



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

Applications of the Request data interface for  
SIMATIC S7 TIA Portal/S7-1500

Manual part 1

Issue 4.0

Measurement Systems for Industry and Energy

[www.iba-ag.com](http://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

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

© iba AG 2024, All rights reserved.

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

The current version is available for download on our web site [www.iba-ag.com](http://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

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

## Contents

<b>1</b>	<b>About this documentation .....</b>	<b>6</b>
1.1	Target group and previous knowledge .....	6
1.2	Notations .....	7
1.3	Used symbols.....	8
1.4	Documentation structure .....	9
<b>2</b>	<b>System requirements .....</b>	<b>10</b>
<b>3</b>	<b>ibaPDA-Request-S7-DP/PN/ibaNet-E .....</b>	<b>12</b>
3.1	General information .....	12
3.2	Configuration and engineering SIMATIC S7 .....	15
3.3	Configuration and engineering ibaPDA.....	15
3.3.1	General interface settings.....	15
3.3.2	General module settings.....	15
3.3.3	Connection settings .....	16
3.3.3.1	Connection mode PC/CP .....	17
3.3.3.2	Connection mode TCP/IP .....	20
3.3.3.3	Connection mode TCP/IP S7-1x00 .....	22
3.3.4	Signal configuration .....	25
3.3.4.1	Selection via the absolute address of the operands.....	25
3.3.4.2	Selection via the symbolic operand addresses .....	26
3.3.4.3	Selection of CFC connectors .....	29
3.3.5	Address books .....	31
3.3.5.1	Creating address books offline from S7 project.....	32
3.3.5.2	Creating address books online from S7-1200/1500 CPU.....	34
<b>4</b>	<b>Request S7 variants.....</b>	<b>35</b>
4.1	Request using ibaNet-E.....	35
4.1.1	General information .....	35
4.1.2	SIMATIC S7-1500 configuration and engineering .....	36
4.1.2.1	Configuration in STEP 7 using the ibaREQ iba block family .....	36
4.1.2.2	Configuration in STEP 7 using the ibaREQsym iba block family .....	39
4.1.3	Configuration and engineering ibaPDA.....	41
4.1.3.1	General interface settings.....	41
4.1.3.2	General module settings.....	42

4.1.3.3	S7 Request module.....	43
4.1.3.4	S7 Request Decoder module .....	43
4.2	Request-S7 for ibaBM-PN .....	44
4.2.1	General information .....	44
4.2.2	Configuration and engineering SIMATIC S7-1500 .....	45
4.2.2.1	Network configuration.....	46
4.2.2.2	Configuring in STEP 7 with the iba block family ibaREQ.....	46
4.2.2.3	Configuration in STEP 7 using the ibaREQsym iba block family .....	48
4.2.3	Configuration and engineering ibaPDA.....	50
4.2.3.1	General interface settings.....	50
4.2.3.2	General module settings.....	52
4.2.3.3	Module S7 Request.....	52
4.2.3.4	Module S7 Request Decoder .....	52
4.2.4	Diagnostics.....	53
4.3	Request-S7 for ibaBM-PN in redundancy mode .....	54
4.3.1	General information .....	54
4.3.2	Configuration and engineering ibaPDA.....	55
4.4	Request-S7 for ibaBM-DP .....	56
4.4.1	General information .....	56
4.4.1.1	ibaCom-L2B compatibility mode .....	58
4.4.2	Configuration and engineering SIMATIC S7-1500.....	58
4.4.2.1	Network configuration.....	59
4.4.2.2	Configuration in STEP 7 using the ibaREQ iba block family .....	59
4.4.3	Configuration and engineering ibaPDA.....	61
4.4.3.1	General interface settings.....	61
4.4.3.2	General module settings.....	63
4.4.3.3	Module S7 Request.....	64
4.4.3.4	Modules S7 Request Decoder.....	64
4.4.4	Diagnostics.....	66
<b>5</b>	<b>Description of Request blocks .....</b>	<b>67</b>
5.1	ibaREQ iba block family .....	67
5.1.1	Device configuration.....	68
5.1.2	ibaREQ_M (FB1400).....	69
5.1.3	ibaREQ_PN (FB1401) .....	70

5.1.4	ibaREQ_DP (FB1402) .....	71
5.1.5	ibaREQ_NetE-Buffer (FB1408) .....	71
5.1.6	ibaREQ_NetE-Send (FB1409) .....	72
5.2	ibaREQsym iba block family .....	75
5.2.1	ibaREQsym_M .....	76
5.2.2	ibaREQsym_PN .....	77
5.2.3	ibaREQsym_NetE .....	78
<b>6</b>	<b>Diagnostics.....</b>	<b>80</b>
6.1	License .....	80
6.2	Visibility of the interface.....	80
6.3	Log files.....	81
6.4	Connection diagnostics with PING.....	82
6.5	Connection diagnostics with PG/PC interface .....	83
6.6	Module diagnostics.....	84
<b>7</b>	<b>Appendix .....</b>	<b>85</b>
7.1	iba S7 library .....	85
7.1.1	iba S7 library for SIMATIC TIA portal.....	85
7.1.1.1	Integrating the library into TIA Portal .....	87
7.1.1.2	Using the blocks in TIA Portal .....	88
7.2	Application examples.....	89
7.3	Setting PG/PC interface/defining new access point.....	91
7.4	S7 routing .....	94
7.4.1	Routing from Ethernet to Ethernet .....	94
7.4.1.1	Configuration of TIA Portal .....	95
7.4.1.2	Configuration of ibaPDA .....	96
7.4.2	Routing from Ethernet to PROFIBUS .....	98
7.4.2.1	Configuration of TIA Portal .....	98
7.4.2.2	Configuration of ibaPDA .....	98
7.5	Error codes of Request blocks .....	100
<b>8</b>	<b>Support and contact.....</b>	<b>104</b>

# 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-CPU's. 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 <i>Logic diagram</i>
Calling the menu command	<i>Step 1 – Step 2 – Step 3 – Step x</i> Example: Select the menu <i>Logic diagram – Add – New function block</i> .
Keys	<Key name> Example: <Alt>; <F1>
Press the keys simultaneously	<Key name> + <Key name> Example: <Alt> + <Ctrl>
Buttons	<Key name> Example: <OK>; <Cancel>
Filenames, paths	<i>Filename, Path</i> Example: <i>Test.docx</i>

## 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 System requirements

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

- *ibaPDA* v8.5.0 or higher
- Base license for *ibaPDA*
- Additional license for *ibaPDA-Request-S7-DP/PN/ibaNet-E*
- *ibaBM-PN*, *ibaBM-DP*, *ibaBM-DPM-S*, *ibaCom-L2B* or *ibaPDA-interface-ibaNet-E*
- *ibaFOB-D* board in the *ibaPDA* computer when using *ibaBM-PN*, *ibaBM-DP*, or *ibaBM-DPM-S* for coupling using fiber optics
- SIMATIC S7 controller S7-300, S7-400, S7-400H, S7-1500, WinAC (not all Request S7 variants are available for all controller types), for access to optimized data blocks S7-1500 firmware V3 or later
- If using PC/CP connections:
  - SIMATIC STEP 7/SIMATIC NET, or
  - SIMATIC TIA Portal
- SIMATIC CFC (from V6.0, not for TIA Portal), if signals are to be selected using drag & drop

For integration of Request blocks into S7 program:

- SIMATIC STEP 7 V5.4 SP5 or higher, or
- SIMATIC STEP 7 (TIA Portal) V16 or higher (function block libraries for older versions may be available on request), V18 or higher for access to optimized data blocks

For further requirements for the used computer hardware and the supported operating systems, refer to the *ibaPDA* documentation.

### System constraints

- Access to S7-1200 controllers is not supported.
- Connectors for STEP 7 CFC blocks that have constant values assigned have no operand address. They are marked as constant in the address book and cannot be selected as signals.
- If function blocks (FBs) are used in STEP 7 CFC, the internal static variables for the FB also appear in the address book because they are handled in exactly the same way as connectors by the compiler. These are to be ignored.
- *ibaPDA* supports the following data types for acquisition:
  - BOOL, BYTE, WORD, DWORD, INT, DINT, REAL, TIME, CHARAll other data types appear in the address book but cannot be entered in the signal list.
- For functions (FCs) with connections of STRING, POINTER, STRUCT or ANY data types under STEP7 CFC, the interpretation of the SCL code does not work, as the source does not contain any references to the data types.

**ibaREQsym block family**

- TIA Portal V18 or higher
- Firmware V3.0 or higher
- Supported items: E, A, M and items from data blocks, only elementary data types and individual items from structured data types
- Variables must have the attribute "Accessible from HMI/OPC UA/Web API" or "Writable from HMI/OPC UA/Web API".
- Max. length of symbolic variable name (fully qualified name including namespace): 254 UTF-16 characters
- Max. 10 Request instances can be simultaneously active per S7 CPU.
- Max. 2000 symbols can be simultaneously used per S7 CPU.

Further information about the "ResolveSymbols" and "MoveResolvedSymbolsToBuffer" functions can be found in the Siemens documentation.

**License information**

Order no.	Product name	Description
31.001310	ibaPDA-Request-S7-DP/PN/ibaNet-E	Extension license for an <i>ibaPDA</i> system for use of Request-S7 with <i>ibaBM-PN</i> , <i>ibaBM-DP</i> , <i>ibaBM-DPM-S</i> , <i>ibaCom-L2B</i> , or <i>ibaPDA-interface-ibaNet-E</i>

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

This manual is divided into three main sections:

- General section, valid for all communication channels:  
Chapter ↗ *ibaPDA-Request-S7-DP/PN/ibaNet-E*, page 12
- Variant-specific section with special information for each variant:  
Chapter ↗ *Request S7 variants*, page 35  
Variants for S7-300 and S7-400 are described in Part 2 of the manual.
- Description of individual S7 function blocks:  
Chapter ↗ *Description of Request blocks*, page 67

### Other documentation



The associated device manuals contain detailed information about the devices:

- ibaCom-L2B-x-8 board
- ibaBM-DP
- ibaBM-DPM-S
- ibaBM-PN

### 3.1 General information

For measurement data acquisition with *ibaPDA-Request-S7-DP/PN/ibaNet-E* the measurement data is actively transmitted from the controller to *ibaPDA* via PROFIBUS slaves or PROFIBUS devices implemented in the device used (e.g. *ibaBM-DP*) or using the *ibaNet-E* (UDP) protocol. For this purpose, one or more Request blocks (FB/FC + DBs) have to be integrated for each slave/device/connection in the S7-CPU program. These Request blocks are used to send the S7 operands/symbols selected by the user within *ibaPDA* to *ibaPDA*. If the signal selection is changed, no changes to the S7 program are needed.

You can select the signals to be measured either using the absolute operand address or using the symbolic name with support from the *ibaPDA* address book browser. This browser allows access to all defined symbols in the connected STEP 7 project.

For S7-1500 you can use optimized data blocks. Signals within these data blocks can only be accessed using their symbolic names and not using the address or the operand. For CPUs with firmware V3 or higher, it is also possible to access this data using special Request blocks.

Block family	S7 CPU firmware	TIA portal	Access to optimized data blocks
ibaREQ	Unlimited	From V16	No
ibaREQsym	From V3	From V18	Yes

For acquisition of data from SIMATIC S7 CPUs with *ibaPDA*, either different hardware devices from iba AG are available or the option of transferring the data using *ibaNet-E* (UDP).

**Via PROFIBUS**

- *ibaBM-DP* or predecessor *ibaBM-DPM-S*
- *ibaCom-L2B* PCI board

**Via PROFINET**

- *ibaBM-PN*

**Via ibaNet-E**

- Default network adapter

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

<b>SIMATIC STEP 7 V5.x (SIMATIC Manager)</b>					
SIMATIC S7 CPU	ibaBM-DP	ibaBM-DPM-S	ibaCom-L2B	ibaBM-PN	ibaNet-E
S7-300 Integrated DP interface	X	X	X	-	
S7-300 CP342-5 (PROFIBUS)	-	-	X	-	-
S7-400 Integrated DP interface and CP443-5	X	X	X	-	-
S7-400H Integrated DP interface and CP443-5 (PROFIBUS)	X	X	-	-	-
S7-300 Integrated PN interface	-	-	-	X	-
S7-400 Integrated PN interface and CP443-1 (PROFINET)	-	-	-	X	-

Table 1: Available Request blocks for SIMATIC Manager

<b>SIMATIC STEP 7 V1x Professional (TIA Portal)</b>				
SIMATIC S7 CPU	ibaBM-DP	ibaBM-DPM-S	ibaBM-PN	ibaNet-E
S7-300 Integrated DP interface	X	X	-	-
S7-300 Integrated PN interface	-	-	X	-
S7-400 Integrated DP interface and CP443-5	X	X	-	-
S7-400 Integrated PN interface and CP443-1 (PROFINET)	-	-	X	-
S7-1500 Integrated DP interface and CM1542-5 or CP1542-5 (PROFIBUS)	X	X	-	-
S7-1500 Integrated PN interface and CM1542-1 (PROFINET)	-	-	X	-
S7-1500 Integrated PN interface and communication processor with UDP capability	-	-	-	X

Table 2: Available Request blocks for SIMATIC TIA portal

**Note**

You can find the Request blocks in the iba S7 library, see [📌 iba S7 library](#), page 85. Always use the latest version of this library.

## 3.2 Configuration and engineering SIMATIC S7

Independent of the used communication interface, carry out the following configuration steps on the SIMATIC side:

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

For detailed information about this subject, see the respective paragraph in chapter [Request S7 variants](#), page 35.

## 3.3 Configuration and engineering ibaPDA

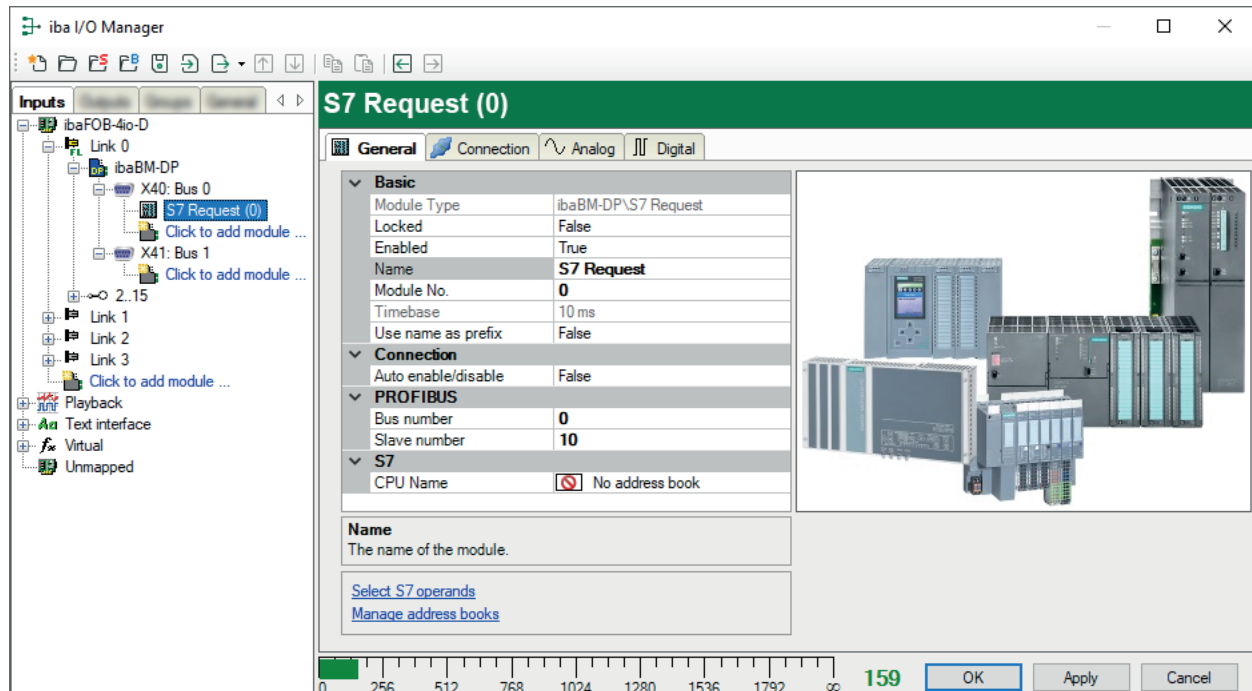
### 3.3.1 General interface settings

For the configuration of the device-specific settings, please see chapter [Request S7 variants](#), page 35.

### 3.3.2 General module settings

To configure a module, select it in the tree structure.

All modules have the following setting options.



#### Note



General settings that are not described here are module-specific settings. These settings are described in chapter [Request S7 variants](#), page 35 under the respective module version.

**Basic settings****Module Type (information only)**

Indicates the type of the current module.

**Locked**

You can lock a module to avoid unintentional or unauthorized changing of the module settings.

**Enabled**

Enable the module to record signals.

**Name**

You can enter a name for the module here.

**Module No.**

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

**Timebase**

All signals of the module are sampled on this timebase.

**Use module name as prefix**

This option puts the module name in front of the signal names.

**S7****CPU Name**

Select the S7-CPU that is connected to this module. When selecting a S7-CPU (incl. the address book), you can select the signals symbolically. Otherwise, you select the signals via the S7 operand.

This requires that address books have already been generated. Otherwise, the selection list is empty. Using *Create address book* in the selection list, you get directly to the address book generator, see [🔗 Creating address books offline from S7 project](#), page 32.

**3.3.3 Connection settings**

Configure the connection of the module to the controller in the *Connection* tab.

*ibaPDA* supports the following controllers, connection modes and selection methods:

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

Configure different settings depending on the connection mode.



### 3.3.3.1 Connection mode PC/CP

This mode activates a connection over the interface cards of the computer that are configured using SIMATIC Net.

You can use the interfaces configured in SIMATIC Net, e.g.:

- MPI adapter (COM)
- MPI adapter (USB)
- PROFIBUS (CP5611, CP5622)
- TCP/IP (RFC1005)
- ...

#### Note



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

#### Connection mode

Selection of the PC/CP connection mode

#### Connection type

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

#### Timeout

Here you can specify a value for the timeout in seconds for establishing the connection and for read accesses. If the time set here is exceeded, *ibaPDA* declares the controller as not accessible or not responsive.

#### Access point for applications

Selection of the access point to be used

For more information on creating and configuring an access point, see

➤ *Setting PG/PC interface/defining new access point*, page 91.

**Note**

Configure available access points in SIMATIC Net with the "PG/PC interface settings" tool by Siemens.

For the connection of *ibaPDA-Request-S7-DP/PN/ibaNet-E* to SIMATIC S7 via PC/CP connections, it is generally recommend setting a specific access point for *ibaPDA* when *ibaPDA-Request-S7-DP/PN/ibaNet-E* and SIMATIC Manager run on the same computer. With an own access point, there is no longer the risk that the access for *ibaPDA-Request-S7-DP/PN/ibaNet-E* will be disturbed in case the standard access point is changed in the SIMATIC Manager.

**<Configure PG/PC interface>**

This button opens the dialog box for setting the PG/PC interface of SIMATIC STEP 7.

**Address**

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

**Rack**

Rack number of the controller (default: 0)

**Slot**

Slot number of the controller in the rack  
(Use "0" for S7-1500 CPUs.)

**Activate S7 routing**

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

Two additional input fields appear:

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

For more information on S7 routing, see ➔ *S7 routing*, page 94.

**DB**

Number of the data block used as *ibaPDA* communication interface (ibaREQ\_DB)

**PROFINET Device Slot or PROFIBUS slave number (not for ibaNet-E)**

- PROFINET: Number of the used slot of the *ibaBM-PN*
- PROFIBUS: 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.

**CPU Name**

Selection of the linked address book

## Detect S7 restart

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

## <Test>

*ibaPDA* tests the connection to the CPU and displays available diagnostic data.

**S7 Request (2)**

General Connection Analog Digital S7 request info

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

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

Address: 192.168.123.1 Rack: 0 Slot: 0 Test

☐ Activate S7 routing

DB: 15 PROFINET device slot: 1

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

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

## Tip



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

Check the following:

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

### 3.3.3.2 Connection mode TCP/IP

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

**S7 Request (2)**

General Connection Analog Digital S7 request info

Connection

Connection mode: TCP/IP Connection type: PG connection Timeout (s): 15

Address: 192.168.123.1 Rack: 0 Slot: 0 Test

☐ Activate S7 routing

DB: 15 PROFINET device slot: 1

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

#### Connection mode

Selection of the TCP/IP connection mode

#### Connection type

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

#### Timeout

Here you can specify a value for the timeout in seconds for establishing the connection and for read accesses. If the time set here is exceeded, *ibaPDA* declares the controller as not accessible or not responsive.

#### Address

IP address of the controller

#### Rack

Rack number of the controller (default: 0)

#### Slot

Slot number of the controller in the rack  
(Use "0" for S7-1500 CPUs.)

