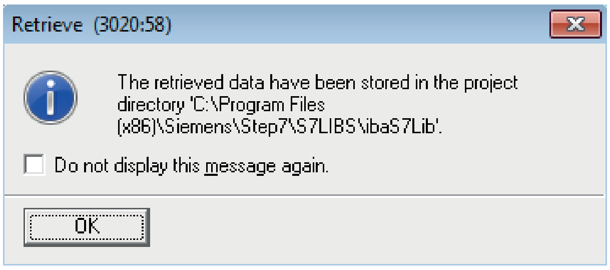
For integrating the library, it has to be retrieved in the SIMATIC Manager. Copy the iba S7 library to a local directory of your computer, on which the SIMATIC Manager is executed.

1. Select the menu *File – Retrieve*.



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

3. Confirm the message for a successful extraction.

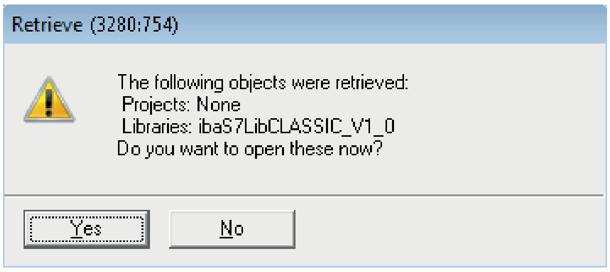


Retrieve (3020:58)

The retrieved data have been stored in the project directory 'C:\Program Files (x86)\Siemens\Step7\S7LIBS\ibaS7Lib'.

☐ Do not display this message again.

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



Retrieve (3280:754)

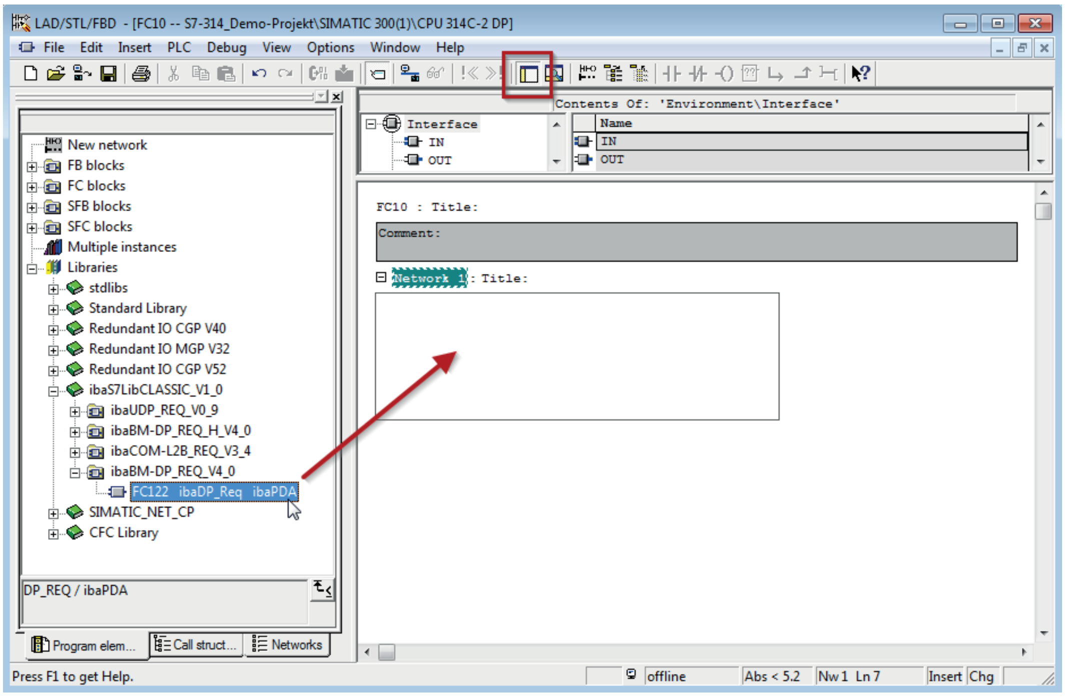
The following objects were retrieved:
Projects: None
Libraries: ibaS7LibCLASSIC_V1_0
Do you want to open these now?

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

5.1.1.2 Using the blocks in SIMATIC Manager

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

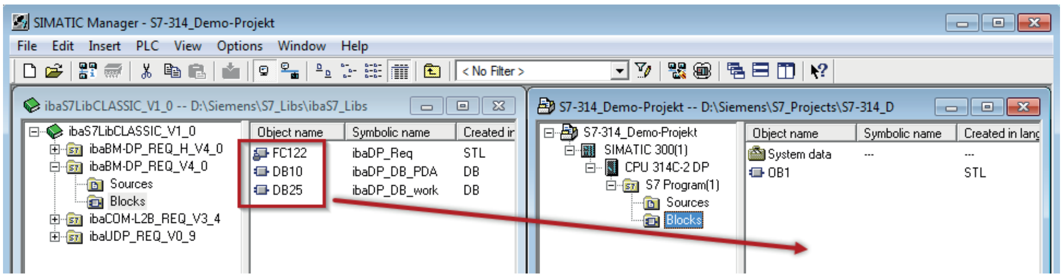
- Display the block library and drag the required blocks to the opened destination block.



- 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.

  You can either drag & drop the blocks or copy & paste them.



Object name	Symbolic name	Created ir
FC122	ibaDP_Req	STL
DB10	ibaDP_DB_PDA	DB
DB25	ibaDP_DB_work	DB

Object name	Symbolic name	Created in lang
System data	---	---
OB1		STL

→ You can now use the blocks in the destination project.

5.2 Application examples

You can find application examples for several different configurations on the data storage medium "iba Software & Manuals".

- For the ibaBM-DPM-S device in the directory
  `\04_Libraries_and_Examples\30_ibaBM-DPM-S\01_SIMATIC_S7\Request-S7\`
- For the ibaBM-DP device in the directory
  `\04_Libraries_and_Examples\31_ibaBM-DP\01_SIMATIC_S7\Request-S7\`
- For the ibaBM-PN device in the directory
  `\04_Libraries_and_Examples\32_ibaBM-PN\01_SIMATIC_S7\Request-S7\`
- For the ibaCom-L2B card in the directory
  `\04_Libraries_and_Examples\40_ibaCOM-L2B\01_SIMATIC_S7\Request-S7\`
- For the interface ibaPDA-Interface-ibaNet-E
  `\04_Libraries_and_Examples\53_ibaPDA-Interface-ibaNet-E\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
		ibaS7TIA_PN_REQsym_Vxx.zap*	ibaPDA_S7TIA_PN_REQsym
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
ibaPDA-Interface-ibaNet-E	S7-1500	ibaS7TIA_NetE_REQ_Vxx.zap*	ibaPDA_S7TIA_NetE_REQ
		ibaS7TIA_NetE_REQsym_Vxx.zap*	ibaPDA_S7TIA_NetE_REQsym

Table 2: Configuration examples available on data storage medium

5.3 S7 cycle time measurements

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

S7-CPUs with integrated DP interface

S7-CPU type	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

S7-CPUs 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 because the data transmission over the backplane bus of the S7-300 is too slow. Otherwise, precise cycle measurements are possible if the PROFIBUS cycle is sufficiently fast.

5.3.2 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

5.3.3 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

5.4 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).

Reference

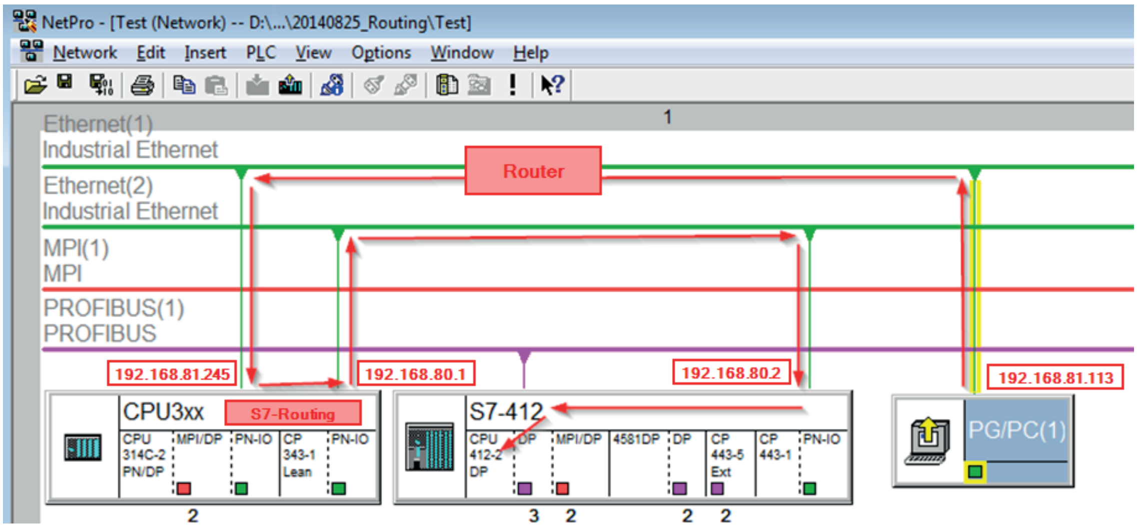
For more information about the S7 Routing, refer to:

- Which modules support the "S7 Routing" function in S7 subnets?
  https://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/en/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

5.4.1 Routing from Ethernet to Ethernet

Do not mix up the *S7 Routing* function with IP routing.

The example shows how to implement the following way of access via S7 Routing in NetPro.



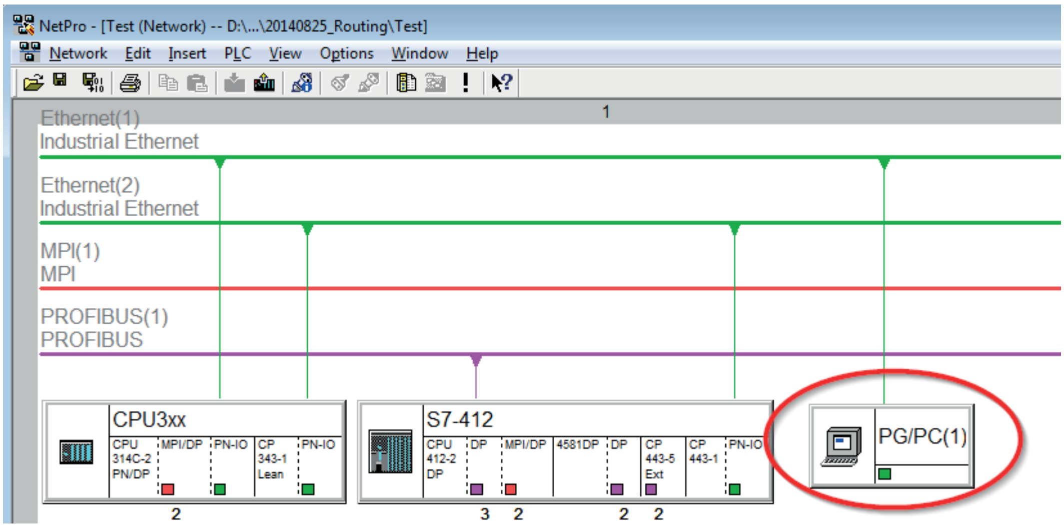
The engineering computer (also with *ibaPDA*) is to access the CPU412 controller. The computer and the controller are not directly connected via a common network/bus. The connection has to run over the CPU314C controller.
"Passing" the communication through this controller is called *S7 Routing*.

In our example, engineering computer 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 mixed up with it.