#### Activate S7 routing

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

Two additional input fields appear:

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

For more information on S7 routing, see [S7 routing](#), page 94.

#### DB

Number of the data block used as *ibaPDA* communication interface (ibaREQ\_DB)

#### PROFINET Device Slot or PROFIBUS slave number (not for ibaNet-E)

- PROFINET: Number of the used slot of the *ibaBM-PN*
- PROFIBUS: 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.

## CPU Name

Selection of the linked address book

## Detect S7 restart

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

## <Test>

*ibaPDA* tests the connection to the CPU and displays available diagnostic data.

**S7 Request (2)**

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

Connection mode: TCP/IP | Connection type: PG connection | Timeout (s): 15

Address: 192.168.123.1 | Rack: 0 | Slot: 0 | **Test**

☐ Activate S7 routing

DB: 15 | PROFINET device slot: 1

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

---

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

Device slot	Length	Status
1	252	available

## Tip



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

Check the following:

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

### 3.3.3.3 Connection mode TCP/IP S7-1x00

This mode activates a connection via the standard network interface of the computer. You can only use this mode in combination with S7-1500 CPUs.

**S7 Request (2)**

General Connection Analog Digital S7 request info

Connection

Connection mode: TCP/IP S7-1x00 Connection type: PG connection Timeout (s): 15

Address: 192.168.123.1 Test

Password:  ☐ Use secure communication

DB: None PROFINET device slot: 1 Load address book from S7

CPU Name:  ☒ Detect S7 restart (This applies to all S7 request modules)

#### Connection mode

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

#### Connection type

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

#### Timeout

Here you can specify a value for the timeout in seconds for establishing the connection and for read accesses. If the time set here is exceeded, *ibaPDA* declares the controller as not accessible or not responsive.

#### Address

IP address of the controller

#### Password

Depending on the configuration in the controller, access to the PLC may be protected by a password. In this case, enter this password here.

#### Use secure communication

The S7-1500 controller supports secure communication via TLS encryption with TIA Portal v17 or higher. In TIA Portal, you can set secure PG/PC and HMI communication.

If you have activated this option in the controller, you must also activate secure communication in *ibaPDA*.

#### DB

Number of the data block used as *ibaPDA* communication interface (ibaREQ\_DB)

#### PROFINET Device Slot or PROFIBUS slave number (not for ibaNet-E)

- PROFINET: Number of the used slot of the *ibaBM-PN*
- PROFIBUS: 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.

## CPU Name

Selection of the linked address book (only TIA Portal address books available).

### Note



**Port 102** in the firewall must be released for the communication with the CPU for signal request.

## Detect S7 restart

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

### <Test>

*ibaPDA* tests the connection to the CPU and displays available diagnostic data.

**S7 Request (2)**

General Connection Analog Digital S7 request info

Connection

Connection mode: TCP/IP S7-1x00 Connection type: PG connection Timeout (s): 15

Address: 192.168.80.90 Test

Password: Use secure communication

DB: ibaREQ\_DB\_UDP (DB) PROFINET device slot: 1 Load address book from S7

CPU Name: PLC (192.168.80.90) Detect S7 restart (This applies to all S7 request modules)

Connection established  
MLFBNr of PLC is: **6ES7 516-3AN00-0AB0**  
Reading **ibaREQ\_DB\_PN (DB17)**  
DB id: **ibaREQ-S7-M**  
DB version: **1.0.0.0**  
FB version: **1.3:1.3**  
DB length: **9120**  
Max. pointers: **512**  
Max. data bytes: **252**

HW version: **0**  
Total memory size: **0**  
DB memory size: **0**  
DB used size: **0**  
Code memory size: **0**  
Code used size: **0**  
No. inputs: **32768**  
No. outputs: **32768**  
No. markers: **16384**  
No. timers: **2048**  
No. counters: **2048**  
I/O space: **0**  
Local datasize: **0**

Device slot	Length	Status
1	252	available

**<Load address book from S7>**

By clicking on this button *ibaPDA* reads the list of symbols directly from the PLC and stores it in an address book for further use in the symbol browser.

**S7 Request (2)**

General Connection Analog Digital S7 request info

Connection

Connection mode: TCP/IP S7-1x00 Connection type: PG connection Timeout (s): 15

Address: 192.168.80.90 Test

Password: Use secure communication

DB: ibaREQ\_DB\_UDP (DB) PROFINET device slot: 1 Load address book from S7

CPU Name: PLC (192.168.80.90) Detect S7 restart (This applies to all S7 request modules)

Connecting to S7 at 192.168.80.90  
 Connected successfully to S7 at 192.168.80.90  
 MLFBNr of PLC is: 6ES7 516-3AN00-0AB0  
 Loading symbols from S7  
 Successfully loaded symbols  
 Writing addressbook  
 Successfully created addressbook: PLC (192.168.80.90)

**Tip**

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

Check the following:

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

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

For more information, see [↗ Device configuration](#), page 68



### 3.3.4 Signal configuration

In the I/O Manager, you select the signals to be acquired. There are 3 ways of selecting measured values:

- Selection using the absolute address of the S7 operands, see [↗ Selection via the absolute address of the operands](#), page 25
- Selection using the S7 symbol addresses (symbol table and symbols from DBs) using a symbol browser, see [↗ Selection via the symbolic operand addresses](#), page 26
- Selection using the CFC connectors (when programming the CPU with SIMATIC CFC), see [↗ Selection of CFC connectors](#), page 29

SIMATIC CPU	Access using absolute address	Access using symbol	Access using CFC connectors <sup>1)</sup>
S7-300	X	X	X
S7-400	X	X	X
WinAC	X	X	X
S7-1500	X	X	-

Supported operand ranges:

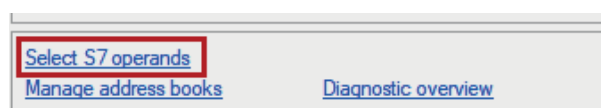
Operand range	SIMATIC CPUs S7-300/400	SIMATIC CPUs S7-1500
Inputs (I)	X	X
Peripheral inputs (PI)	X	-
Outputs (O)	X	X
Markers (M)	X	X
Data blocks (DB)	X	X

For S7-1500 you can use optimized data blocks. Signals within these data blocks can only be accessed using their symbolic names and not using the address or the operand. To access these data ranges, you must use the Request blocks from the ibaREQsym family, see [↗ ibaREQsym iba block family](#), page 75.

#### 3.3.4.1 Selection via the absolute address of the operands

You have two options to select the measurement values via the operand address:

- In the module's *General* tab, click on the *Select S7 operands* link.



The S7 operand editor opens.

<sup>1)</sup> The prerequisite for this is use of the optional S7-CFC SIMATIC STEP 7 package. For TIA portal, SIMATIC CFC is not supported.

- In the *Analog* or *Digital* tab, click in a cell in the *S7 operand* column.

The button <...> appears. Click the button to open the S7 operand editor.

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

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

S7 General Connection Analog Digital Diagnostics							
	Name	Unit	Gain	Offset	S7 Operand	S7 DataType	Active
0	counter 16bit		1	0	MW 100	WORD	<input checked="" type="checkbox"/>
1	counter 32bit		1	0	MD 104	DWORD	<input checked="" type="checkbox"/>
2	sinus		1	0	MD 112	REAL	<input checked="" type="checkbox"/>
3	cosinus		1	0	MD 116	REAL	<input checked="" type="checkbox"/>
4			1	0		INT	<input type="checkbox"/>

You can enter the desired operand address also directly in the *S7 Operand* column without using the S7 operand editor.

### 3.3.4.2 Selection via the symbolic operand addresses

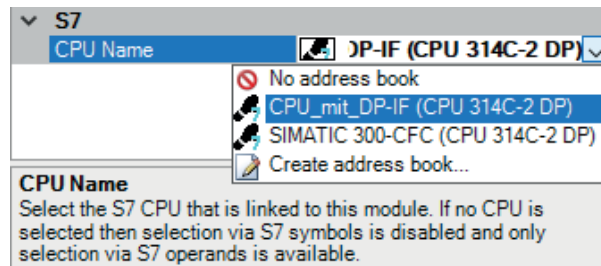
An advantage of this way of access is that the symbol addresses are applied automatically in *ibaPDA* as signal names.

Requirements for this method of access:

- The signals to be measured already have an entry in the S7 symbol table, the PLC variable list or in a data block.
- An address book has been created (see chapter [➤ Address books](#), page 31).

## Integrating an address book into a module

- In the module's *General* tab under *CPU Name*, select the S7-CPU you want to assign this module to.



→ In the *Analog* and *Digital* tabs, an additional column *S7 Symbol* is displayed.

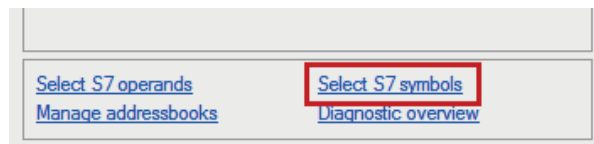
S7 General Connection Analog Digital Diagnostics							
Name	Unit	Gain	Offset	S7 Operand	S7 DataType	Active	
0		1	0		INT	<input type="checkbox"/>	
1		1	0		INT	<input type="checkbox"/>	
2		1	0		INT	<input type="checkbox"/>	
3		1	0		INT	<input type="checkbox"/>	
4		1	0		INT	<input type="checkbox"/>	

→ Now you can access symbol addresses using the S7 CFC- and Symbol Browser (in short: symbol browser).

## Selecting signals via the symbol browser

You have two options to select the signals to be measured:

- In the module's *General* tab, click on the *Select S7 symbols* link.



The symbol browser opens.

In the symbol browser, you can select all symbols of the address book. *ibaPDA* enters the selected signals automatically in the appropriate table *Analog* or *Digital*. You can add several signals successively.

- On the *Analog* or *Digital* tab, click in a cell of the *S7 Symbol* column.

The button <...> appears. Click this button to open the symbol browser.

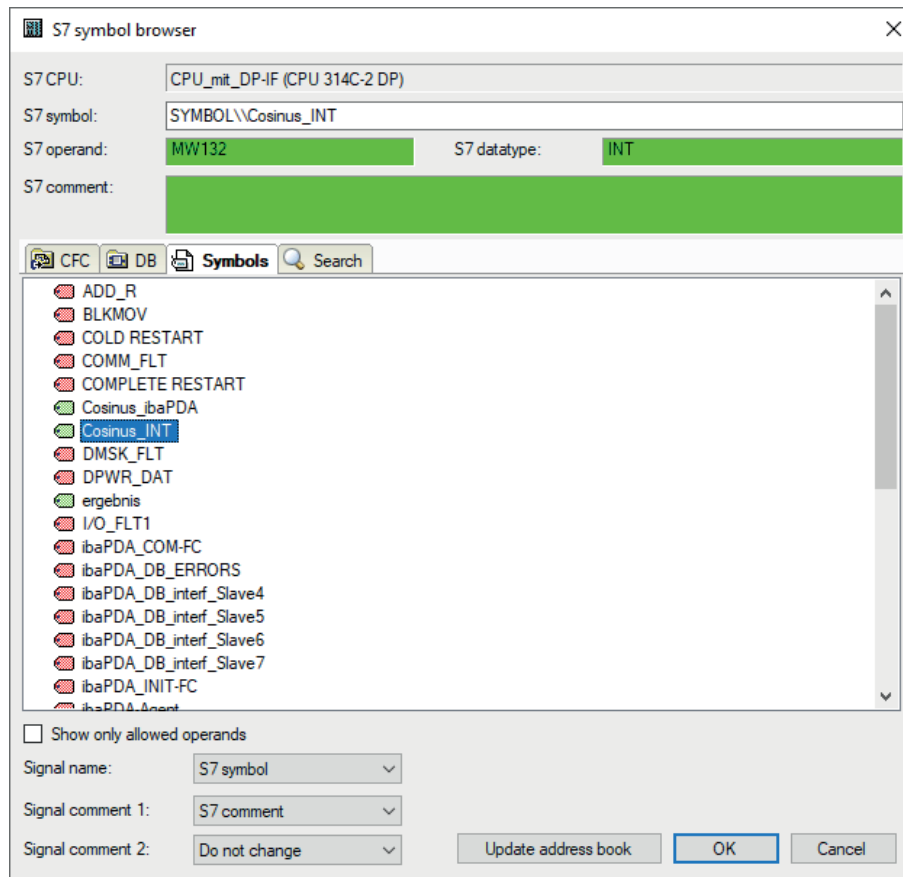
In the symbol browser, you can only select symbols with the data type matching the table. *ibaPDA* enters the symbol in the previously selected row of the signal table. After every selection, the symbol browser closes.

## User interface of the symbol browser

In the symbol browser, you have the following options:

- CFC variables:  
On the *CFC* tab, you can select the configured CFC variables consisting of the configured names of chart, block and connector.

- **DB variables:**  
On the *DB* tab, you can select the individual data blocks and their variables.
- **Symbol table:**  
On the *Symbols* tab, you can select the entries of the S7 symbol table.
- **Search tab:**  
You can search the variables with a part of their name.



After selecting the variable, the symbol browser shows the operand address, the data type and a comment.

The variables have the following colors:

Green	The operand is valid. You can transfer it to the signal table with <Add> or <OK>.
Yellow	The operand has a data type that does not match the selected row or table, e. g. in case you have selected a Boolean variable as analog value or an integer value as digital value.
Red	The operand has a data type that is not supported by <i>ibaPDA</i> , or the operand is a constant.

### Show only allowed operands

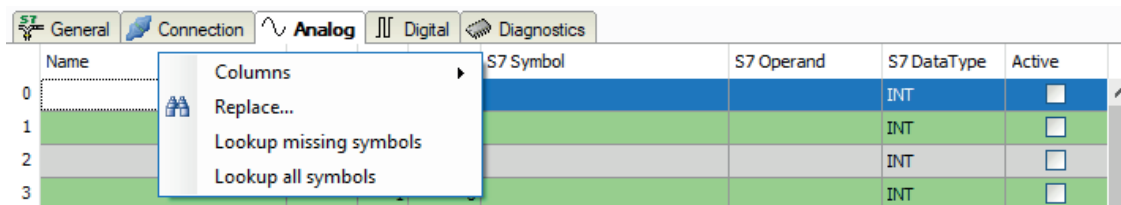
If you activate this option, the symbol browser shows only operands and symbols that *ibaPDA* supports or that match the signal table from which you opened the browser (i.e. no red and yellows ones).

### Signal name, signal comment 1 and 2

Usually, *ibaPDA* adopts the symbolic signals name from STEP 7 as signal name in the I/O Manager. Using these three drop-down menus, you can change the signal name and both comments.

Please select an option from the alternatives offered. If a signal name or comment in the signal table should not be changed, select *Do not change*.

### Looking up symbols in the signal table



*ibaPDA* can look up the symbol that corresponds with an operand.

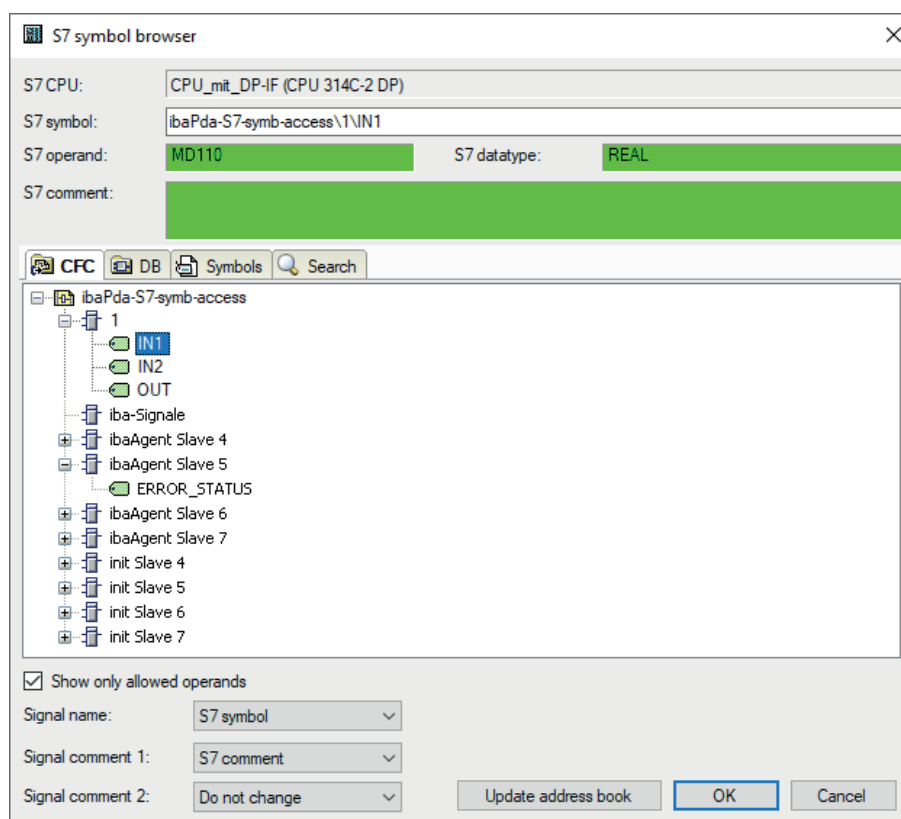
Right-click on the table header of the signal table.

- Select *Lookup missing symbols* to search only for the missing symbols.
- Select *Lookup all symbols* to search and replace all symbols. The command executes a backward resolution of the S7 symbols out of the S7 operands. *ibaPDA* searches the symbol table first, then CFC and finally the DBs for the operand.

### 3.3.4.3 Selection of CFC connectors

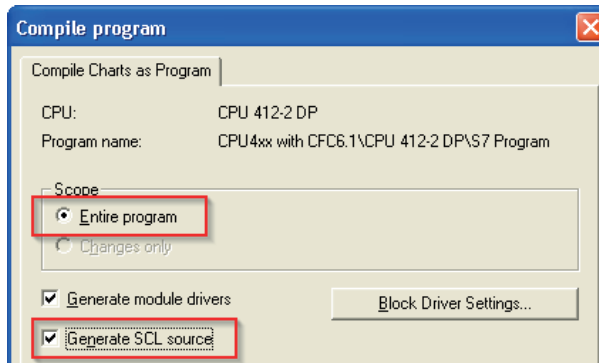
In order to select CFC connectors for the measurement, open the symbol browser first, see [↗ Selection via the symbolic operand addresses](#), page 26.

In the symbol browser, you open the *CFC* tab and select the signal. The connectors are listed hierarchically by chart name, module name and connector name:



**Note**

If no connectors are shown in the CFC tab, the SCL sources might not have been translated in the STEP 7 project. Activate the following options in the dialog for compiling the program in the SIMATIC software:



Afterwards, generate the address books again.

**Note**

When compiling a CFC program, DB addresses are assigned automatically in STEP 7 to the connectors. It may happen that other DB addresses are assigned to connectors, depending on the scope of the program changes that have been made between two compilation runs.

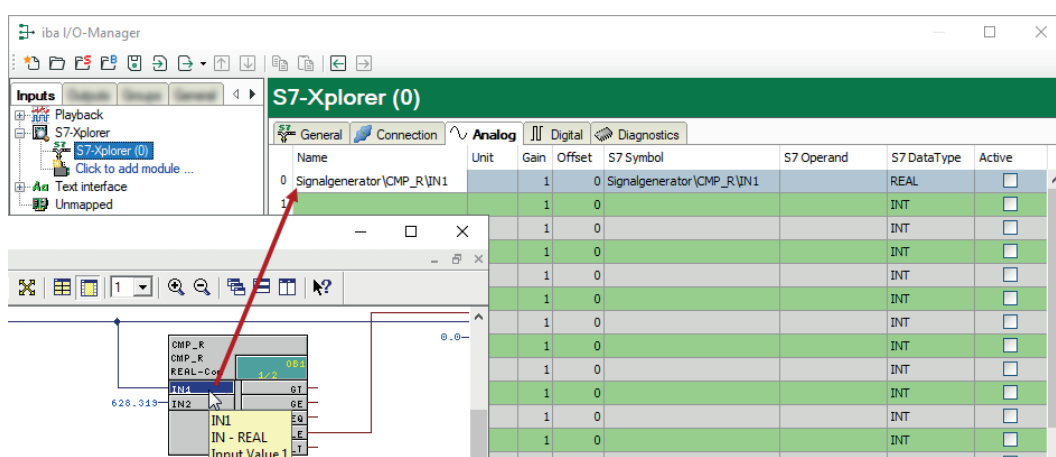
In this case, also the address book for *ibaPDA* has to be generated again. The symbolically configured signals are automatically checked in the I/O Manager whereas the related absolute S7 operands are updated.