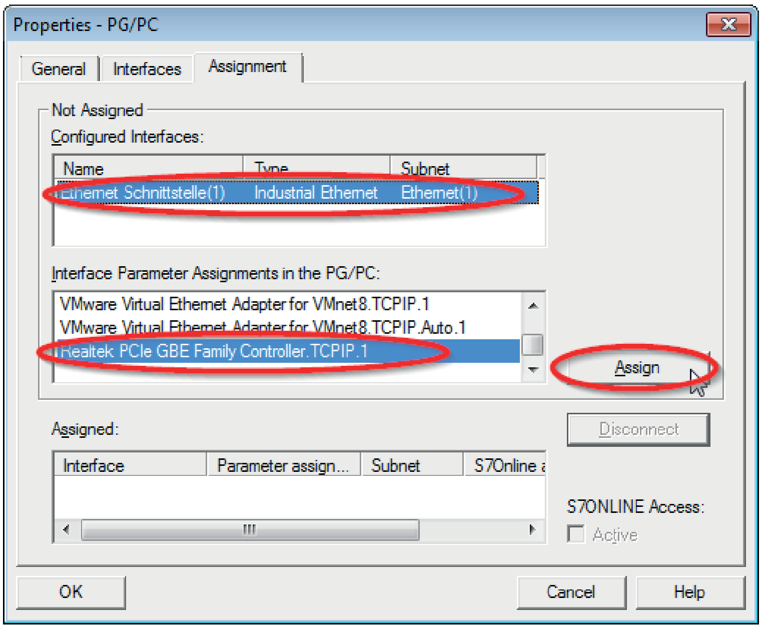
5.4.1.1 Configuration of STEP 7/NetPro

The following configuration steps are required to be able to access the secondary CPU412 control with the SIMATIC STEP 7 programming software. For SINUMERIK, SINAMICS, or SIMOTION, you can apply similar steps. For using *ibaPDA*, these configuration steps are not required.

1. Add a PG/PC station and configure it.



2. Assign an interface (network card).


Properties - PG/PC

General | Interfaces | Assignment

Not Assigned

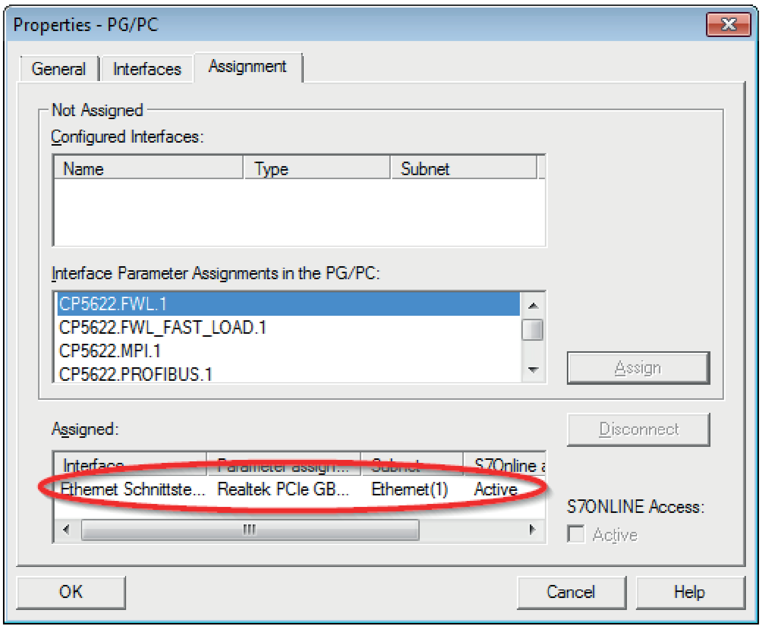
Configured Interfaces:

Name	Type	Subnet
Ethernet Schnittstelle(1)	Industrial Ethernet	Ethernet(1)

Interface Parameter Assignments in the PG/PC:

VMware Virtual Ethernet Adapter for VMnet8.TCPIP.1
VMware Virtual Ethernet Adapter for VMnet8.TCPIP.Auto.1
Realtek PCIe GBE Family Controller.TCPIP.1

→ Result:



Properties - PG/PC

General | Interfaces | Assignment

Not Assigned

Configured Interfaces:

Name	Type	Subnet

Interface Parameter Assignments in the PG/PC:

CP5622.FWL.1
CP5622.FWL_FAST_LOAD.1
CP5622.MPI.1
CP5622.PROFIBUS.1

Assign

Assigned:

Disconnect

Interface	Parameter assign...	Subnet	S7Online a
Ethernet Schnittste...	Realtek PCIe GB...	Ethernet(1)	Active

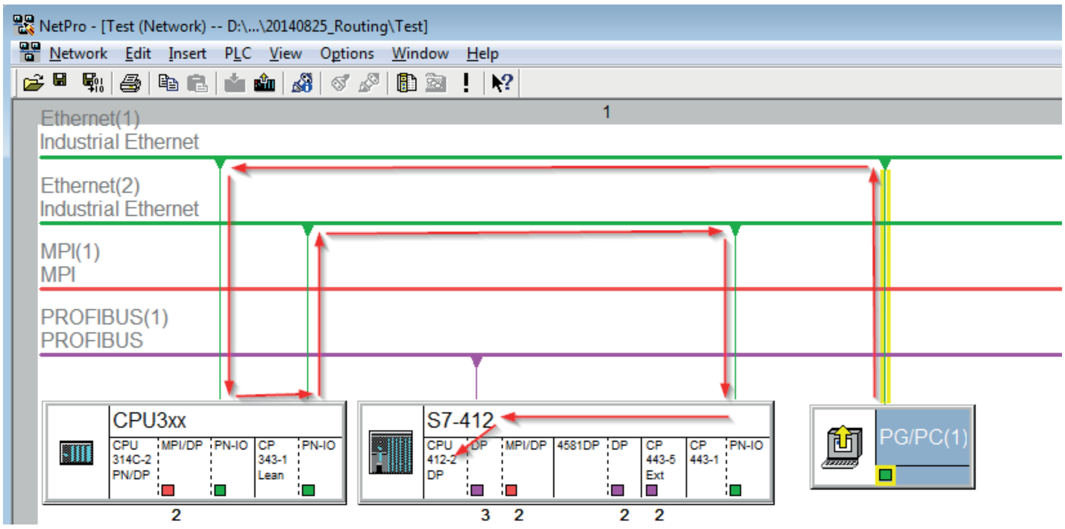
S7ONLINE Access:

□ Active

OK Cancel Help

Now, the connection line from PG/PC to the network has to be marked in yellow.

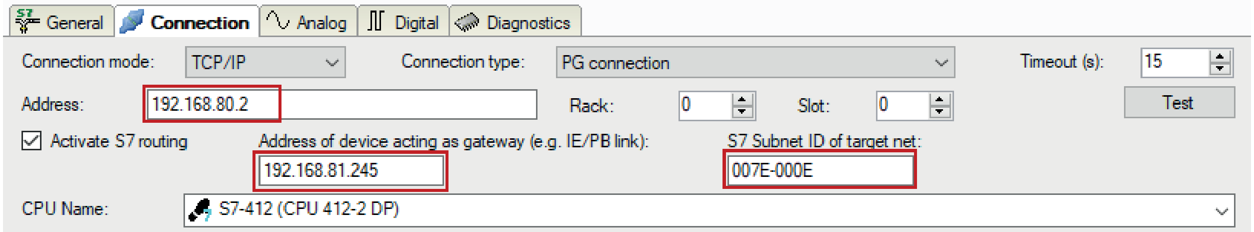
In the following figure, the communication path is shown using arrows (these are not displayed in SIMATIC NetPro).



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

5.4.1.2 Configuration of ibaPDA

Configure the following settings.



Connection mode: TCP/IP
Connection type: PG connection
Timeout (s): 15
Address: 192.168.80.2
Rack: 0
Slot: 0
☑ Activate S7 routing
Address of device acting as gateway (e.g. IE/PB link): 192.168.81.245
S7 Subnet ID of target net: 007E-000E
CPU Name: S7-412 (CPU 412-2 DP)

Activate S7 routing
Enable this option to use S7 routing.

Address
Enter the address of the target control (here CPU412)

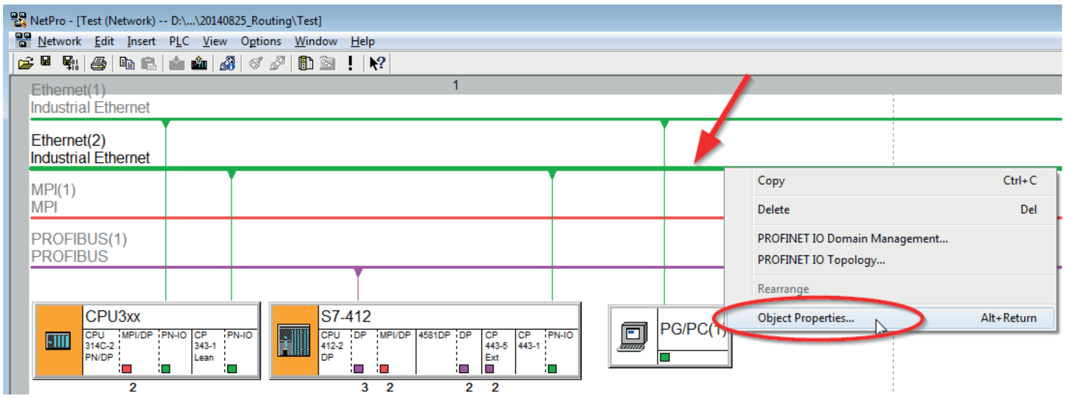
Address of device acting as gateway
Enter the address of the gateway (here CPU314C).

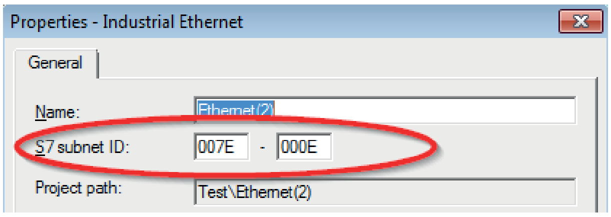
S7 subnet ID of target net
Enter the subnet ID from STEP 7 NetPro or TIA Portal.

Identifying the S7 subnet ID in NetPro

You can identify the S7 subnet ID in NetPro.

Right-click on the secondary bus system and open the *Object Properties*.


NetPro - [Test (Network) -- D:\...\20140825_Routing\Test]
Network Edit Insert PLC View Options Window Help
1
Ethernet(1)
Industrial Ethernet
Ethernet(2)
Industrial Ethernet
MPI(1)
MPI
PROFIBUS(1)
PROFIBUS
CPU3xx
S7-412
PG/PC(1)
Copy Ctrl+C
Delete Del
PROFINET IO Domain Management...
PROFINET IO Topology...
Rearrange
Object Properties... Alt+Return

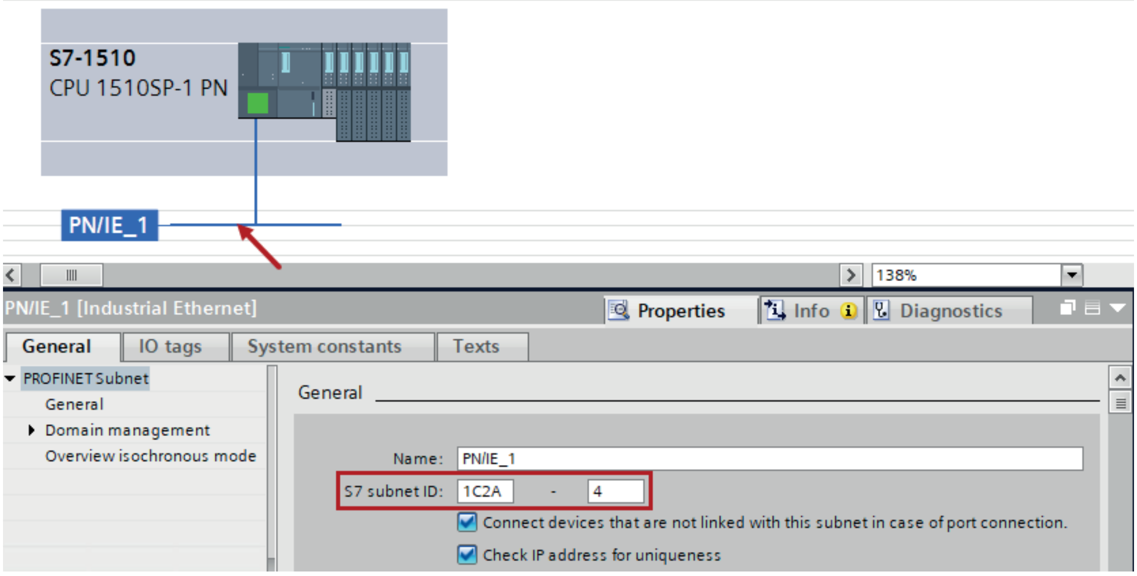


Name: Ethernet(2)
S7 subnet ID: 007E - 000E
Project path: Test\Ethernet(2)

Identifying the S7 subnet ID in TIA Portal

You can identify the S7 subnet ID in TIA Portal.

Click on the bus system and go to *Properties – General – General*.