**Special function Drag & Drop**

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

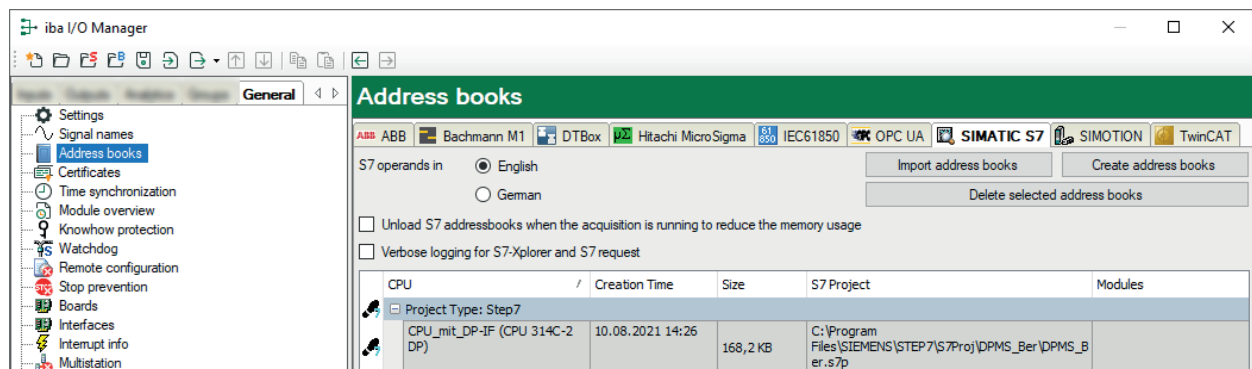
1. Open the I/O Manager of *ibaPDA* and the signal table of the required module.
2. Start the CFC editor on the same computer as the *ibaPDA* client.
3. Drag the connector from the CFC editor into the desired line of the signal table in the I/O Manager of *ibaPDA*.

→ The CFC connector is now a measured signal in the signal table of the module.



### 3.3.5 Address books

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



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

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

#### S7 operands in English/German

Here you can choose the language in which the S7 operands will later be available when browsing through the signal tables.

#### <Create address books>

This button opens the "S7 address book generator" dialog. You can select the source directory of an S7 project for creating the S7 address book. This can be a local or network drive.

#### <Import address books> button

Import address books which are already available as ZIP files.

#### <Delete selected address books> button

Delete address books from the *ibaPDA* server's directory.

#### Unload S7 address book when the acquisition is running to reduce the memory usage

By enabling this option, the address book is outsourced to the hard disk during the acquisition in order to free up the main memory for the acquisition.

#### Table

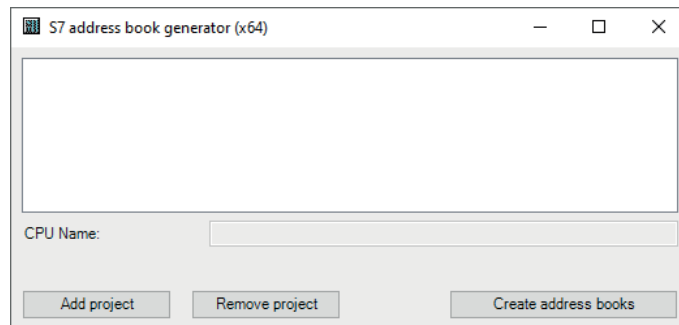
List of all address books that are currently available in the system with name, creation date, path of STEP 7 project or IP address of the CPU in case of online generated address books and location where the address book is used.

### 3.3.5.1 Creating address books offline from S7 project

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

You create an address book using the S7 address book generator.

#### S7 address book generator



#### CPU Name

Name of the CPU

#### Step 7 HW Config export

A HW config export file can be selected as an option (useful when using an iba bus monitor in sniffer mode)

#### Comment language

Selection of the language that is to be imported for comment texts (only available for SIMATIC TIA portal projects)

#### Add project

Adds a new project to the list

#### Remove project

Removes the marked project from the list

#### Create address books

Creates address books from the selected projects

---

#### Note



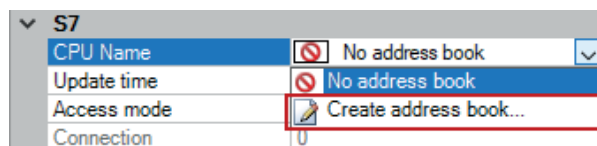
You can overwrite the entry in the *CPU Name* field. Thus, you can assign a unique name for the CPU that differs from that in the STEP 7 project. This is especially interesting when you use several STEP 7 projects in which the CPUs have the same name.

---

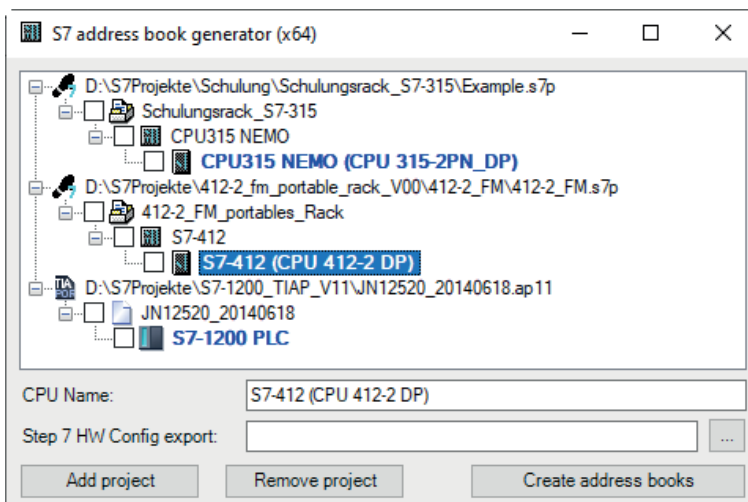


## Creating address books via the S7 address book generator

1. Open the S7 address book generator by one of the following ways:
  - On the *General* tab *Address books* node via the <Create address books> button.
  - In the module configuration on the *General* tab under *S7 – CPU Name*. Select *Create address book* in the drop-down menu.



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



### Note



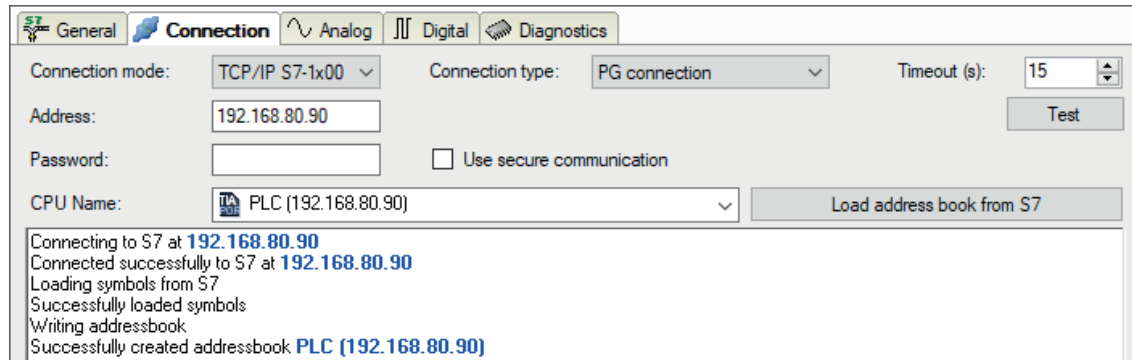
When address books of these projects are generated, TIA Portal projects must be compiled, saved and closed.

### 3.3.5.2 Creating address books online from S7-1200/1500 CPU

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

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

The CPU name of the address book is given automatically.



Address books that have been created online also contain operand type address information and hence can be used in connection mode TCP/IP as well. Just change the connection mode after creating the address book.

## 4 Request S7 variants

This chapter describes the Request S7 variants for applications with S7-1500 controllers. For applications with S7-300 and S7-400 controllers, refer to the manual Part 2.

S7 family	Firmware	Addressing type	Access to optimized data blocks	iba block family	iba data paths
S7-1500	Unlimited	Operands	No	ibaREQ	DP, PN, UDP, ibaNet-E
	From V3	Purely symbolic	Yes	ibaREQsym	PN, UDP, ibaNet-E

### 4.1 Request using ibaNet-E

The Request S7 variant for *ibaNet-E* is described below.

#### 4.1.1 General information

You can configure Request S7 for *ibaNet-E* in the following system configurations.

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

The prerequisite for use of Request S7 using *ibaNet-E* is a license for *ibaPDA-Interface-ibaNet-E* (item no. 31.001006).

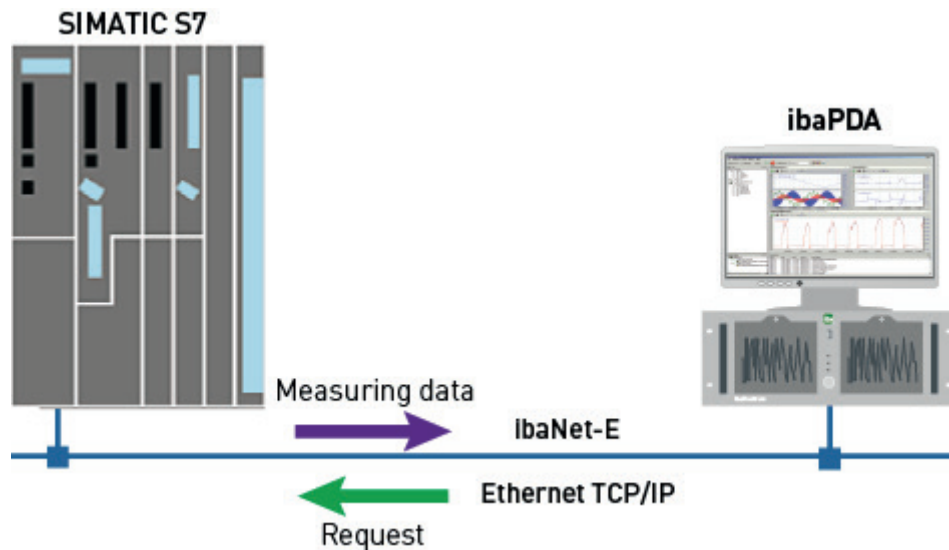
With the *ibaNet-E* Request, the measured values (Request Handshake) are requested via a separate TCP/IP connection .

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

- TCP/IP: The connection to the SIMATIC S7 is established 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 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 interfaces.
  - 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.

The following figure shows the system topology for an S7 Request using *ibaNet-E*.



#### Other documentation



Detailed information about *ibaPDA-interface-ibaNet-E* can be found in the interface manual.

### 4.1.2 SIMATIC S7-1500 configuration and engineering

The configuration and engineering on the SIMATIC S7 side with the SIMATIC TIA Portal is described below.

Carry out the following configuration and engineering steps on the SIMATIC TIA Portal side:

- Software engineering:  
Integration of the Request blocks into the S7 program
- Device configuration and engineering:  
Setting the CPU protection properties

#### 4.1.2.1 Configuration in STEP 7 using the ibaREQ iba block family

This section describes how to configure the Request blocks in TIA Portal STEP 7.

##### For each Request module

1. Copy the following blocks from the iba S7 library to the blocks folder of your STEP 7 project, see [iba S7 library](#), page 85. One set of Request blocks has to be called for each Request module (connection) in *ibaPDA*.

- ibaREQ\_M (FB1400), see ↗ *ibaREQ\_M (FB1400)*, page 69
- ibaREQ\_NetE-Buffer (FB1408), see ↗ *ibaREQ\_NetE-Buffer (FB1408)*, page 71
- ibaREQ\_NetE-Send (FB1409), see ↗ *ibaREQ\_NetE-Send (FB1409)*, page 72
- ibaREQ\_DB, ibaREQ\_DB-Interface (PLC data type)
- ibaREQ\_NetE-Bufferdata (PLC data type)

**Note**

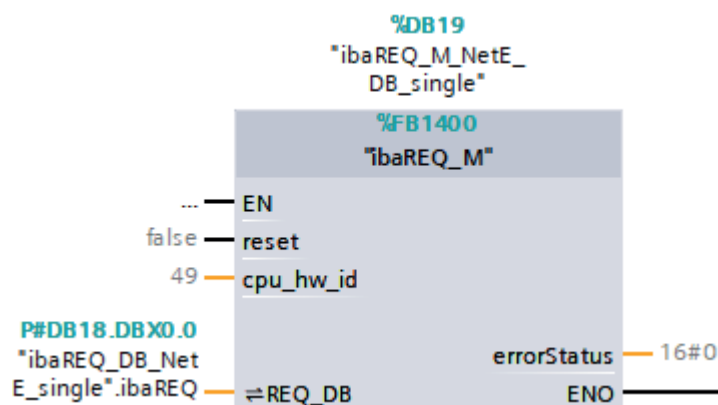
Only use Request blocks from the latest iba S7 library!

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

**Note**

The request blocks do not support multi-instance calls.

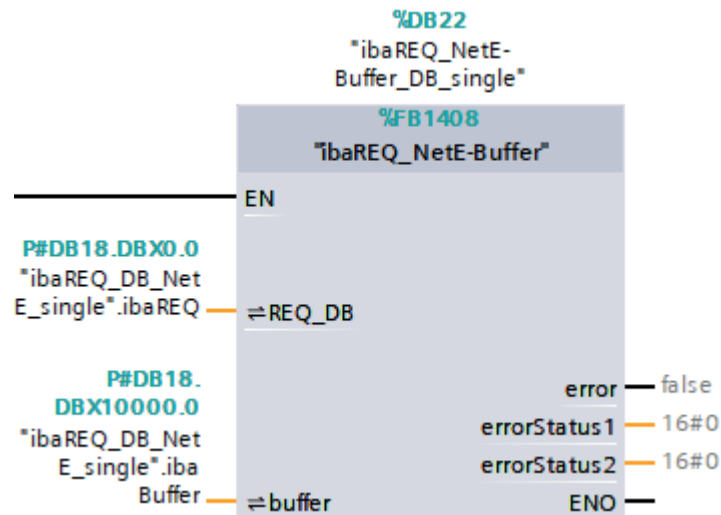
2. Call ibaREQ\_M (FB1400) preferably within OB1.



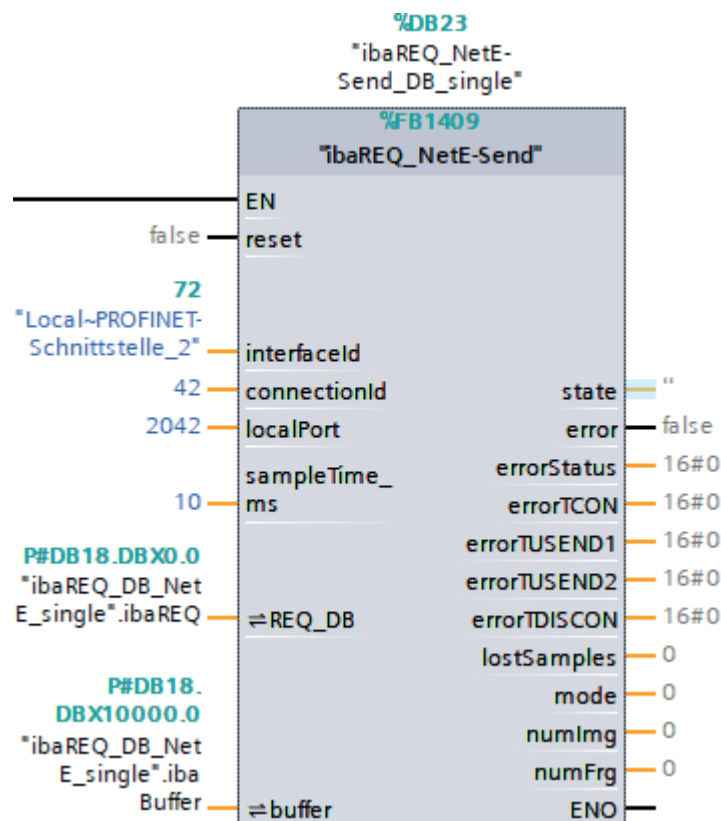
3. Enable the *Retain* option for the entire instance data block you have just created.

workTIARequestV18 > 20_testTIAReqSym [CPU 1518-4 PN/DP] > Program blocks > ibaREQ > ibaREQ_M_NetE_DB_single [DB19]										
ibaREQ_M_NetE_DB_single										
	Name	Data type	Offset	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervision
1	Input									
2	reset	Bool	0.0	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
3	cpu_hw_id	HW_IO	2.0	49	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
4	Output									
5	errorStatus	Word	4.0	16#0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
6	InOut									
7	REQ_DB	*ibaREQ_DB-Interfa...	6.0		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
8	Static									
9	EXPERT	Struct	12.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
10	sIDBinitialized	Bool	14.0	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
11	sOperandsInvalid	Bool	14.1	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
12	sIMDataValid	Bool	14.2	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
13	sOk	Bool	14.3	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
14	sGET_IM_DATA	Get_IM_Data			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
15	sIM_DATA	Array[0..53] of Byte	16.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
16	sAdrOPList	DInt	70.0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
17	cDBAddr_PDA2S7	Int	74.0	64	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
18	cDBAddr_S72PDA	Int	76.0	1440	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
19	cDBAddr_Xchange	Int	78.0	2464	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
20	cDBAddr_Ops	Int	80.0	2528	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
21	cDBAddr_actVal	Int	82.0	12768	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
22	cMaxDataLength	UInt	84.0	1472	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
23	sREQ_AnzOperandenl...	Int	86.0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
24	sREQ_AnzOperandenl...	Int	88.0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

4. Call ibaREQ\_NetE-Buffer (FB1408) within OB1 or a cyclic interrupt OB (OB3x).



5. Call ibaREQ\_NetE-Send (FB1409) preferably within OB1.



#### For each additional Request module

- In the blocks folder, an ibaREQ\_DB data block has to be available for each Request module. Copy the data block and assign a new unique DB number.
- Within OB1, ibaREQ\_M (FB1400) has to be called once more with the new DB number for each Request module.
- Within OB1 or a cyclic interrupt OB (Ob3x), ibaREQ\_NetE-Buffer (FB1408) has to be called once more with the new DB number for each Request module.

- Within OB1, ibaREQ\_NetE-Send (FB1409) has to be called once more with the new DB number for each Request module.
- Make sure that all instance data blocks are unique and that unique values are assigned for the connectionId parameters.

### Final

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

#### 4.1.2.2 Configuration in STEP 7 using the ibaREQsym iba block family

This section describes how to configure the Request blocks in TIA Portal STEP 7.

##### For each Request module

1. Copy the following blocks from the iba S7 library to the blocks folder of your STEP 7 project, see ↗ *iba S7 library*, page 85. One set of Request blocks has to be called for each Request module (connection) in *ibaPDA*.
  - ibaREQsym\_M, see ↗ *ibaREQsym\_M*, page 76
  - ibaREQsym\_NetE, see ↗ *ibaREQsym\_NetE*, page 78
  - ibaREQsym\_DB\_PDA
  - ibaREQsym-Interface (PLC data type)

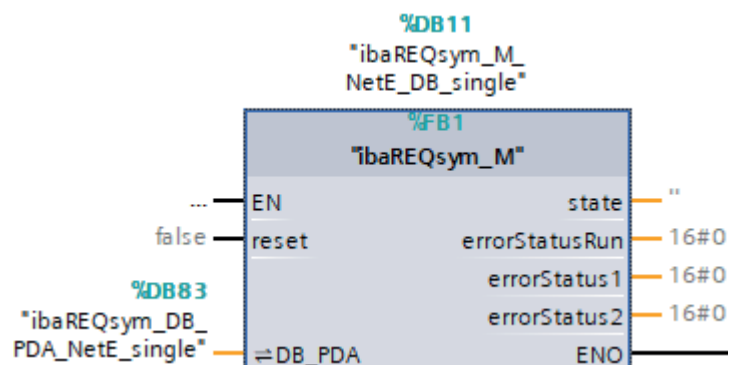
### Note



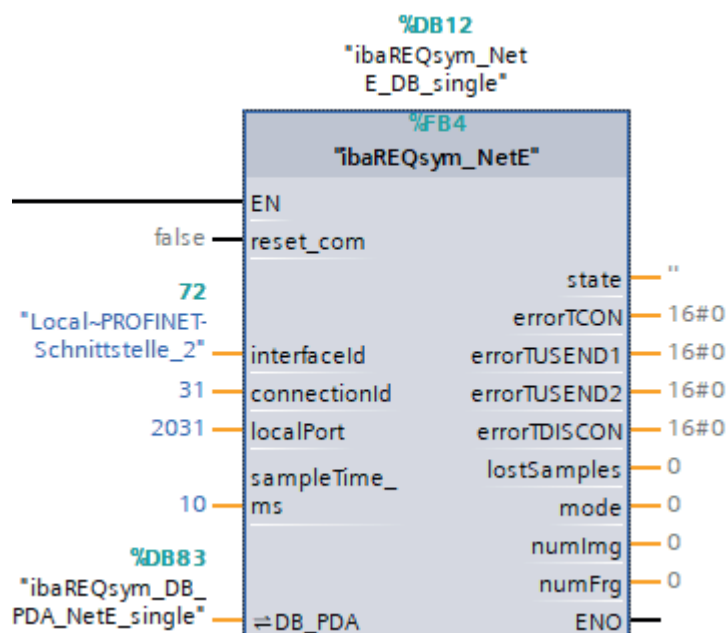
Only use Request blocks from the latest iba S7 library!