Name: PN/IE_1

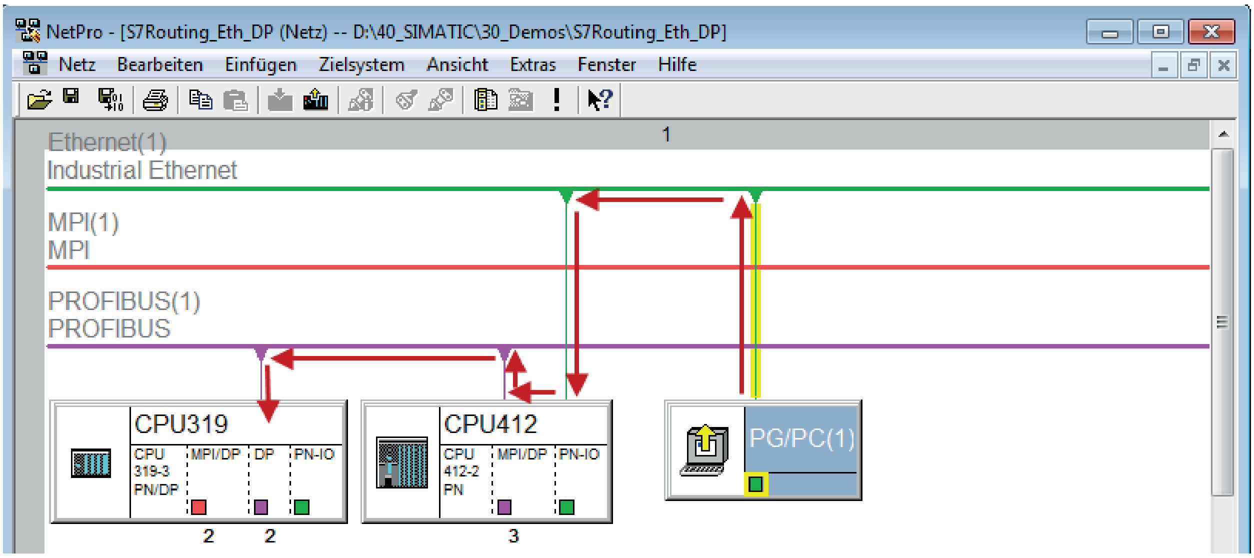
S7 subnet ID: 1C2A - 4

☑ Connect devices that are not linked with this subnet in case of port connection.

☑ Check IP address for uniqueness

5.4.2 Routing from Ethernet to PROFIBUS

The example shows how to implement the following way of access via S7 Routing and an example system topology for Ethernet PROFIBUS in NetPro.



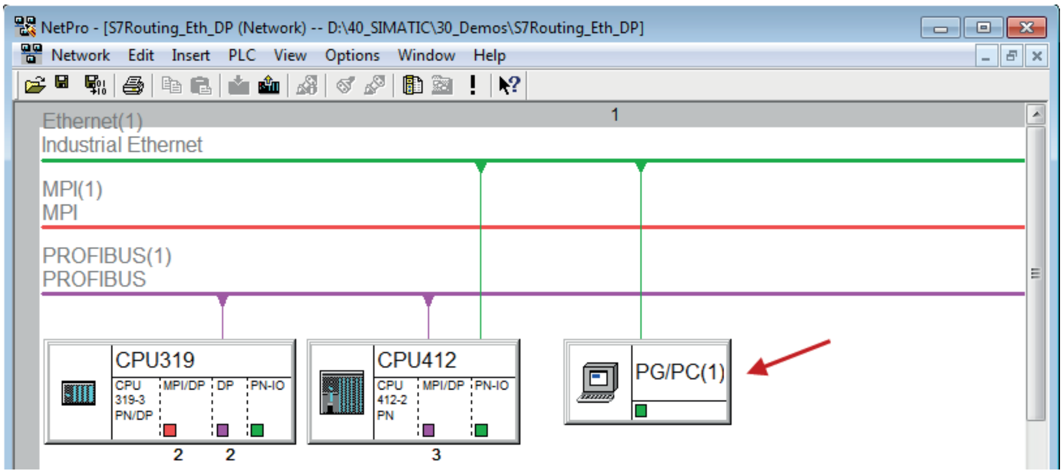
The engineering computer (also with *ibaPDA*) is to access the CPU319 controller. The computer and the controller are not directly connected via a common network/bus. The connection has to run over the CPU412 controller.

"Passing" the communication through this controller is called *S7 Routing*.

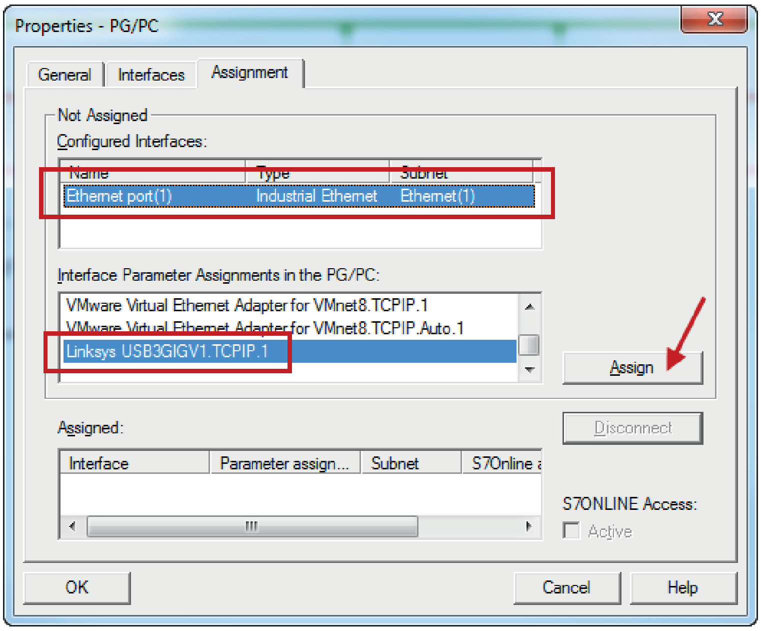
5.4.2.1 Configuration of STEP 7/NetPro

The following configuration steps are exclusively required for accessing the subordinate controller CPU319 via the SIMATIC STEP 7 programming software. For SINUMERIK, SINAMICS, or SIMOTION, you can apply similar steps. For using *ibaPDA*, these configuration steps are not required.