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

2. Call ibaREQsym\_M.



3. Call ibaREQsym\_NetE preferably within OB1.



#### For each additional Request module

- In the blocks folder, an ibaREQsym\_DB\_PDA data block has to be available for each Request module. Copy the data block and assign a new unique DB number.
- Within OB1 or a cyclic interrupt OB (OB3x), ibaREQsym\_M and ibaREQsym\_NetE have to be called once more with the new DB number for each Request module.
- Make sure that all instance data blocks are unique and that unique values are assigned for the connectionId parameters.

#### Final

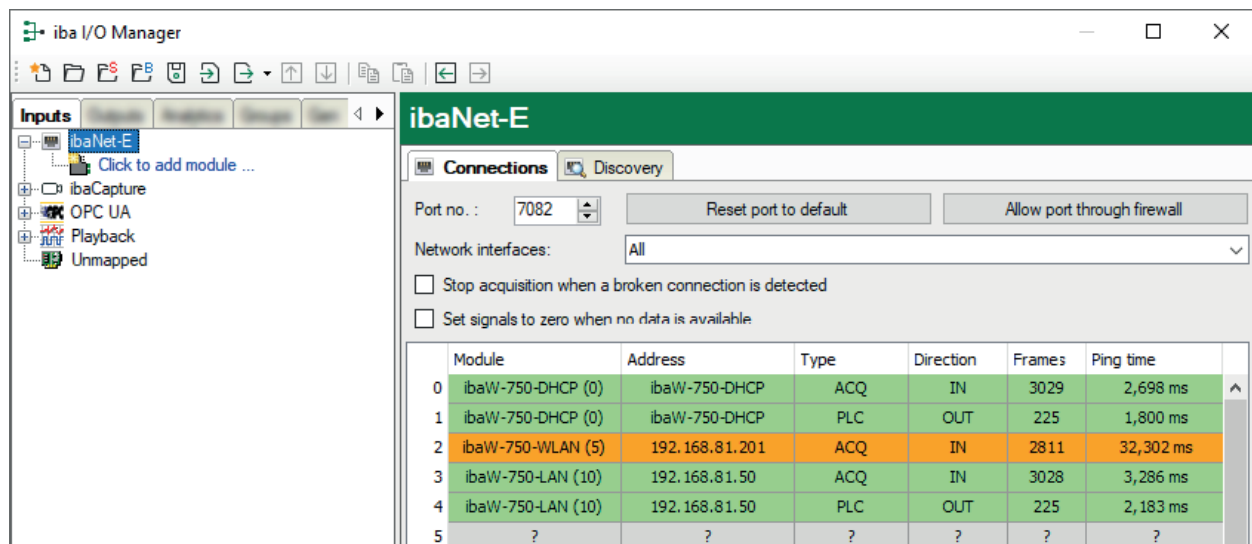
- Load all blocks to the S7 CPU and restart the S7 CPU.



### 4.1.3 Configuration and engineering ibaPDA

#### 4.1.3.1 General interface settings

If all system requirements are fulfilled, *ibaPDA* displays the *ibaNet-E* interface in the interface tree of the I/O Manager.



Module	Address	Type	Direction	Frames	Ping time
0 ibaW-750-DHCP (0)	ibaW-750-DHCP	ACQ	IN	3029	2,698 ms
1 ibaW-750-DHCP (0)	ibaW-750-DHCP	PLC	OUT	225	1,800 ms
2 ibaW-750-WLAN (5)	192.168.81.201	ACQ	IN	2811	32,302 ms
3 ibaW-750-LAN (10)	192.168.81.50	ACQ	IN	3028	3,286 ms
4 ibaW-750-LAN (10)	192.168.81.50	PLC	OUT	225	2,183 ms
5 ?	?	?	?	?	?

#### Other documentation



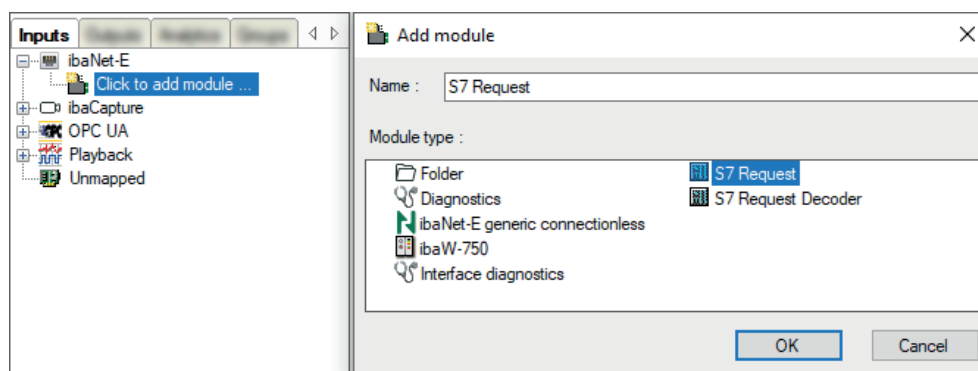
Detailed information about the *ibaNet-E* interface can be found in the respective manual.

#### Configuring ibaBM-PN

1. Start the *ibaPDA* client and open the I/O Manager.
2. Click on the blue command *Click to add module* under the interface.

Select a *Request S7* module in the dialog box and assign a name via the input field if required. Confirm the selection with <OK>.

- S7 Request (for acquiring analog and digital signals)
- S7 Request Decoder (for acquiring up to 32000 digital signals)



3. 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.
4. After you have finished the configuration, click <Apply> or <OK> to transfer the new configuration to the device and start data acquisition with *ibaPDA*.

#### 4.1.3.2 General module settings

You can find the description of all general settings that are identical for all Request-S7 modules in chapter [General module settings](#), page 15.

Information about the connection settings on the *Connection* tab can be found in chapter [Connection settings](#), page 16.

The *S7 Request* module for *ibaNet-E* has the following specific setting options in the *General* tab:

**S7 Request (9)**

General Connection Analog Digital Diagnostics

**Basic**

Module Type	S7 Request
Locked	False
Enabled	True
Name	S7 Request
Module No.	9
Timebase	10 ms
Use module name as prefix	False

**Module Layout**

No. analog signals	64
No. digital signals	64

**Connection**

Auto enable/disable	False
---------------------	-------

**ibaNet-E**

Sample time	10 ms
Samples per message	1
Drift compensation	True

**S7**

CPU Name	No address book
----------	-----------------

**Name**  
The name of the module.

[Select S7 operands](#)  
[Manage address books](#)

#### ibaNet-E

##### Sample time

Sample time on the controller side, corresponding to the cycle time of the function block requests.

##### Samples per message

Number of samples transmitted in an *ibaNet-E* frame

##### Drift compensation

Disable drift compensation if the data is sent less frequently than every 10 ms to improve processing of the received data.

## Connection

### Auto enable/disable

If this option is enabled and *ibaPDA* is unable to establish a connection to the S7 at the start of acquisition, *ibaPDA* disables this module and starts acquisition without the module. During acquisition, *ibaPDA* tries to connect to the S7. If successful, acquisition is restarted with the module enabled. If this option is disabled and *ibaPDA* is unable to establish a connection to the S7 at the start of acquisition, *ibaPDA* does not start acquisition.

### 4.1.3.3 S7 Request module

Configure a separate Request block call for each module.

For further information about module settings, see ↗ *General module settings*, page 15 and ↗ *General module settings*, page 42.

### 4.1.3.4 S7 Request Decoder module

The *S7 Request Decoder* module allows acquisition of up to 32000 digital signals sent as a maximum of 1000 double words (32 bit). This module type is well suited for applications where large numbers of digital signals have to be acquired and the max. 1024 directly addressable digital values are not sufficient.

Configure a separate Request block call for each module.

### General tab

For further information about module settings, see ↗ *General module settings*, page 15 and ↗ *General module settings*, page 42.

### Module-specific settings

#### Module structure – Number 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 table is adjusted accordingly.

### Connection configuration

The connection for the *S7 Request Decoder* module is configured in the same way as the connection for an S7 Request module, see ↗ *Connection settings*, page 16.

### Digital tab

The signals are configured in the same way as for the *ibaBM-DP* device, see ↗ *Digital tab*, page 64.

## 4.2 Request-S7 for ibaBM-PN

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

### 4.2.1 General information

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

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

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 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.
- 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.

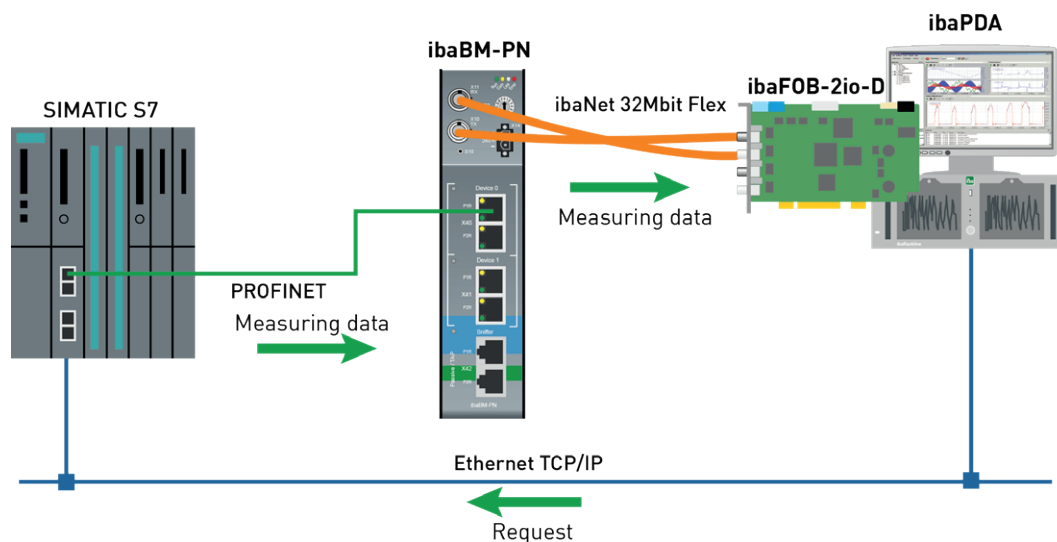
### System integration with ibaBM-PN

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 89.

## 4.2.2 Configuration and engineering SIMATIC S7-1500

This section describes the configuration and engineering on the SIMATIC S7 side with the SIMATIC TIA Portal.

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

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

### 4.2.2.1 Network configuration

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

Use the GSDML file [GSDML-Vx.yy-ibaBM-PN-yyyymmdd.xml](#).

#### Note

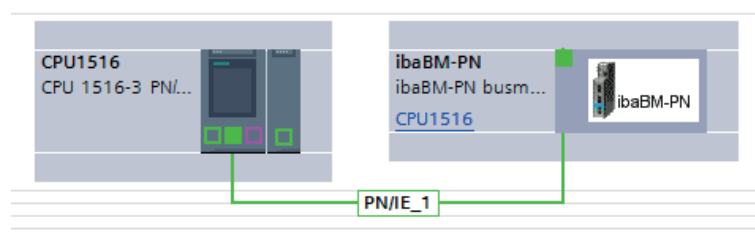


You find the latest version of the GSDML file on the data storage medium "iba Software & Manuals" in the following directory:

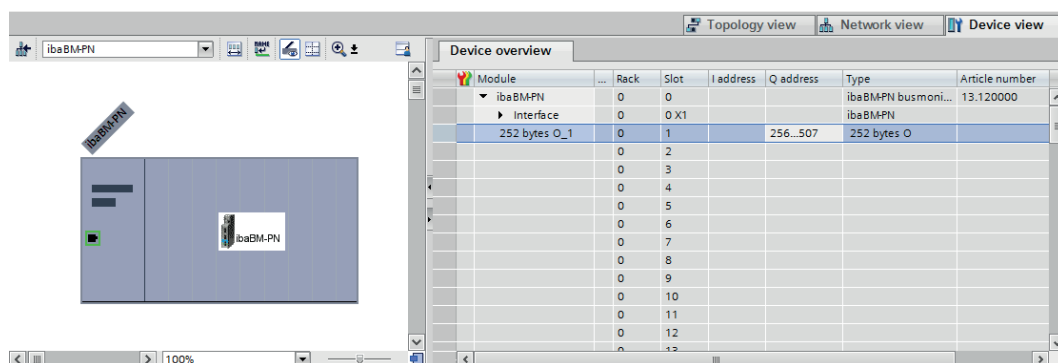
`\02_iba_Hardware\ibaBM-PN\01_GSD_Files\`

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

The following figure shows the net view.



The following figure shows the device view.



### 4.2.2.2 Configuring in STEP 7 with the iba block family ibaREQ

This section describes how to configure the Request blocks in TIA Portal STEP 7.

#### For each Request module

- Copy the following blocks from the iba S7 library to the blocks folder of your STEP 7 project, see [iba S7 library](#), page 85. One set of Request blocks has to be called for each Request module (connection) in *ibaPDA*.
  - ibaREQ\_M (FB1400), see [ibaREQ\\_M \(FB1400\)](#), page 69
  - ibaREQ\_PN (FB1401), see [ibaREQ\\_PN \(FB1401\)](#), page 70
  - ibaREQ\_DB (DB15)
  - ibaREQ\_DB-Interface (PLC data type)

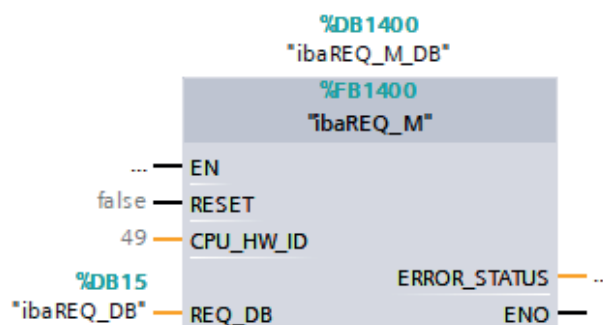
**Note**

Only use Request blocks from the latest iba S7 library!  
Request blocks in application examples can be outdated and, thus, cause errors.

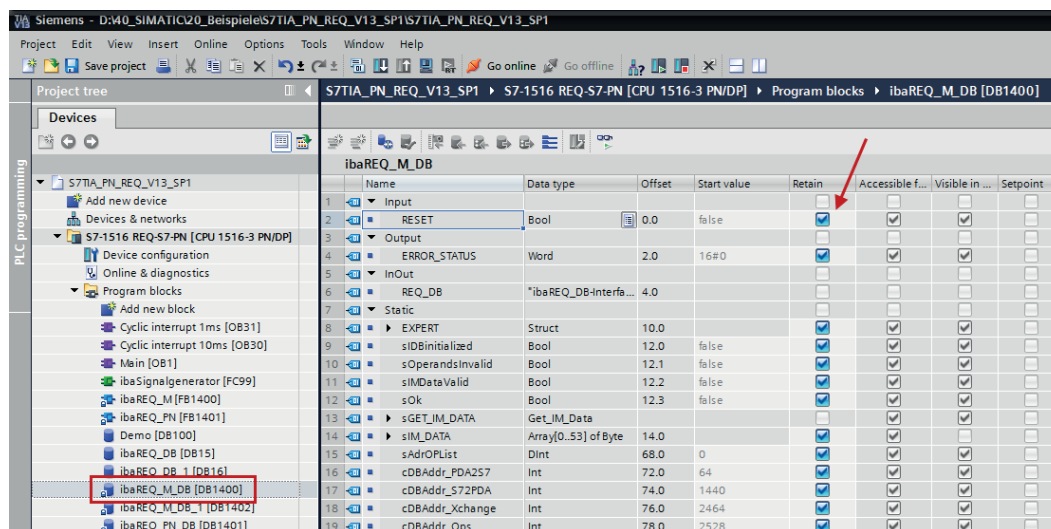
**Note**

The request blocks do not support multi-instance calls.

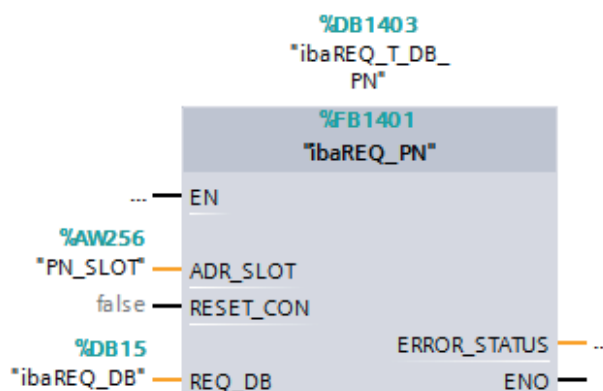
2. Call the ibaREQ\_M (FB1400) preferably within the OB1.



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



4. Call the ibaREQ\_PN (FB1401) preferably within the OB1 or a cyclic interrupt OB (OB3x).



### For each additional Request module

- In the blocks folder, an ibaREQ\_DB (DB15) data block has to be available for each Request module. Copy the data block and assign a new unique DB number.
- Within the OB1, ibaREQ\_M (FB1400) has to be called once more with the new DB number for each Request module.
- Within the OB1 or within a cyclic interrupt (OB3x), the ibaREQ\_PN (FB1401) has to be called once more with the new DB number for each Request module.
- Make sure that all instance data blocks are unique and that unique values are assigned for the ADR\_SLOT.

### Final

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

### 4.2.2.3 Configuration in STEP 7 using the ibaREQsym iba block family

This section describes how to configure the Request blocks in TIA Portal STEP 7.

#### For each Request module

1. Copy the following blocks from the iba S7 library to the blocks folder of your STEP 7 project, see [iba S7 library](#), page 85. One set of Request blocks has to be called for each Request module (connection) in *ibaPDA*.
  - ibaREQsym\_M, see [ibaREQsym\\_M](#), page 76
  - ibaREQsym\_PN, see [ibaREQsym\\_PN](#), page 77
  - ibaREQsym\_DB\_PDA
  - ibaREQsym\_Interface (PLC data type)

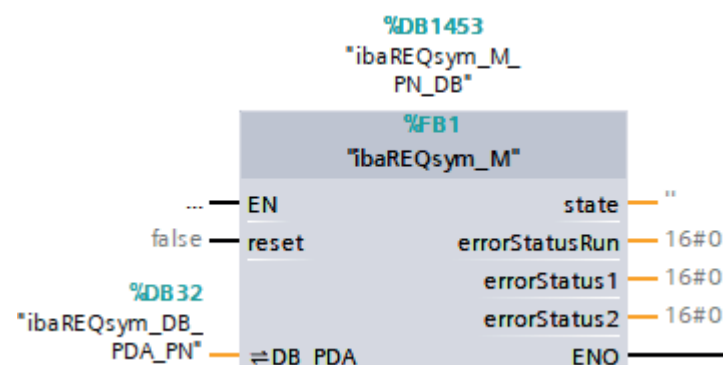
#### Note



Only use Request blocks from the latest iba S7 library!

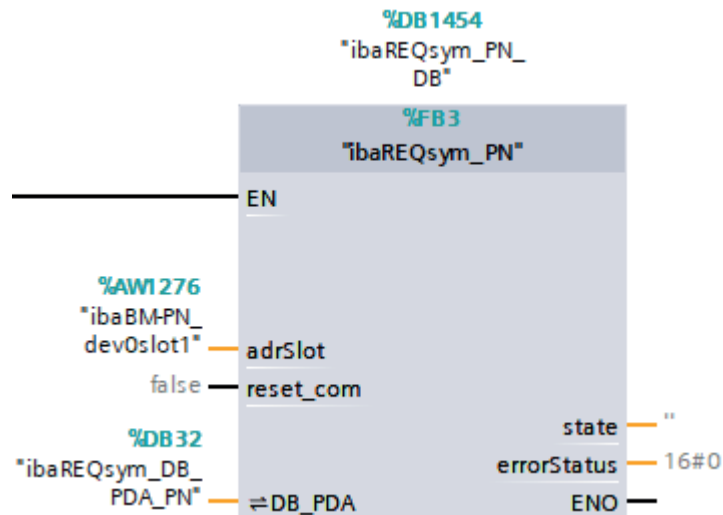
Request blocks in application examples can be outdated and, thus, cause errors.

2. Call ibaREQsym\_M.





3. Call ibaREQsym\_PN preferably within OB1.



#### For each additional Request module

- In the blocks folder, an ibaREQsym\_DB\_PDA data block has to be available for each Request module. Copy the data block and assign a new unambiguous DB number.
- Within OB1 or a cyclic interrupt OB (OB3x), ibaREQsym\_M and ibaREQsym\_PN have to be called once more with the new DB number for each Request module.
- Make sure that all instance data blocks are unique and that unique values are assigned for the ADR\_SLOT parameter.