1. Add a PG/PC station and configure it.



2. Assign an interface (network card).



Not Assigned

Configured Interfaces:

Name	Type	Subnet
Ethernet port(1)	Industrial Ethernet	Ethernet(1)

Interface Parameter Assignments in the PG/PC:

VMware Virtual Ethernet Adapter for VMnet8.TCPIP.1
VMware Virtual Ethernet Adapter for VMnet8.TCPIP.Auto.1
Linksys USB3GIGV1.TCPIP.1

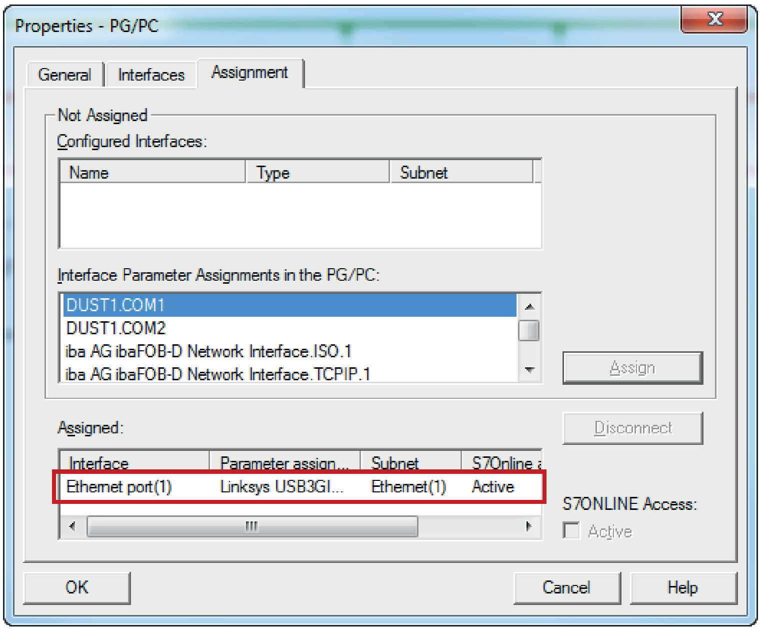
Assigned:

Interface	Parameter assign...	Subnet	S7Online a

S7ONLINE Access:

□ Active

→ Result:



Properties - PG/PC

Not Assigned

Configured Interfaces:

Name	Type	Subnet

Interface Parameter Assignments in the PG/PC:

DUST1.COM1
DUST1.COM2
iba AG ibaFOB-D Network Interface.ISO.1
iba AG ibaFOB-D Network Interface.TCPIP.1

Assigned:

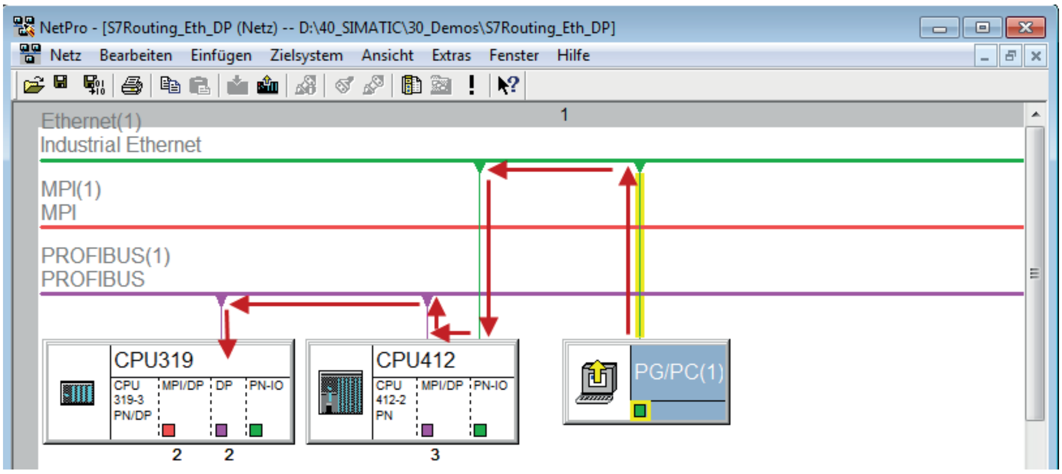
Interface	Parameter assign...	Subnet	S7Online a
Ethernet port(1)	Linksys USB3GI...	Ethernet(1)	Active

S7ONLINE Access:

□ Active

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).

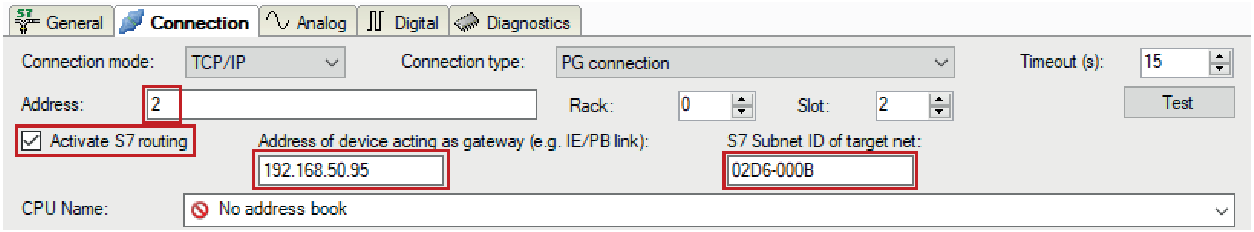


NetPro - [S7Routing_Eth_DP (Netz) -- D:\40_SIMATIC\30_Demos\S7Routing_Eth_DP]

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

5.4.2.2 Configuration of ibaPDA

Configure the following settings.



Connection mode: TCP/IP
Connection type: PG connection
Timeout (s): 15
Address: 2
Rack: 0
Slot: 2
☑ Activate S7 routing
Address of device acting as gateway (e.g. IE/PB link): 192.168.50.95
S7 Subnet ID of target net: 02D6-000B
CPU Name: No address book

Activate S7 routing
Enable this option to use S7 routing.

Address
Enter the address of the target control (here CPU319).