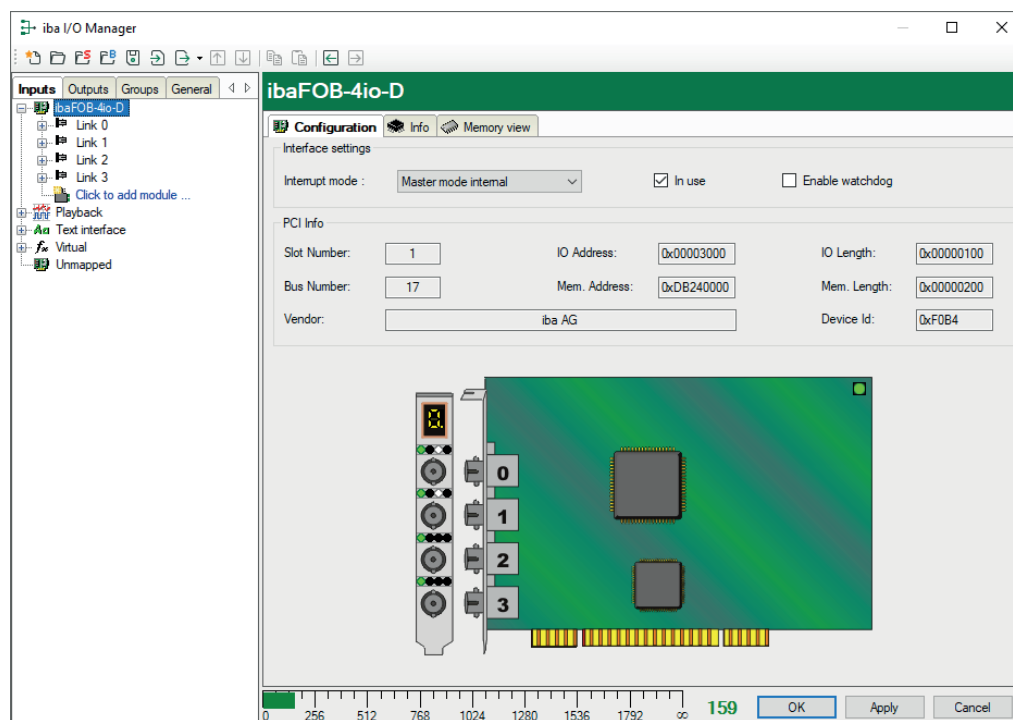
#### Final

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

## 4.2.3 Configuration and engineering ibaPDA

### 4.2.3.1 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.



### Other documentation

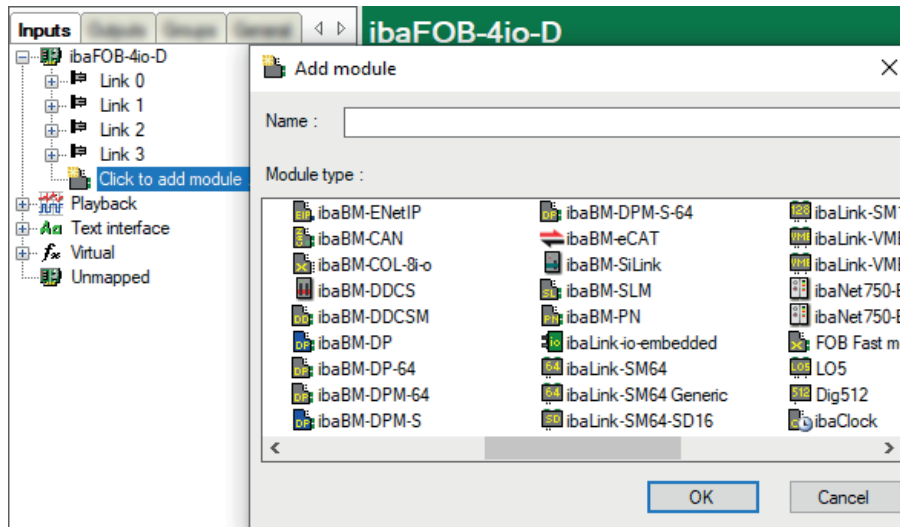


Detailed information about the *ibaFOB-D* card can be found in the respective manual.

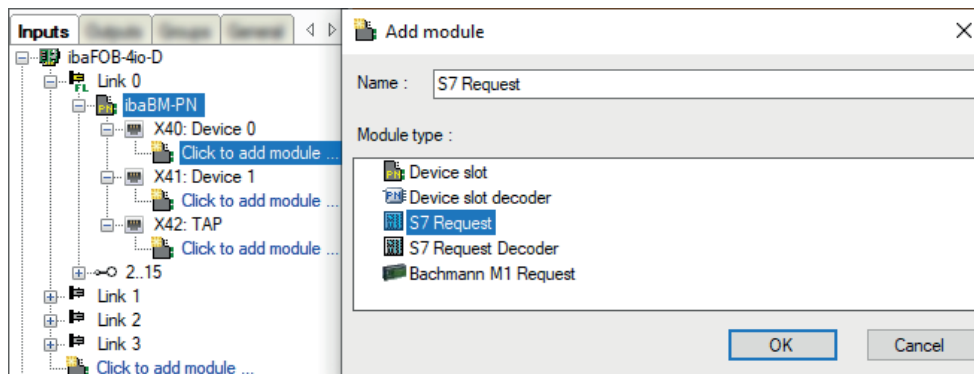
## Configuring ibaBM-PN

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-CPU's) to the *ibaBM-PN* module on the respective PROFINET connection Device 0 or Device 1. Available modules are:
  - S7 Request (for acquiring analog and digital signals)
  - S7 Request Decoder (for acquiring up to 1024 digital signals)



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

### 4.2.3.2 General module settings

You can find the description of all settings that are identical for all Request-S7 modules in chapter [➤ General module settings](#), page 15.

You can find information on connection settings from the *Connection* tab in chapter [➤ Connection settings](#), page 16.

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 54.

### 4.2.3.3 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 [➤ General module settings](#), page 15 and [➤ General module settings](#), page 52.

### 4.2.3.4 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 [➤ General module settings](#), page 15 and [➤ General module settings](#), page 52.

#### 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 ➔ *Connection settings*, page 16.

Digital tab

Configure the signals in the same way as for an *ibaBM-DP* device, see ➔ *Digital tab*, page 64.

4.2.4 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.

Inputs

ibaDAQ-S

Fiber optic link

ibaBM-PN

X40: Device 0

Click to add module ...

X41: Device 1

1: S7 Req SINUMERIK PN (3)

Click to add module ...

X42: TAP

Click to add module ...

2..15

Link 1

Link 2

ibaBM-PN

General

Analog

Digital

Diagnostics

	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.

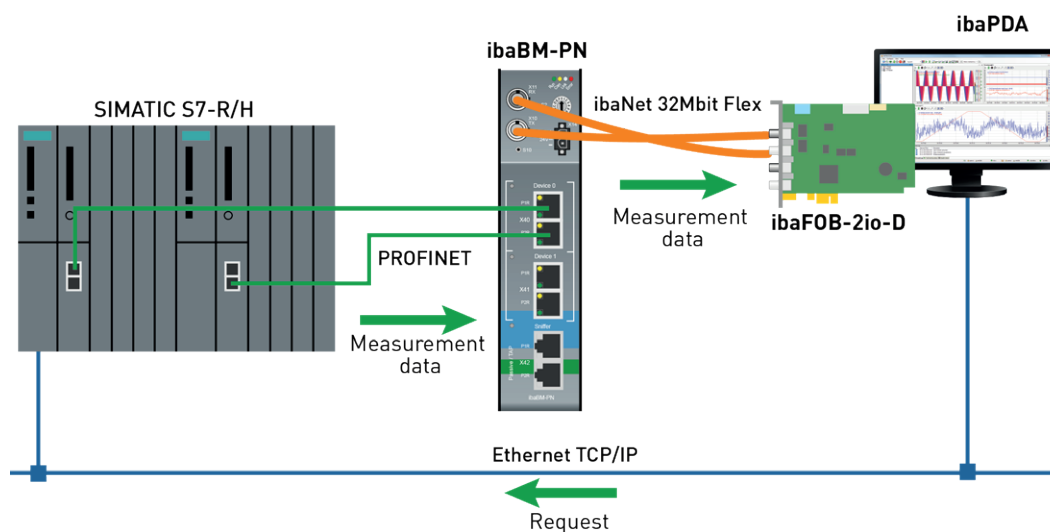
## 4.3 Request-S7 for ibaBM-PN in redundancy mode

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

### 4.3.1 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 in the device manual.

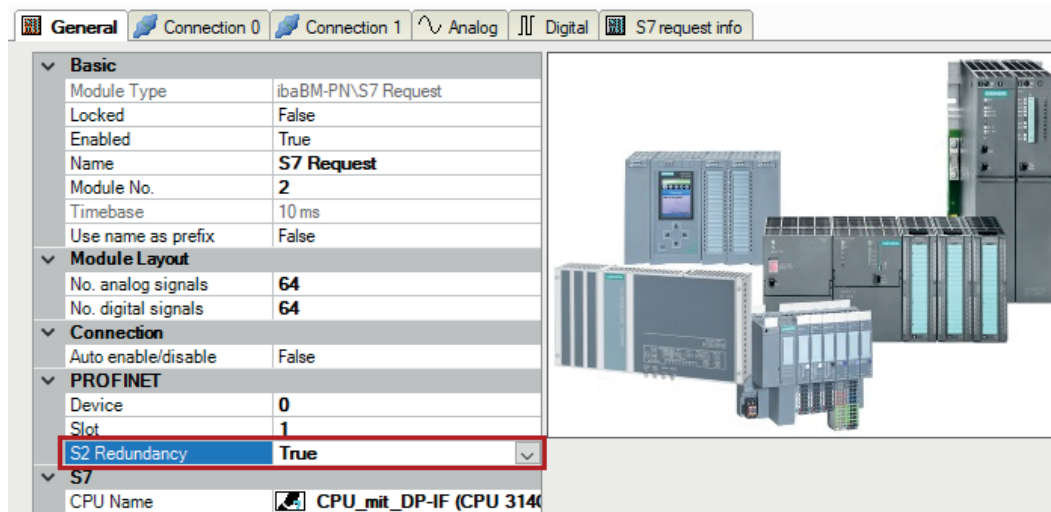
The functionality of the Request-S7 for *ibaBM-PN* in redundancy mode mostly corresponds to the functionality in standard mode (see chapter [Request-S7 for ibaBM-PN](#), page 44). The differences and extensions are described in the following.

### 4.3.2 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.

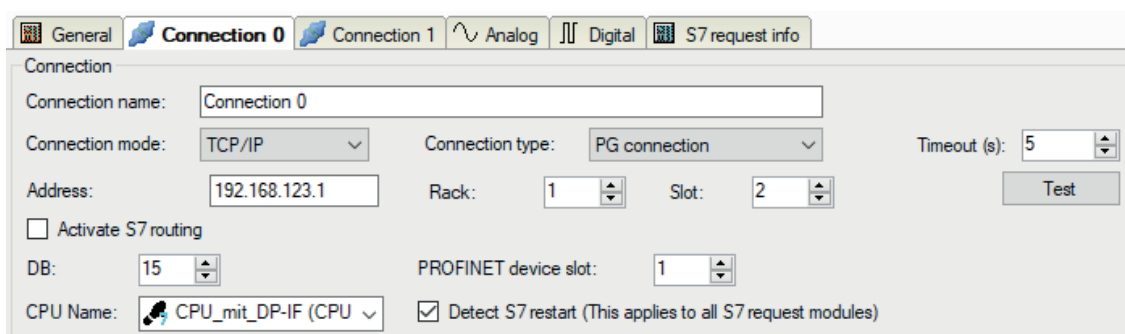


The configuration of the modules otherwise corresponds to the configuration in standard mode, see [General module settings](#), page 52.

#### 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.



The other settings are identical to the settings for operation without redundancy mode, see [Connection settings](#), page 16.

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.

## 4.4 Request-S7 for ibaBM-DP

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

### 4.4.1 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.



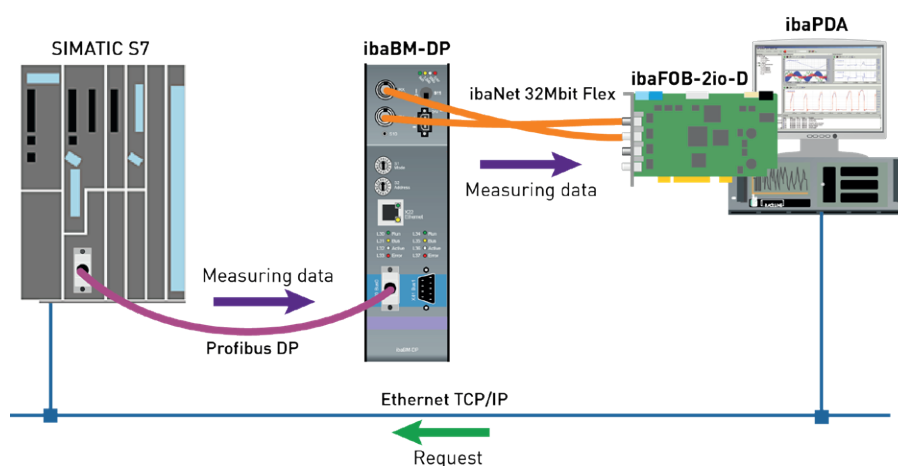
### System integration with ibaBM-DP

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 Manual part 2.

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

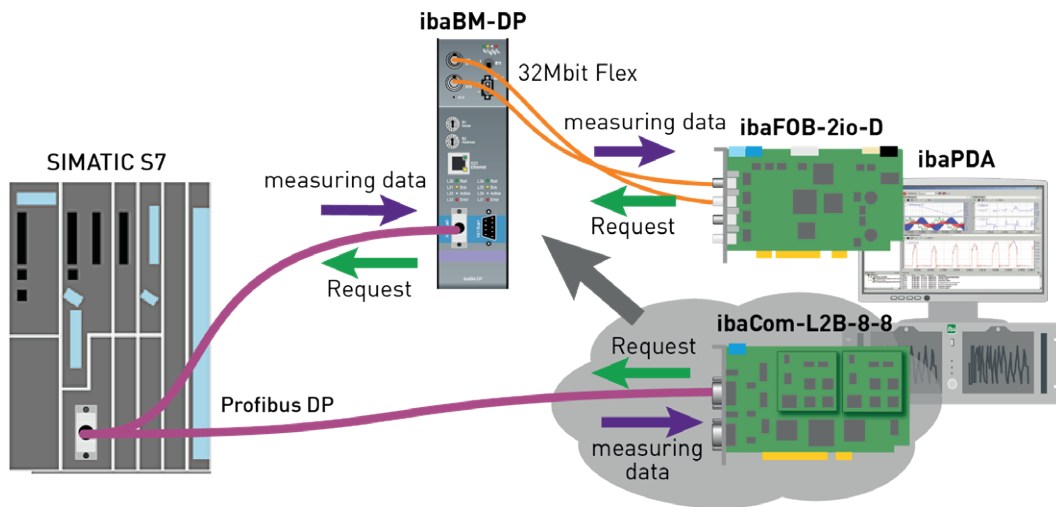


Detailed information about *ibaBM-DP* can be found in the device manual.

For information about application examples, see chapter ➤ *Application examples*, page 89.

#### 4.4.1.1 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).

Detailed information about this subject can be found see Manual part 2.

#### Note



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

#### 4.4.2 Configuration and engineering SIMATIC S7-1500

This section describes the configuration and engineering on the SIMATIC S7 side with the SIMATIC TIA Portal.

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

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

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

#### 4.4.2.1 Network configuration

Specify an iba-PROFIBUS slave for each module.

Use the GSD file [ibaDPMSi.gsd](#) (version V2.2 and higher).

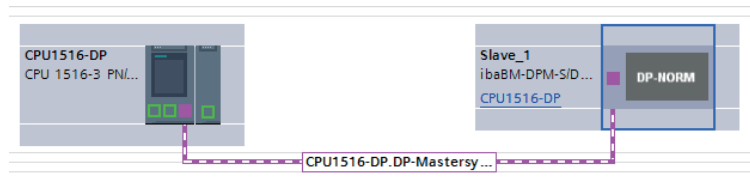
##### Note



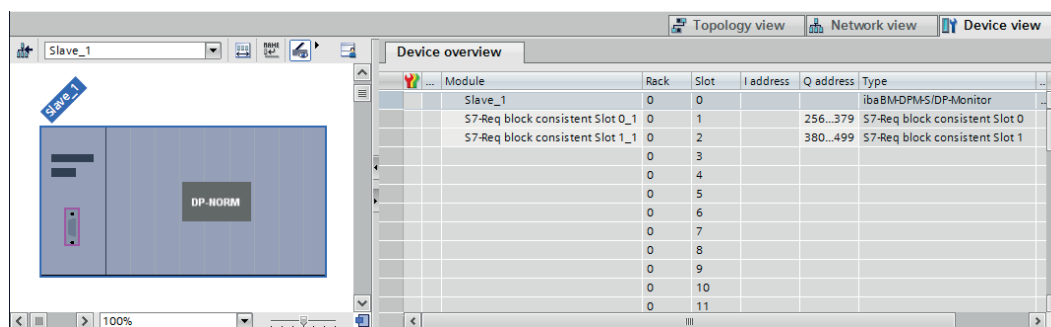
You find the GSD file [ibaDPMSi.gsd](#) on the data storage medium "iba Software & Manuals" in the following directory:

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

The following figure shows the net view.



The following figure shows the device view.



#### 4.4.2.2 Configuration in STEP 7 using the ibaREQ iba block family

This section describes how to configure the Request blocks in TIA Portal STEP 7.

##### For each Request module

- Copy the following blocks from the iba S7 library to the blocks folder of your STEP 7 project, see [iba S7 library](#), page 85. One set of Request blocks has to be called for each Request module (connection) in *ibaPDA*.
  - ibaREQ\_M (FB1400), see [ibaREQ\\_M \(FB1400\)](#), page 69
  - ibaREQ\_DP (FB1402), see [ibaREQ\\_DP \(FB1402\)](#), page 71
  - ibaREQ\_DB (DB15)
  - ibaREQ\_DB-Interface (PLC data type)

##### Note



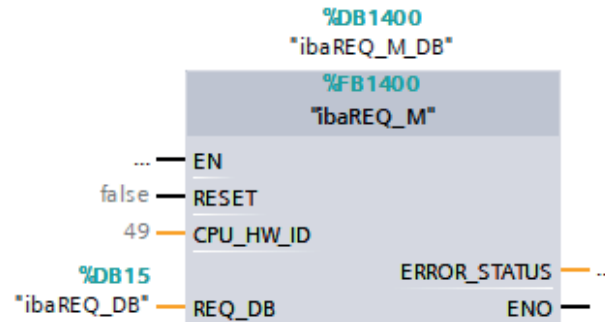
Only use Request blocks from the latest iba S7 library!

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

**Note**

The request blocks do not support multi-instance calls.

2. Call ibaREQ\_M (FB1400) preferably within OB1.

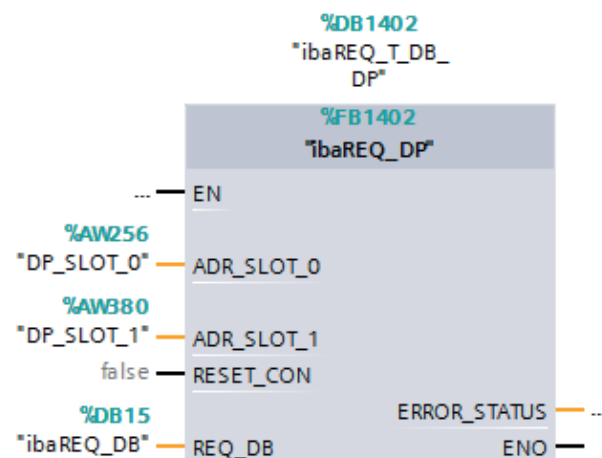


3. Enable the *Retain* option for the entire instance data block you have just created.

The screenshot shows the **Properties** window for the **ibaREQ\_M\_DB** data block. The **Retain** checkbox is checked, indicating that the data is retained across power cycles. A red arrow points to the **Retain** checkbox.