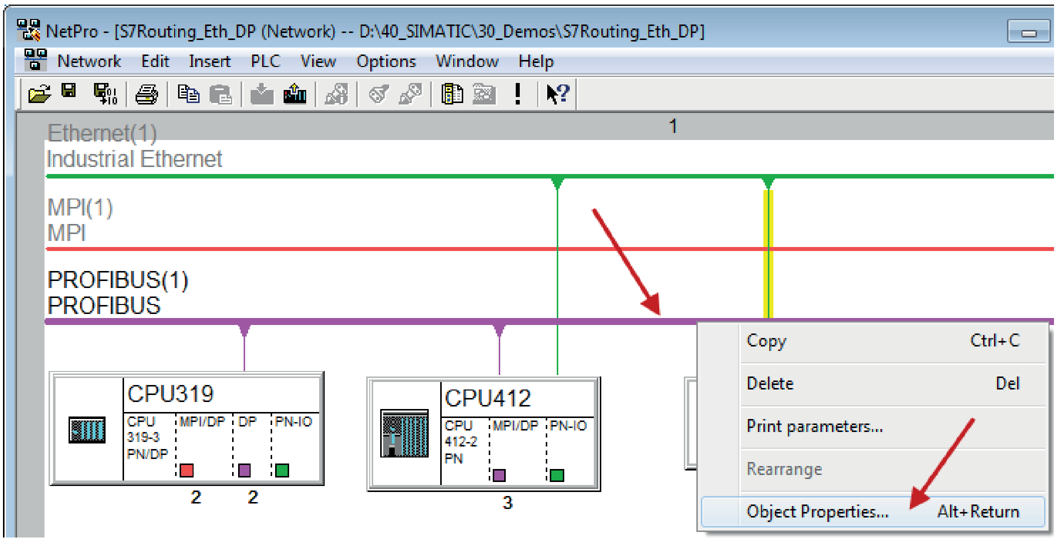
Address of device acting as gateway
Enter the address of the gateway (here CPU412).

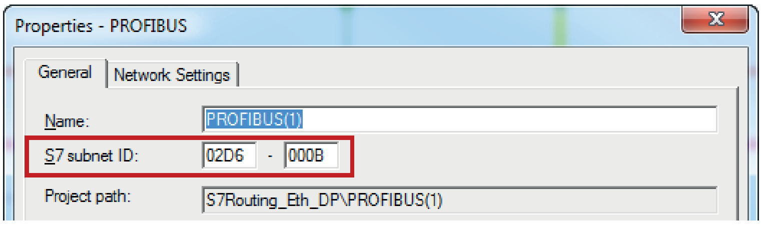
S7 subnet ID of target net
Enter the subnet ID from STEP 7 NetPro.

Identifying the S7 subnet ID in NetPro

You can identify the S7 subnet ID in NetPro.

Right-click on the secondary bus system and open the *Object Properties*.



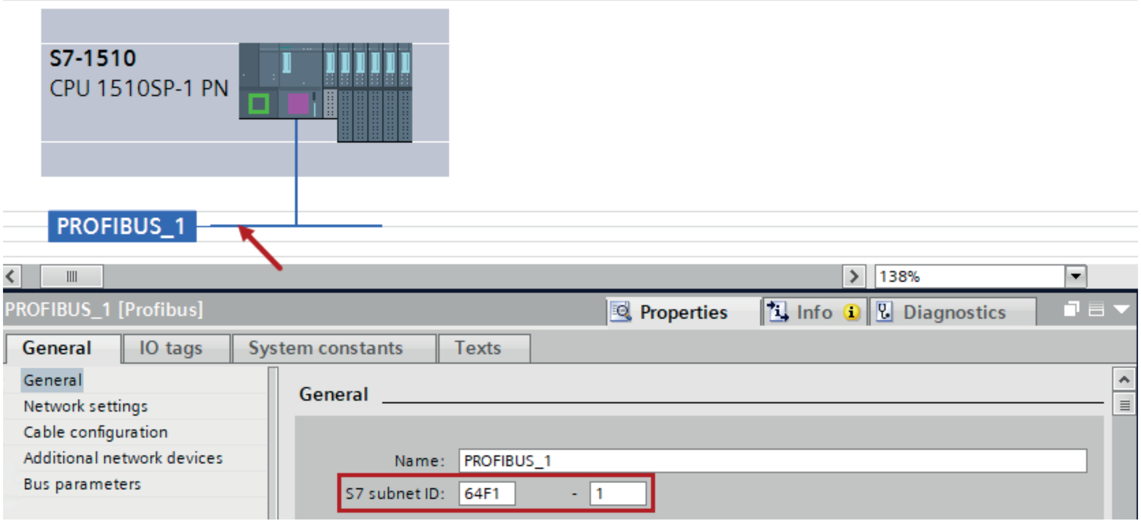


Name: PROFIBUS(1)
S7 subnet ID: 02D6 - 000B
Project path: S7Routing_Eth_DP\PROFIBUS(1)

Identifying the S7 subnet ID in TIA Portal

You can identify the S7 subnet ID in TIA Portal.

Click on the bus system and go to *Properties – General – General*.

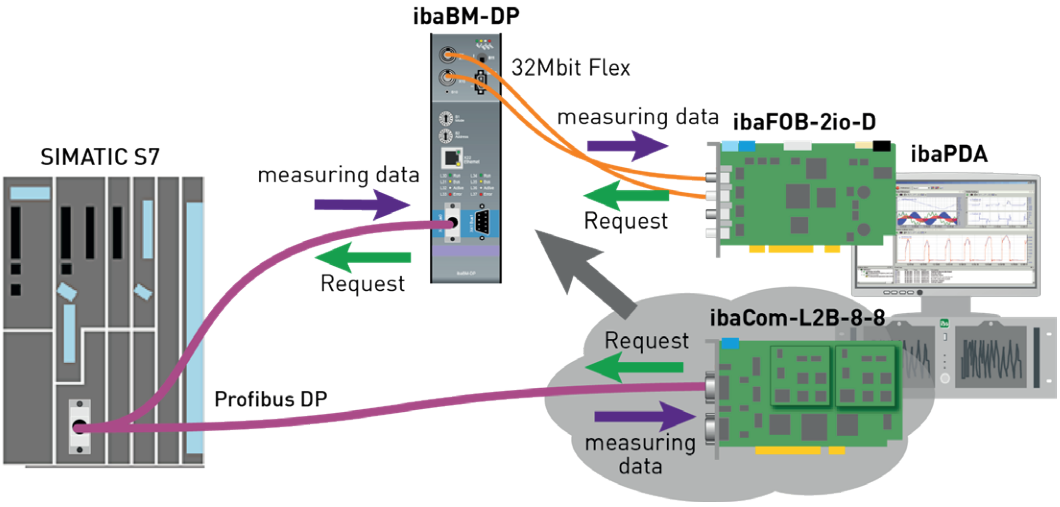


Name: PROFIBUS_1

S7 subnet ID: 64F1 - 1

5.5 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.