Name	Data type	Offset	Start value	Retain	Accessible f...	Visible in ...	Setpoint
1. Input							
2. RESET	Bool	0.0	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3. Output							
4. ERROR_STATUS	Word	2.0	16#0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5. InOut							
6. REQ_DB	"ibaREQ_DB-interfa...	4.0		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7. Static							
8. EXPERT	Struct	10.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9. sIDinitialized	Bool	12.0	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
10. sOperandsInvalid	Bool	12.1	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
11. sIMDataValid	Bool	12.2	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
12. sOk	Bool	12.3	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
13. sGET_IM_DATA	Get_IM_Data			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
14. sIM_DATA	Array(0..53) of Byte	14.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
15. sAdrOPList	Dint	68.0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
16. cDBAddr_PDA257	Int	72.0	64	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
17. cDBAddr_S72PDA	Int	74.0	1440	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
18. cDBAddr_Xchange	Int	76.0	2464	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
19. cDBAddr_Ops	Int	78.0	2528	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

4. Call ibaREQ\_DP (FB1402) within 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 OB1, ibaREQ\_M (FB1400) has to be called once more with the new DB numbers for each Request module.
- Within OB1 or a cyclic interrupt OB (OB3x), ibaREQ\_DP (FB1402) has to be called once more with the new DB numbers for each Request module.
- Make sure that all instance data blocks are unique and that unique values are assigned for the ADR\_SLOT\_0 and ADR\_SLOT\_1 parameters.

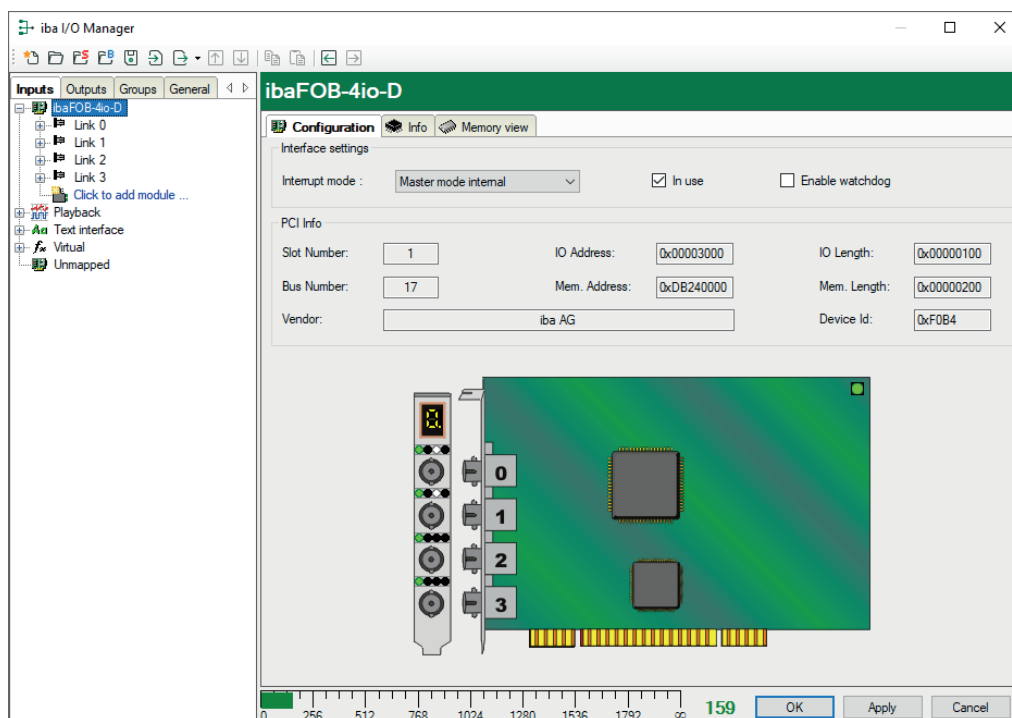
### Final

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

## 4.4.3 Configuration and engineering ibaPDA

### 4.4.3.1 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.



### Other documentation

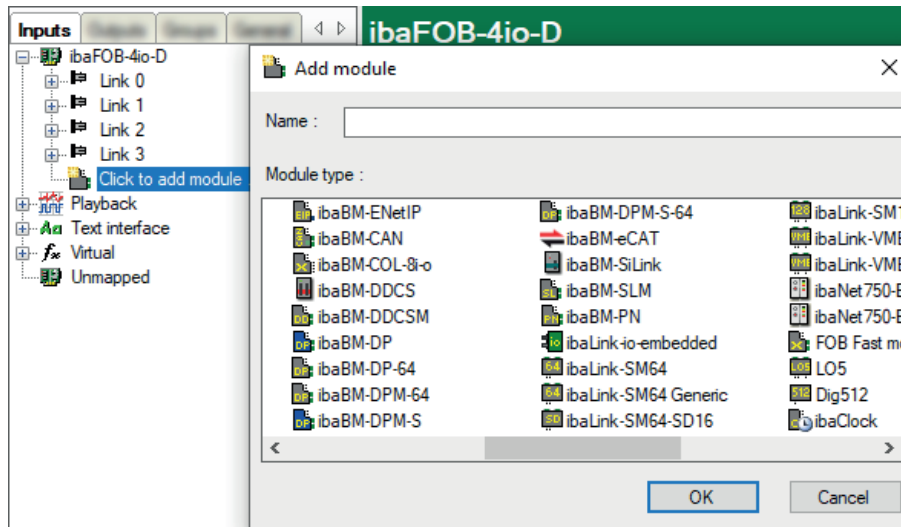


Detailed information about the *ibaFOB-D* card can be found in the respective manual.

## Configuring ibaBM-DP

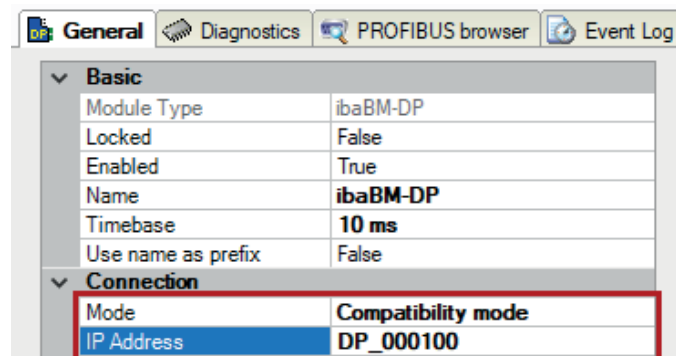
1. Start the *ibaPDA* client and open the I/O Manager.
2. In the interface tree (left), select the link of the *ibaFOB* card to which *ibaBM-DP* is connected.

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>.



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".



### Note

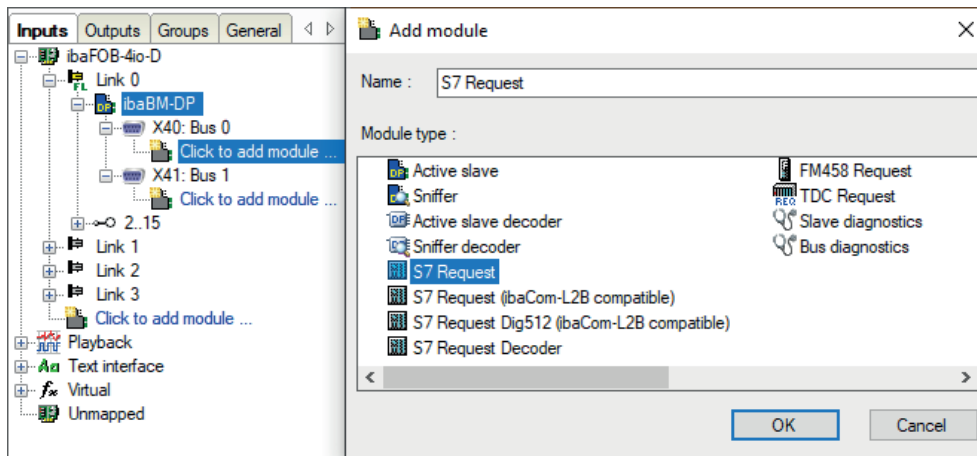


An online connection to the device via Ethernet or USB is mandatory in compatibility mode. The manual of the *ibaBM-DP* device explains how to establish and check this connection.

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-CPU's) to the *ibaBM-DP* module on the respective PROFIBUS connection Bus 0 or Bus 1. Available modules are:

- S7 request  
(for acquiring analog and digital signals)
- S7 request decoder  
(for the acquisition of up to 1024 digital signals)



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

#### 4.4.3.2 General module settings

You can find the description of all settings that are identical for all Request-S7 modules in chapter [General module settings](#), page 15.

You can find information on connection settings from the *Connection* tab in chapter [Connection settings](#), page 16.

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.

### 4.4.3.3 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 chapter ↗ *General module settings*, page 15 and ↗ *General module settings*, page 63.

### 4.4.3.4 Modules S7 Request Decoder

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

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 15 and ↗ *General module settings*, page 63.

#### Connection configuration

Configure the connection of the *S7 Request Decoder* module in the same way as the connection for an S7-Request module, see ↗ *Connection settings*, page 16.

#### 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 ↗ *Selection via the symbolic operand addresses*, page 26. 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	<input checked="" type="checkbox"/>
1 + DB 3.DBW 2	DB 3.DBW 2	WORD	<input checked="" type="checkbox"/>
2 + DB 3.DBW 4	DB 3.DBW 4	WORD	<input checked="" type="checkbox"/>
Name			
Digital Signal 0			<input checked="" type="checkbox"/>
Digital Signal 1			<input checked="" type="checkbox"/>
Digital Signal 2			<input checked="" type="checkbox"/>
Digital Signal 3			<input checked="" type="checkbox"/>
Digital Signal 4			<input checked="" type="checkbox"/>
Digital Signal 5			<input checked="" type="checkbox"/>
Digital Signal 6			<input checked="" type="checkbox"/>
Digital Signal 7			<input checked="" type="checkbox"/>
Digital Signal 8			<input checked="" type="checkbox"/>
Digital Signal 9			<input checked="" type="checkbox"/>
Digital Signal 10			<input checked="" type="checkbox"/>
Digital Signal 11			<input checked="" type="checkbox"/>
Digital Signal 12			<input checked="" type="checkbox"/>
Digital Signal 13			<input checked="" type="checkbox"/>
Digital Signal 14			<input checked="" type="checkbox"/>
Digital Signal 15			<input checked="" type="checkbox"/>
3 + DB 3.DBW 6	DB 3.DBW 6	WORD	<input checked="" type="checkbox"/>
4 +		WORD	<input type="checkbox"/>

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.

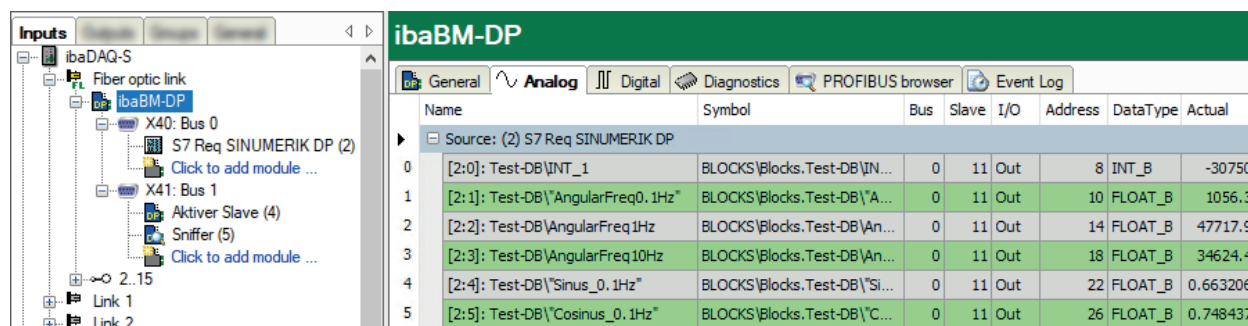
### Note



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

#### 4.4.4 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.

## 5 Description of Request blocks

### 5.1 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 85.

#### Note



Only use Request blocks from the latest iba S7 library!

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

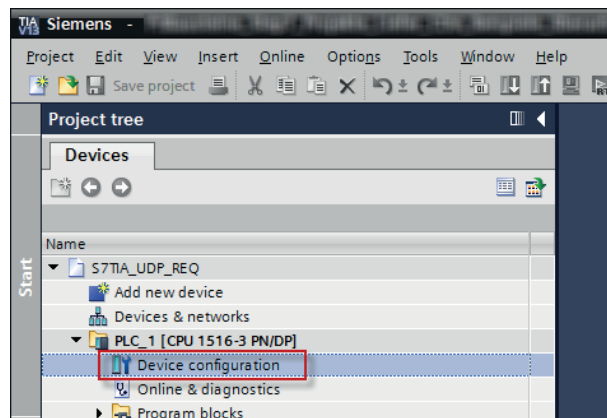
Use different Request block combinations depending on the current system configuration:

Request block	S7-1500 CPU with integrated PN interface	S7-1500 CPU with CM1542-1	Recommended call level
ibaREQ_M (FB1400)	X	X	OB1
ibaREQ_PN (FB1401)	X	X	OB1 or OB3x
ibaREQ_DP (FB1402)	X	X	OB1 or OB3x
ibaREQ_DB (DB15)	X	X	-
ibaREQ_DB-Interface	X	X	-
ibaREQ_NetE-Buffer (FB1408)	X	X	OB1 or OB3x
ibaREQ_NetE-Send (FB1409)	X	X	OB1 or OB3x

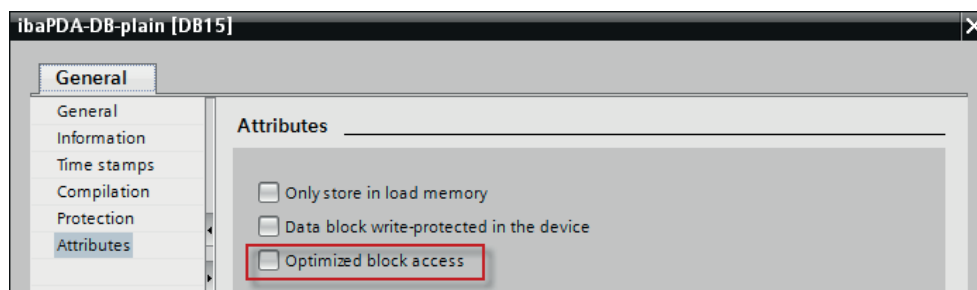
- **ibaREQ\_M (Management)**  
The block realizes the communication with *ibaPDA*. Ideally, the block is called in OB1.
- **ibaREQ\_PN/ibaREQ\_DP (provision and sending of current signal values)**  
The block provides the current signal values using the transmission cycle. Ideally, the block is called in a cyclic interrupt OB.
- **ibaREQ\_DB (interface DB)**  
This DB acts as an interface to *ibaPDA* and between the different Request blocks.
- **ibaREQ\_NetE-Buffer**  
The block provides the current signal values using the transmission cycle and buffers them. Ideally, the block is called in a cyclic interrupt OB.
- **ibaREQ\_NetE-Send**  
The block sends the buffered signal values. Ideally, the block is called in a cyclic interrupt OB.

### 5.1.1 Device configuration

Make the following setting in CPU device configuration:



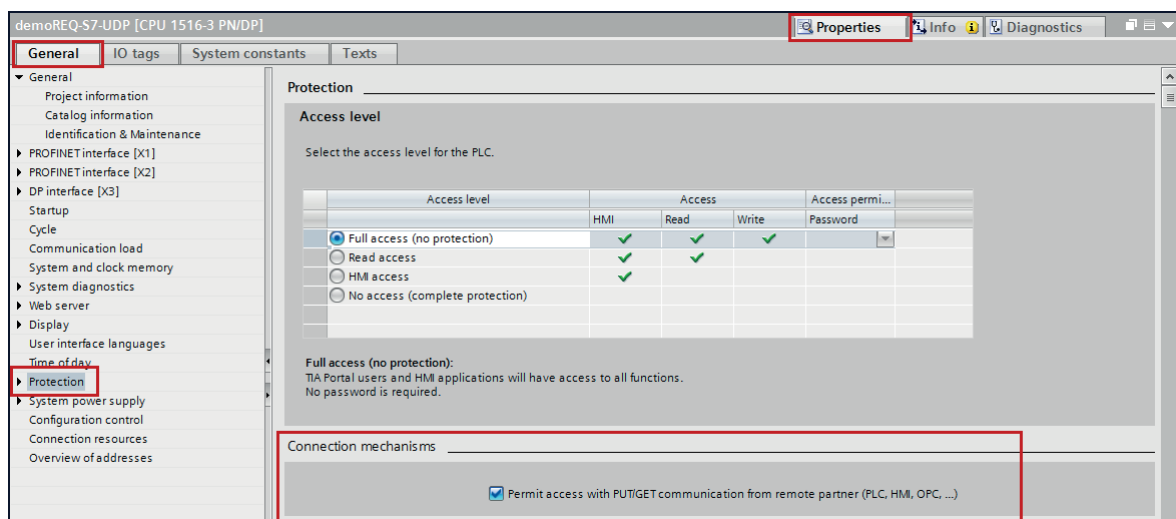
Under *ibaREQ\_DB (DB15)* block properties – *Attributes* disable the *Optimized block access* option.



#### S7-1500 CPUs with TCP/IP connection mode (not TCP/IP S7-1x00)

Up to TIA Portal V18, make the following settings in the TIA Portal:

In the CPU properties (*Properties – General – Protection – Connection mechanisms*), activate the access via PUT/GET communication.



For the S7-1200, this option is only available firmware V4.0 or higher.

As of TIA Portal V19 and CPU firmware V3.1 (V4.7 for S7-1200), activate the PUT/GET access for S7-1500 and S7-1200 CPUs as follows:

1. In the project navigation, navigate to *Security settings – Users and roles – Roles* tab.
2. Add a new role and enter a name, e.g. "Put/Get".
3. In the *Runtime rights* tab, select your PLC under *Function rights categories*.
4. Under *Function rights*, activate the access level *HMI access*.
5. Under *Users and Roles*, go to the *Users* tab.
6. Activate the user "Anonymous".
7. Confirm the following message with <OK>.
8. Under *Assigned roles*, assign the newly created role to the anonymous user.
9. In the *Device view*, open the properties of the CPU.
10. In the section navigation, navigate to *Protection & Security – Connection mechanisms*.
11. Activate the option *Permit access with PUT/GET communication from remote partner*.
12. Save and compile the configuration and load the changes into the CPU.

You can find further information in the SiePortal under

<https://support.industry.siemens.com/cs/ww/en/view/109925755>.

### S7-1500 access protection

You can enable access protection on S7-1500 CPUs. There is the following dependency with *ibaPDA*:

Access level	CPU access	<i>ibaPDA</i> reads symbolic from CPU	S7 access
Full access (no protection)	HMI, read, write	OK	OK
Read access	HMI, read	OK	OK
HMI access	HMI	No	OK
No access (complete protection)		No	No

## 5.1.2 ibaREQ\_M (FB1400)

### Description of formal parameters

Name	Type	Data Type	Description
REQ_DB	IN	DB_ANY	DB of the <i>ibaPDA</i> communication interface <i>ibaREQ_DB</i>
RESET	IN	BOOL	FALSE: No reset (default) TRUE: Block reset
CPU_HW_ID	IN	HW_IO	Hardware ID of local CPU
ERROR_STATUS	OUT	WORD	Error code

The following SIMATIC standard block is used internally:

- GET\_IM\_DATA (FB801)

### Detailed description

#### REQ\_DB

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

The length of the data block is fixed.

#### 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.

#### CPU\_HW\_ID

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

#### ERROR\_STATUS

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

For a list of all possible error codes, see chapter [↗ Error codes of Request blocks](#), page 100.

### 5.1.3 ibaREQ\_PN (FB1401)

#### Description of formal parameters

Name	Type	Data Type	Description
ADR_SLOT	IN	VARIANT	Start address of the output range
RESET_CON	IN	BOOL	FALSE: No reset (default) TRUE: Reset of the communication connection
REQ_DB	INOUT	UDT	DB of the <i>ibaPDA</i> communication interface ibaREQ_DB
ERROR_STATUS	OUT	WORD	Internal error code

### Detailed description

#### ADR\_SLOT

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

#### 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 100.

### 5.1.4 ibaREQ\_DP (FB1402)

#### Description of 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 <i>ibaPDA</i> 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 100.

### 5.1.5 ibaREQ\_NetE-Buffer (FB1408)

#### Description of formal parameters

Name	Type	Data Type	Description
REQ_DB	INOUT	UDT	Pointer to the communication data range <i>ibaREQ_DB-Interface</i> in ibaREQ_DB
buffer	INOUT	UDT	Pointer to the buffer data range <i>ibaREQ_NetE-Bufferdata</i> in ibaREQ_DB
error	OUT	BOOL	Internal error indicator
errorStatus1	OUT	WORD	Internal error code
errorStatus2	OUT	WORD	Internal error code

## Detailed description

### REQ\_DB

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

### buffer

Pointer to the buffer data range. This range is used to temporarily store the buffered signal values. For all related Request blocks, configure the identical DB.

### error

Internal error indicator for the block. If there is no error, the value FALSE is output.

### errorStatus1

Internal error code for the block. If there is no error, the value 0 is output.

### errorStatus2

Internal error code for the block. If there is no error, the value 0 is output.

For a list of all possible error codes, see [➤ Error codes of Request blocks](#), page 100.

## 5.1.6 ibaREQ\_NetE-Send (FB1409)

### Description of formal parameters

Name	Type	Data Type	Description
reset	IN	BOOL	FALSE: No reset (default) TRUE: Block reset
interfaceId	IN	HW_ANY	HW ID of the interface used
connectionId	IN	CONN_OUC	Unique connection ID of the send block (TSEND_C)
localPort	IN	UINT	Local port number
sampleTime_ms	IN	UINT	Signal sample time in milliseconds
REQ_DB	INOUT	UDT	Pointer to the communication data range ibaREQ_DB-Interface in ibaREQ_DB
buffer	INOUT	UDT	Pointer to the buffer data range ibaREQ_NetE-Bufferdata in ibaREQ_DB
state	OUT	STRING[16]	Block status
error	OUT	BOOL	Internal error indicator
errorStatus	OUT	WORD	Internal error code
errorStatusTCON	OUT	WORD	Error code for the internally called TCON block
errorStatusTUSEND1	OUT	WORD	Error code for the internally called 1st TUSEND block
errorStatusTUSEND2	OUT	WORD	Error code for the internally called 2nd TUSEND block
errorStatusTDISCON	OUT	WORD	Error code for the internally called TDISCON block



Name	Type	Data Type	Description
lostSamples	OUT	UNIT	Counter for lost measured values
mode	OUT	UNIT	Detected ibaNet-E mode
numImg	OUT	UNIT	Detected number of images per frame
numFrg	OUT	UNIT	Detected number of fragments per frame

### Detailed description

#### reset

Used to manually reset the Request block.

#### interfaceId

HW ID of the interface used

#### connectionId

Unambiguous reference to the connection to be established, value range: 1 to 4095.

#### localPort

Number of the local port used

#### sampleTime\_ms

Signal sample time in milliseconds

#### REQ\_DB

Pointer to the communication data range. This range is used for the data exchange with *ibaPDA*. For all related Request blocks, configure the identical DB.

#### buffer

Pointer to the buffer data range. This range is used to temporarily store the buffered signal values. For all related Request blocks, configure the identical DB.

#### state

Block status in plain text

#### error

Internal error indicator for the block. If there is no error, the value FALSE is output.

#### errorStatus

Internal error code for the block. If there is no error, the value 0 is output.

For a list of all possible error codes, see ➤ *Error codes of Request blocks*, page 100.

#### errorStatusTCON

Error code for the internally called TCON block.

#### errorStatusTUSEND1

Error code for the internally called 1st TUSEND block.

#### errorStatusTUSEND2

Error code for the internally called 2nd TUSEND block.

#### errorStatusTDISCON

Error code for the internally called TDISCON block.

A list of all possible error codes for the TCON, TUSEND, TDISCON system blocks can be found in the Siemens documentation.

**lostSamples**

The counter is incremented each time a block is called if a new UDP telegram cannot be sent to *ibaPDA* because the preceding send job has not yet been completed. A constantly increasing value indicates a bottleneck in communication performance.

**mode**

Detected ibaNet-E mode.

0: invalid, 1: single image, 2: multi image, 3: fragmented image

**numImg**

Detected number of images per frame

**numFrg**

Detected number of fragments per frame

## 5.2 ibaREQsym iba block family

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

The ibaREQsym block family allows access to both optimized and non-optimized data blocks. Addressing is done by using only the symbol names.

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 85.

### 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	S7-1500 CPU with integrated PN interface	S7-1500 CPU with CM1542-1	Recommended call level
ibaREQsym_M	X	X	OB1 or OB3x <sup>2)</sup>
ibaREQsym_NetE	X	X	
ibaREQsym_PN	X	X	
ibaREQsym_DB_PDA	X	X	-
ibaREQsym-Interface	X	X	-

#### ■ ibaREQsym\_M (Management)

The block realizes the communication with *ibaPDA*. Ideally, the block is called in OB1.

#### ■ ibaREQsym\_NetE (buffering and sending of current signal values)

The block buffers the current signal values and sends them to *ibaPDA*.

#### ■ ibaREQsym\_PN (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.

#### ■ ibaREQsym\_DB\_PDA (DB interfaces)

This DB acts as an interface to *ibaPDA* and between the different Request blocks.

<sup>2)</sup> You must use the blocks at the same call level.

### 5.2.1 ibaREQsym\_M

#### Description of formal parameters

Name	Type	Data Type	Description
reset	IN	BOOL	FALSE: No reset (default) TRUE: Block reset
DB_PDA	INOUT	UDT	DB for the <i>ibaPDA</i> communication interface ibaREQ_DB
state	OUT	STRING[16]	Block status
errorStatusRun	OUT		Internal error code
errorStatus1	OUT		Internal error code
errorStatus2	OUT		Error code for internally called blocks

#### Detailed description

##### reset

Used to manually reset the block

##### DB\_PDA

Pointer to the communication data range. This range is used for the data exchange with *ibaPDA*. For all related Request blocks, configure the identical DB.

##### state

Block status in plain text

##### errorStatusRun

Internal error code for the block. If there is no error, the value 0 is output.

##### errorStatus1

Internal error code for the block. If there is no error, the value 0 is output.

##### errorStatus2

Internal error code for the block. If there is no error, the value 0 is output.

For a list of all possible error codes, see ➤ *Error codes of Request blocks*, page 100.

## 5.2.2 ibaREQsym\_PN

### Description of formal parameters

Name	Type	Data Type	Description
adrSlot	IN	VARIANT	Starting address of the output range
reset_com	IN	BOOL	FALSE: No reset (default) TRUE: Reset of the communication connection
DB_PDA	INOUT	UDT	DB for the <i>ibaPDA</i> communication interface ibaREQ_DB
state	OUT	STRING[16]	Block status
errorStatus	OUT	WORD	Internal error code

### Detailed description

#### adrSlot

Starting address of the slot used in the output range of the process image.

#### reset\_com

Used for manually resetting the communication connection.

#### DB\_PDA

Pointer to the communication data range. This range is used for the data exchange with *ibaPDA*. For all related Request blocks, configure the identical DB.

#### state

Block status in plain text

#### errorStatus

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 100.

### 5.2.3 ibaREQsym\_NetE

#### Description of formal parameters

Name	Type	Data Type	Description
reset	IN	BOOL	FALSE: No reset (default) TRUE: Block reset
interfaceId	IN	HW_ANY	HW ID of the interface used
connectionId	IN	CONN_OUC	Unique connection ID of the send block (TSEND_C)
localPort	IN	UINT	Local port number
sampleTime_ms	IN	UINT	Signal sample time in milliseconds
DB_PDA	INOUT	UDT	DB for the <i>ibaPDA</i> communication interface ibaREQ_DB
state	OUT	STRING[16]	Block status
errorStatusTCON	OUT	WORD	Error code for the internally called TCON block
errorStatusTUSEND1	OUT	WORD	Error code for the internally called 1st TUSEND block
errorStatusTUSEND2	OUT	WORD	Error code for the internally called 2nd TUSEND block
errorStatusTDISCON	OUT	WORD	Error code for the internally called TDISCON block
lostSamples	OUT	UNIT	Counter for lost measured values
mode	OUT	UNIT	Detected ibaNet-E mode
numImg	OUT	UNIT	Detected number of images per frame
numFrg	OUT	UNIT	Detected number of fragments per frame

#### Detailed description

##### reset

Used to manually reset the Request block.

##### interfaceId

HW ID of the interface used

##### connectionId

Unambiguous reference to the connection to be established, value range: 1 to 4095.

##### localPort

Number of the local port used

##### sampleTime\_ms

Signal sample time in milliseconds

##### DB\_PDA

Pointer to the communication data range. This range is used for the data exchange with *ibaPDA*. For all related Request blocks, configure the identical DB.

**state**

Block status in plain text

**errorStatusTCON**

Error code for the internally called TCON block.

**errorStatusTUSEND1**

Error code for the internally called 1st TUSEND block.

**errorStatusTUSEND2**

Error code for the internally called 2nd TUSEND block.

**errorStatusTDISCON**

Error code for the internally called TDISCON block.

A list of all possible error codes for the TCON, TUSEND, TDISCON system blocks can be found in the Siemens documentation.

**lostSamples**

The counter is incremented each time a block is called if a new UDP telegram cannot be sent to *ibaPDA* because the preceding send job has not yet been completed. A constantly increasing value indicates a bottleneck in communication performance.

**mode**

Detected ibaNet-E mode.

0: invalid, 1: single image, 2: multi image, 3: fragmented image

**numImg**

Detected number of images per frame

**numFrg**

Detected number of fragments per frame

## 6 Diagnostics

### 6.1 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.

The screenshot shows a 'License information' window. On the left, under 'License information', there are fields for: License container (3-4), Customer name (i...), License time limit (Unlimited), Container type (WIBU CmStick v4.40), Container host (...), Required EUP date (01.02.2023), and EUP date (31.12.2025). On the right, under 'Licenses:', a list of licenses is shown. One license, 'ibaPDA-Interface-Codesys-Xplorer (16)', is highlighted with a red rectangular box.

### 6.2 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

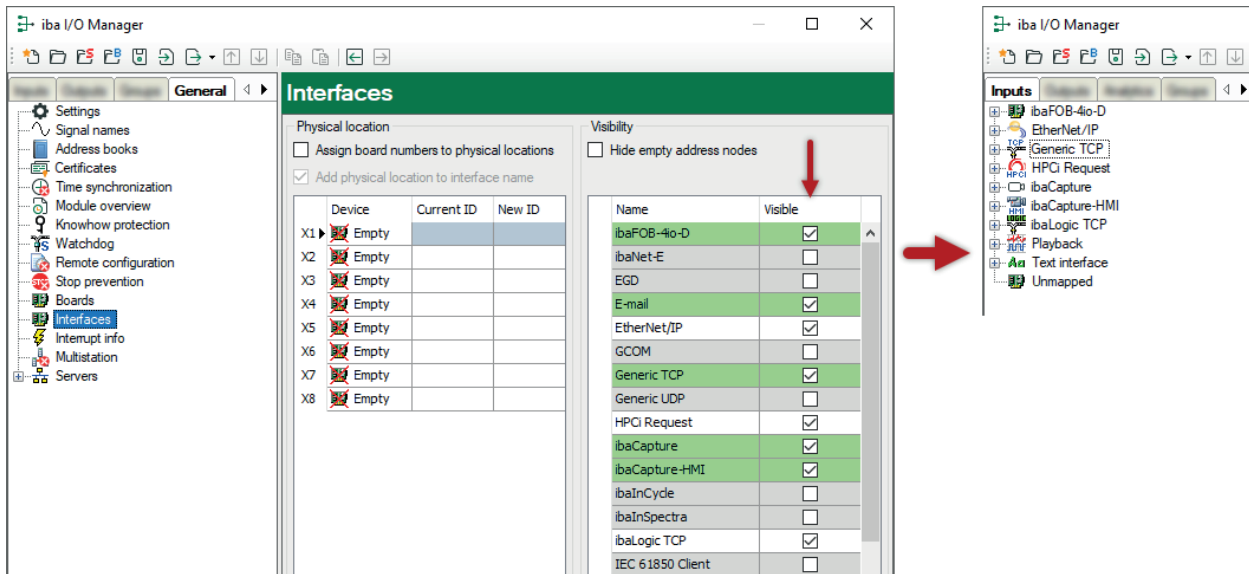
The table *Visibility* lists all the interfaces that are available either through licenses or installed cards. These interfaces can also be viewed in the interface tree.

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:





## 6.3 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.

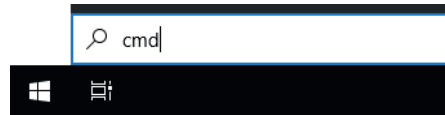
Examples:

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

## 6.4 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.

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>ping 192.168.1.10

Pinging 192.168.1.10 with 32 bytes of data:
Reply from 192.168.1.10: bytes=32 time=1ms TTL=30
Reply from 192.168.1.10: bytes=32 time<1ms TTL=30
Reply from 192.168.1.10: bytes=32 time<1ms TTL=30
Reply from 192.168.1.10: bytes=32 time<1ms TTL=30

Ping statistics for 192.168.1.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\Windows\system32>
```

→ With no existing connection you receive error messages.

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>ping 192.168.1.10

Pinging 192.168.1.10 with 32 bytes of data:
Reply from 192.168.1.10: Destination host unreachable.
Reply from 192.168.1.10: Destination host unreachable.
Reply from 192.168.1.10: Destination host unreachable.
Reply from 192.168.1.10: Destination host unreachable.

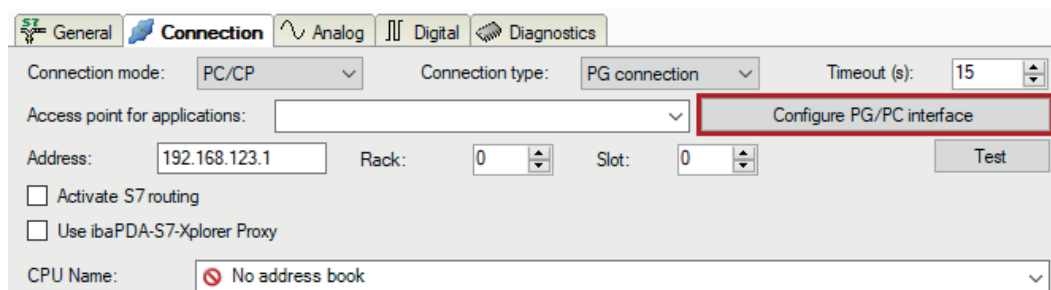
Ping statistics for 192.168.1.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

C:\Windows\system32>
```

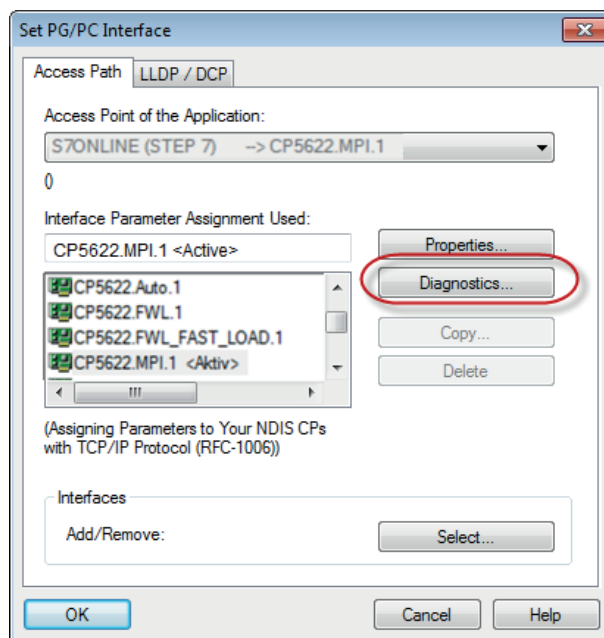
## 6.5 Connection diagnostics with PG/PC interface

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.



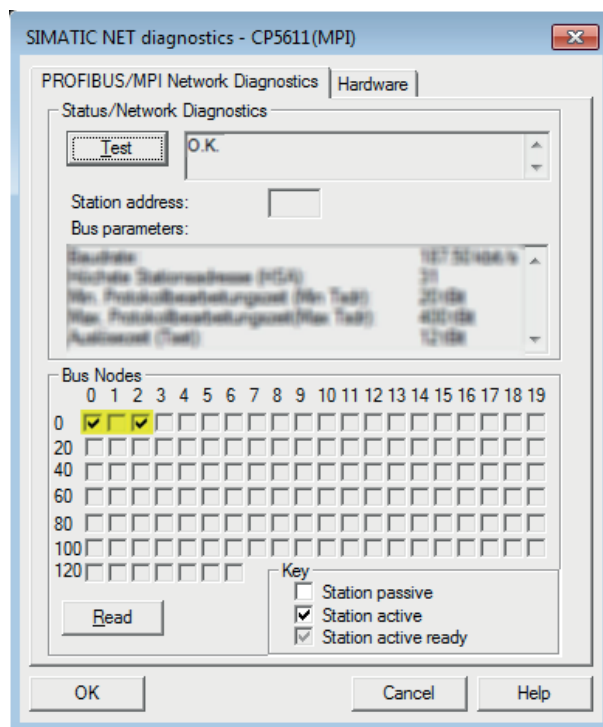
2. Open the diagnostics dialog with the <Diagnostics> button.



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.



## 6.6 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 53

See chapter Request-S7 for ibaBM-DP ➔ *Diagnostics*, page 66

See chapter Request-S7 for ibaBM-DPM-S in part 2 of this manual

See chapter Request-S7 for ibaCom-L2B in part 2 of this manual

## 7 Appendix

### 7.1 iba S7 library

The iba S7 library is available in two versions:

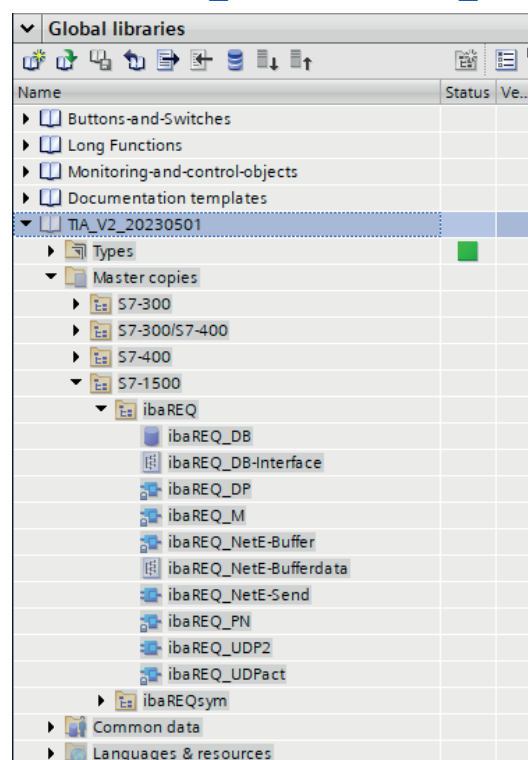
- SIMATIC Manager : STEP 7 ≥ V5.5
- SIMATIC TIA Portal STEP 7 ≥ V16

#### 7.1.1 iba S7 library for SIMATIC TIA portal

The iba S7 library for SIMATIC TIA Portal ("ibaS7LibTIA\_Vx\_y") is suitable for SIMATIC TIA Portal. 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\`



#### Note



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

The following components are included:

**Target platform S7-300, S7-400, WinAC**

iba connection	Block name	Block no.	Note
ibaBM-DP	ibaDP_Req	FC122	
	ibaDP_DB_PDA	DB10	
	ibaDP_DB_work	DB25	
ibaBM-PN	ibaREQ_M	FB140	
	ibaREQ_PN	FB141	
	ibaREQ_PNdev	FB150	
	ibaREQ_DB	DB15	
	ibaUDT_UDPact	UDT145	
ibaCom-L2B	ibaL2B_Init	FC111	
	ibaL2B_Req	FC112	
	ibaL2B_Req_CP	FC113	only necessary when using a CP342-5 instead of the FC112
	ibaL2B_DB_work	DB22	
	ibaL2B_DB_Struct	UDT22	
	ibaL2B_CP_SNDRCV	DB10	only necessary when using a CP342-5
ibaPDA-Inter- face-S7-TCP/ UDP	ibaREQ_M	FB140	
	ibaREQ_UDPact	FB145	
	ibaREQ_UDPint	FB146	
	ibaREQ_UDPext3	FB147	
	ibaREQ_UDPext4	FB148	
	ibaREQ_DB	DB15	
	ibaUDT_UDPact	UDT145	