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!

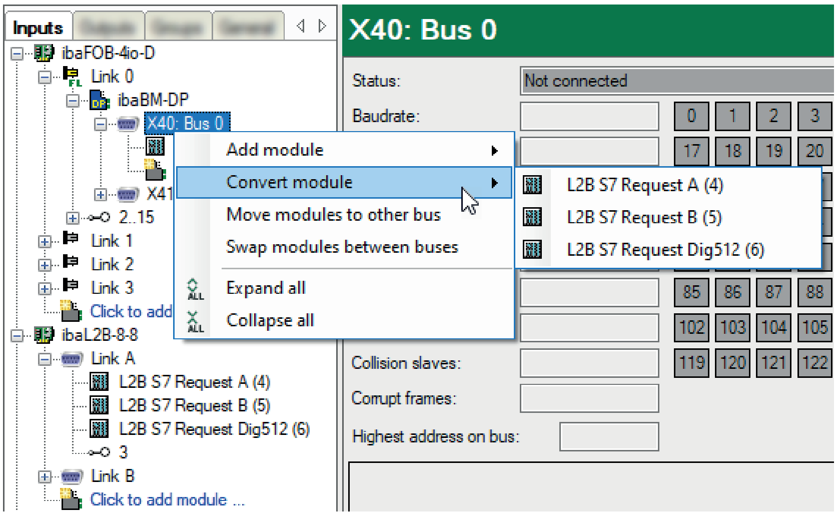
As for the *ibaCom-L2B* card also for the compatible solution with *ibaBM-DP* no network connection between *ibaPDA* computer and the S7-CPU is required. The measurement values (Request handshake) are requested via FO and are sent via the PROFIBUS I/O range.

The following steps have to be performed for the replacement:

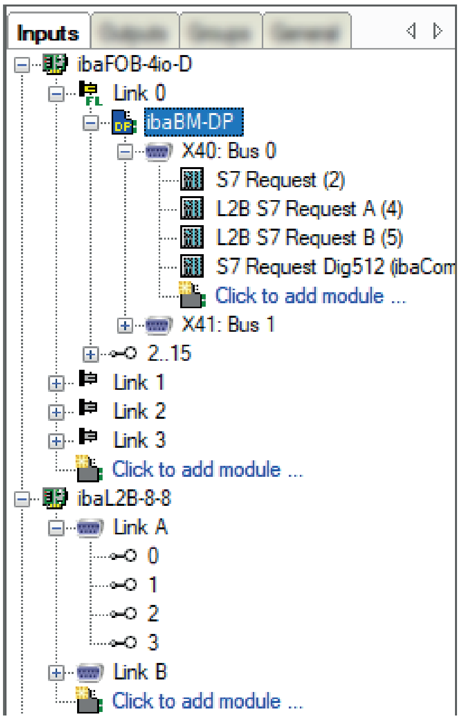
1. If required, install a new *ibaFOB-D* card in the *ibaPDA* computer. A card with input and output is needed (*ibaFOB-io-D*, *ibaFOB-2io-D* or *ibaFOB-4io-D*) as the compatible modules can only be used in the bidirectional 32Mbit Flex mode.
2. Connect the *ibaBM-DP* to the *ibaFOB-D* card via the bidirectional 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 *Convert module* in the context menu.

→ All available L2B Request-S7 modules are displayed.

Select here the modules you want to convert to an *ibaCom-L2B* compatible module (usually all modules).



→ The selected modules are deleted on the *ibaCom-L2B* card and converted into compatible modules on the *ibaBM-DP* device.

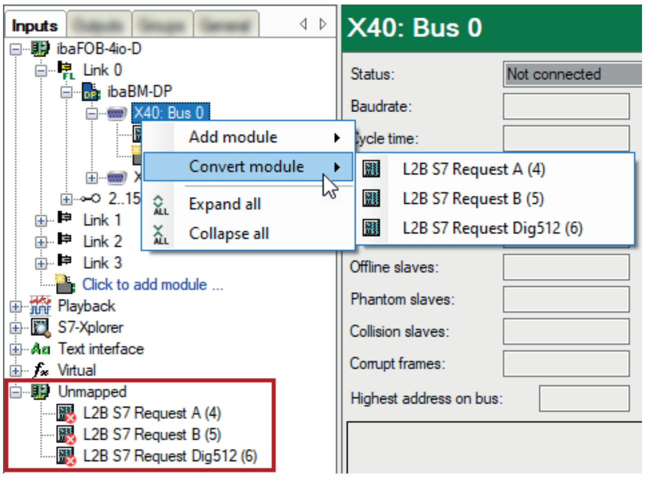


→ The conversion is finished.

6. Apply the new configuration with <OK>.

Note

You can also convert L2B Request-S7 modules if the modules are filed under the unmapped 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*.



Status: Not connected

5.6 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
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

Value ERROR_STATUS	Description
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_UDPact does not match with ibaREQ_M (ID)
301	version of ibaREQ_UDPact does not match with ibaREQ_M (FB)
302	version of ibaREQ_UDPact 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	error code of inner TUSEND/AG_SEND/AG_LSEND

Table 3: 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 07
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 4: 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" is 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" is 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 07
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 5: Error codes of Request block FC123

5.7 Usage of MPI/DP-TCP-adapters

You can connect Siemens S7 CPUs that do not have an Ethernet interface via MPI/DP-TCP-adapters to their MPI interface.

Different manufacturers offer adapters that convert the S7 communication from TCP/IP to MPI/PROFIBUS DP.

Basically, you can use these adapters together with *ibaPDA*. Therefore, you must configure the connection on the *ibaPDA*-side as a TCP-connection.

The access on the adapter is possible over its IP-address, which can be set by means of the associated software provided by the manufacturer.

In the connection configuration enter "0" for frame and the MPI address or DP address of the S7 CPU as slot.

For more information, refer to the corresponding device documentation.

6 Support and contact

Support

Phone:	+49 911 97282-14
Email:	support@iba-ag.com

Note

ℹ If you need support for software products, please state the number of the license container. 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
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