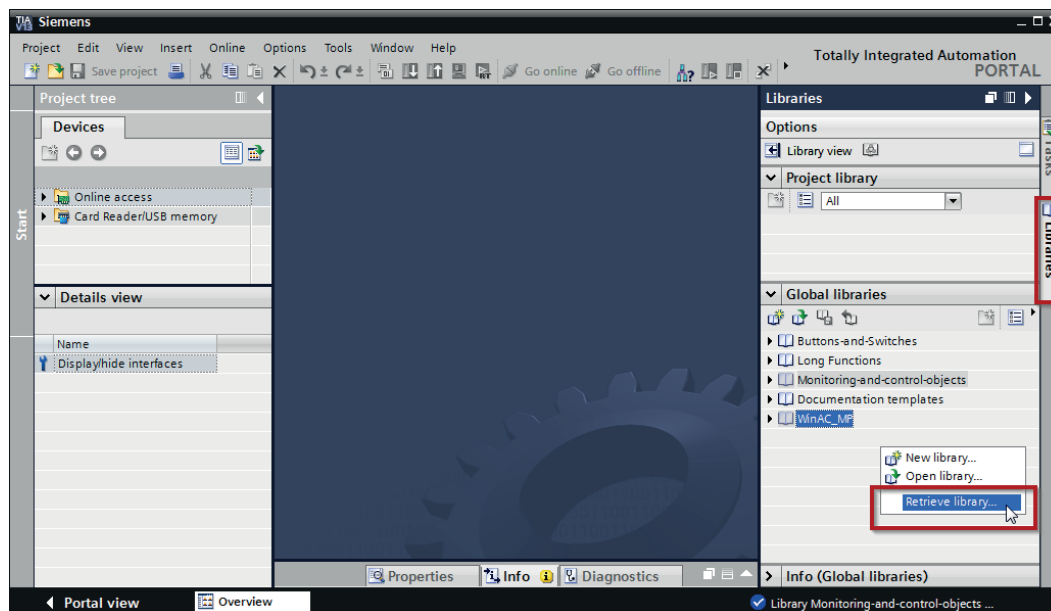
**Target platform S7-1500**

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

### 7.1.1.1 Integrating the library into TIA Portal

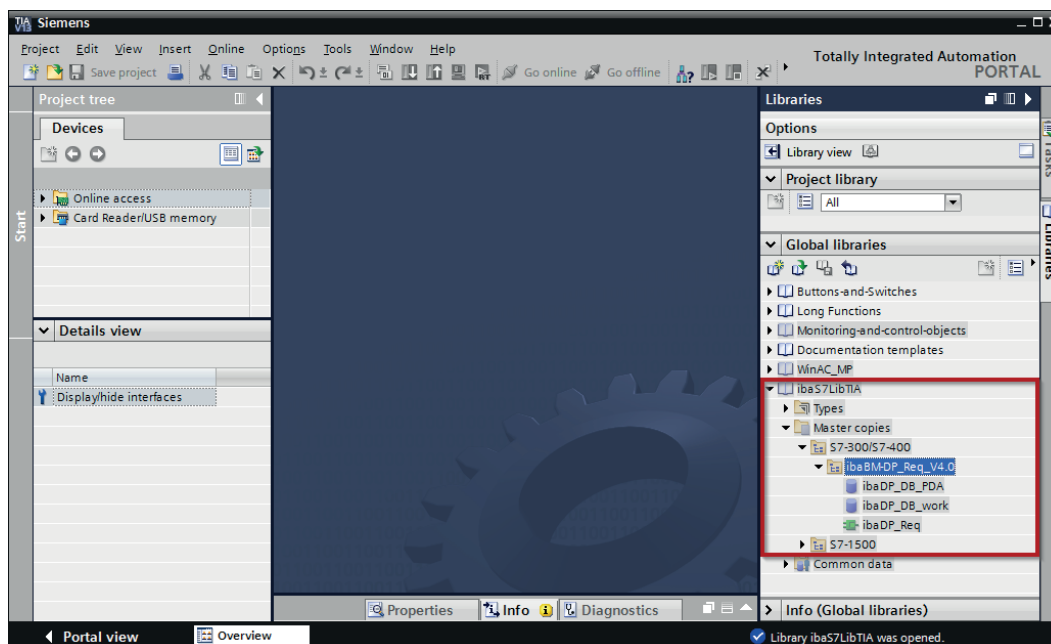
For integrating the library, you have to retrieve it in the TIA portal. Copy the iba S7 library to a local directory of your computer, where the TIA Portal is executed.

1. On the *Libraries* tab, select *Retrieve library...* from the context menu.



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

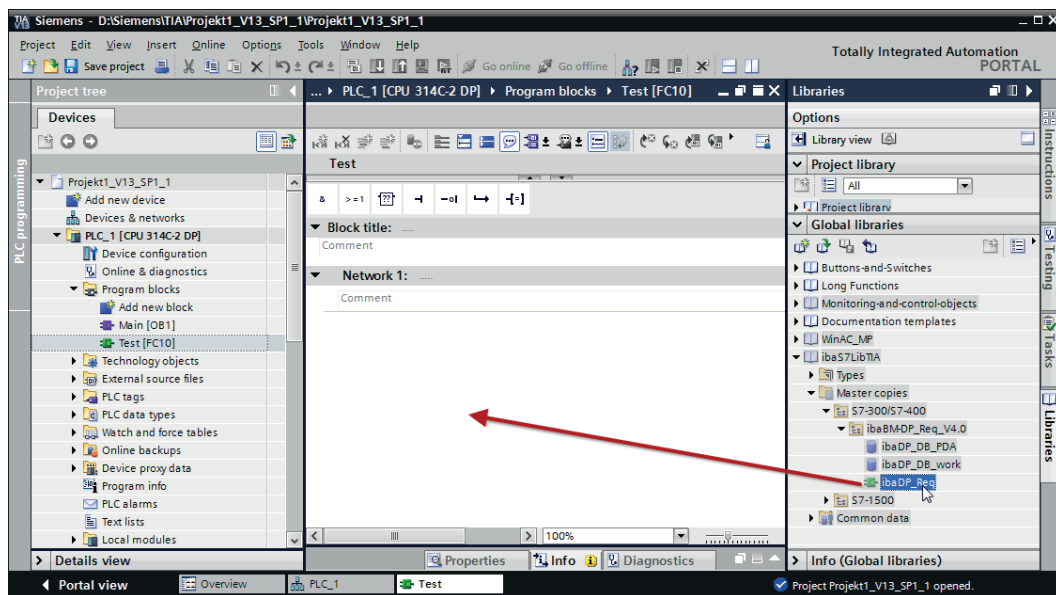
→ Now the library is integrated.



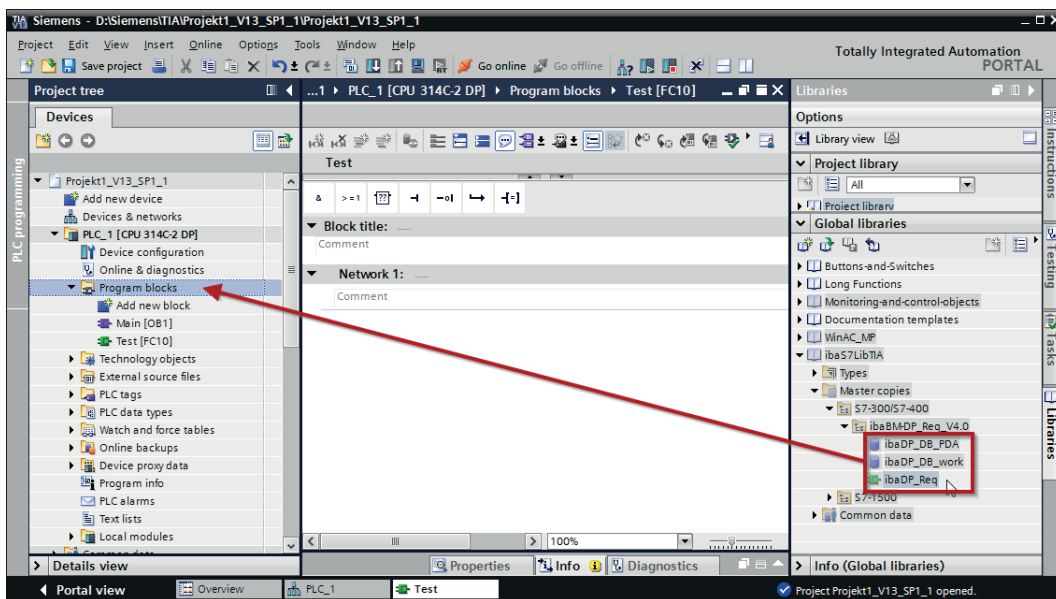
### 7.1.1.2 Using the blocks in TIA Portal

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

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



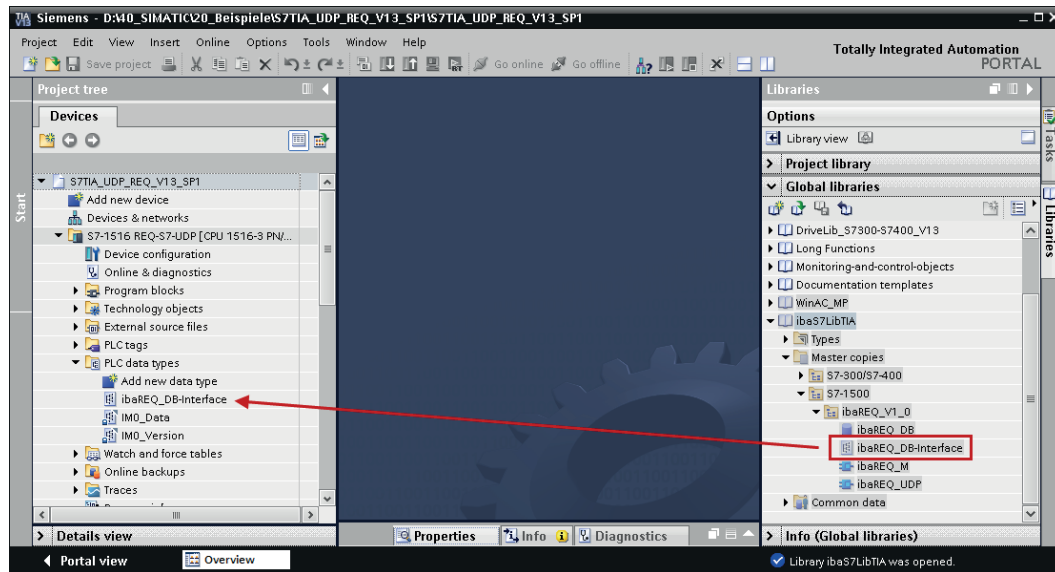
- Drag or copy the blocks to the program blocks directory in the project navigation.



→ You can call the blocks within a project block.



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



## 7.2 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_	ibaPDA_S7CLASSIC_
	S7-400 PN-IF	PN_REQ_Vxx.zip	PN_REQ_Vxx.zip
	S7-1500 PN-IF	S7TIA_	ibaPDA_S7TIA_
		PN_REQ_Vxx.zip	PN_REQ_Vxx.zip
		ibaS7TIA_PN_REQsym_	ibaPDA_S7TIA_PN_REQsym
		Vxx.zap*	
ibaBM-DP	S7-300 DP-IF	S7CLASSIC_	ibaPDA_S7CLASSIC_
		DP_REQ_Vxx.zip	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_	ibaPDA_S7CLASSIC_
		DP_REQ_CFC_Vxx.zip	DP_REQ_CFC_Vxx.zip
	S7-400H	S7CLASSIC_	ibaPDA_S7CLASSIC_
		DP_REQ_H_Vxx.zip	DP_REQ_CPU4xxH_Vxx.zip
	S7-300 DP-IF	S7CLASSIC_	ibaPDA_S7CLASSIC_
		L2B_REQ_Vxx.zip	DP_L2Bcomp_REQ_Vxx.zip
	S7-1500 DP-IF	S7TIA_	ibaPDA_S7TIA_
		DP_REQ_Vxx.zip	DP_REQ_Vxx.zip
ib-aBM-DPM-S	S7-300 DP-IF	S7CLASSIC_	ibaPDA_S7CLASSIC_
		DP_REQ_Vxx.zip	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_	ibaPDA_S7CLASSIC_
		DP_REQ_CFC_Vxx.zip	DPMS_REQ_CFC_Vxx.zip
	S7-400H	S7CLASSIC_	ibaPDA_S7CLASSIC_
		DP_REQ_H_Vxx.zip	DPMS_REQ_CPU4xxH_Vxx.zip
ibaCom-L2B	S7-300 DP-IF	S7CLASSIC_	ibaPDA_S7CLASSIC_
		L2B_REQ_Vxx.zip	L2B_REQ_Vxx.zip
	S7-300 + CP342-5		
	S7-400 with CFC	S7CLASSIC_	ibaPDA_S7CLASSIC_
		L2B_REQ_CFC_Vxx.zip	L2B_REQ_CFC_Vxx.zip
ibaPDA-Interface-ibaNet-E	S7-1500	ibaS7TIA_NetE_REQ_	ibaPDA_S7TIA_NetE_REQ
		Vxx.zap*	
		ibaS7TIA_NetE_	ibaPDA_S7TIA_NetE_REQsym
		REQsym_Vxx.zap*	

Table 3: Configuration examples available on data storage medium

## 7.3 Setting PG/PC interface/defining new access point

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

There are 2 possible remedies:

### Changing the interface with remaining access point name

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

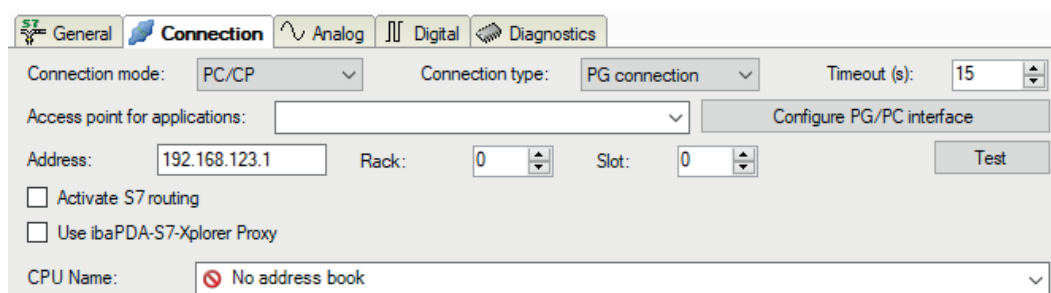
Disadvantage of this method: If the setting of the access point is changed again in the SIMATIC Manager, the measurement does no longer work because *ibaPDA* no longer has any access.

### Adding a special access point for ibaPDA

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

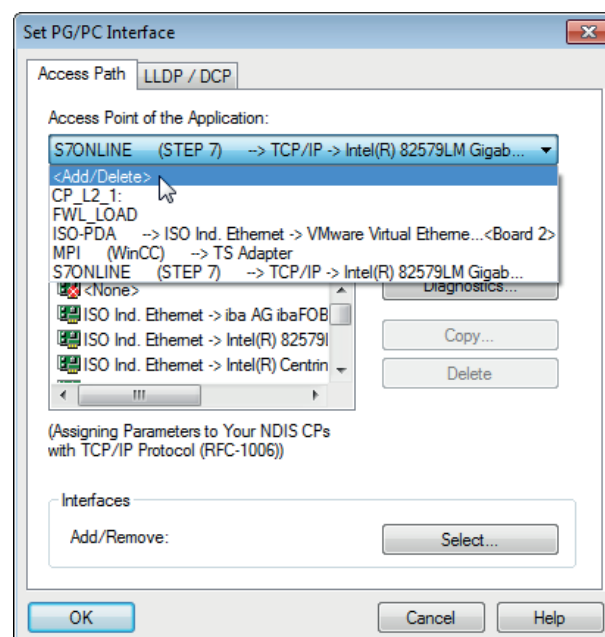
There is the <Configure PG/PC interface> button in the dialog window of the module. With this button, you can open the dialog for configuring the PG/PC interface.

The setting for the SIMATIC Manager is also changed.

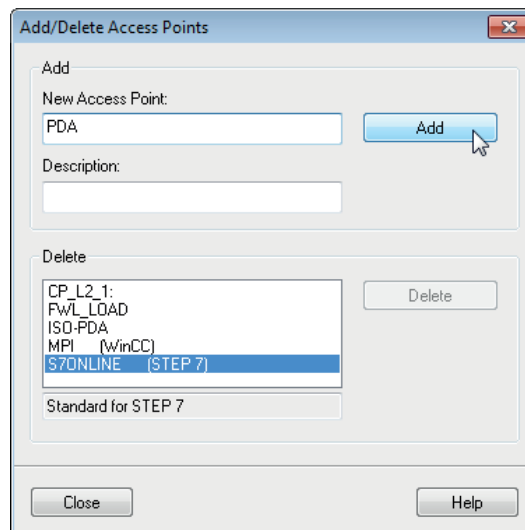


### Procedure

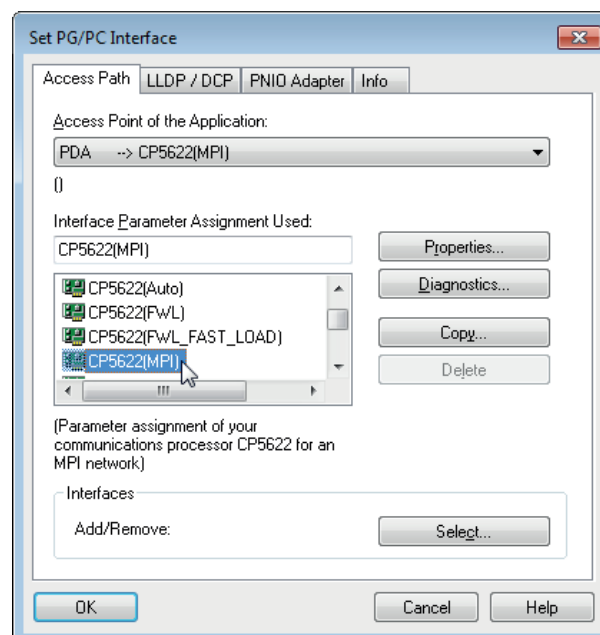
1. Open the dialog for configuring the PG/PC interface with the <Configure PG/PC interface> button.
2. Under *Access Point of the Application* select the row <Add/Delete>.



- Define a new access point: Enter a name, e.g. "PDA", and optionally a description for a better understanding.  
Confirm your entries with <Add> and <Close>.

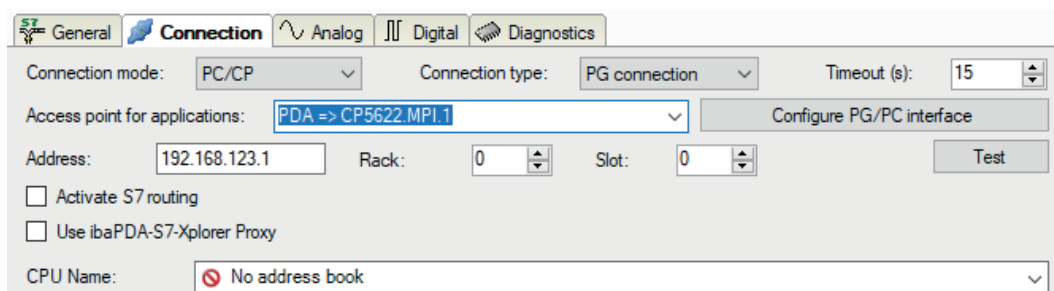


- Add an interface to the access point, e.g. "CP5622 (MPI)".



- Exit the configuration with <OK>.

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



### Notes on the different access points

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

Basically, there are 3 types of access points:

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

#### TCP/IP

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

#### ISO

If you select an access point using an ISO interface, you need to enter the MAC address, rack number and slot number. If you do not know the rack number and/or slot number, enter "0" for slot and click on the <Test> button.

#### Bus system PROFIBUS or MPI

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

## 7.4 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



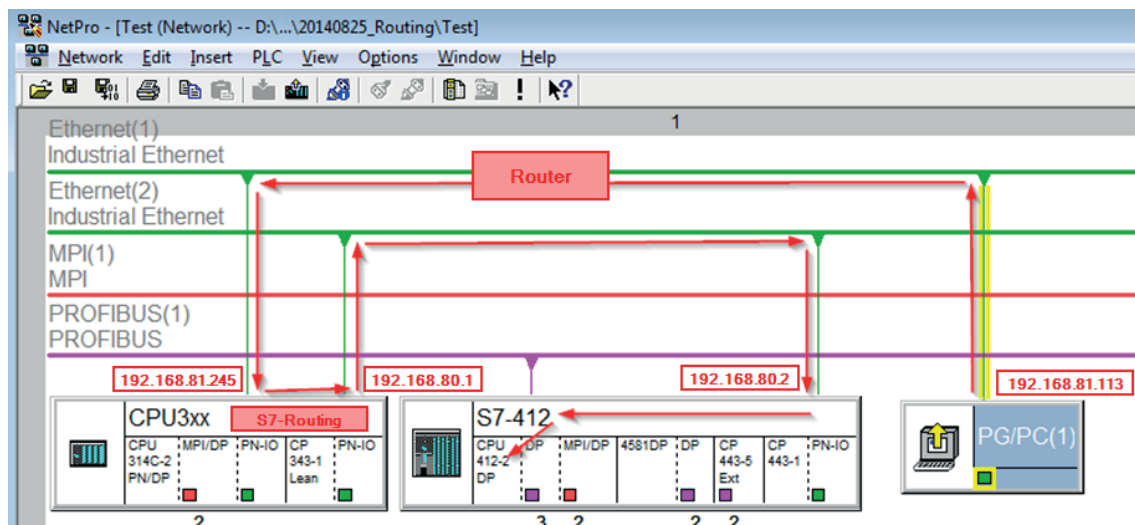
For more information about the S7 Routing, refer to:

- 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>

### 7.4.1 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.

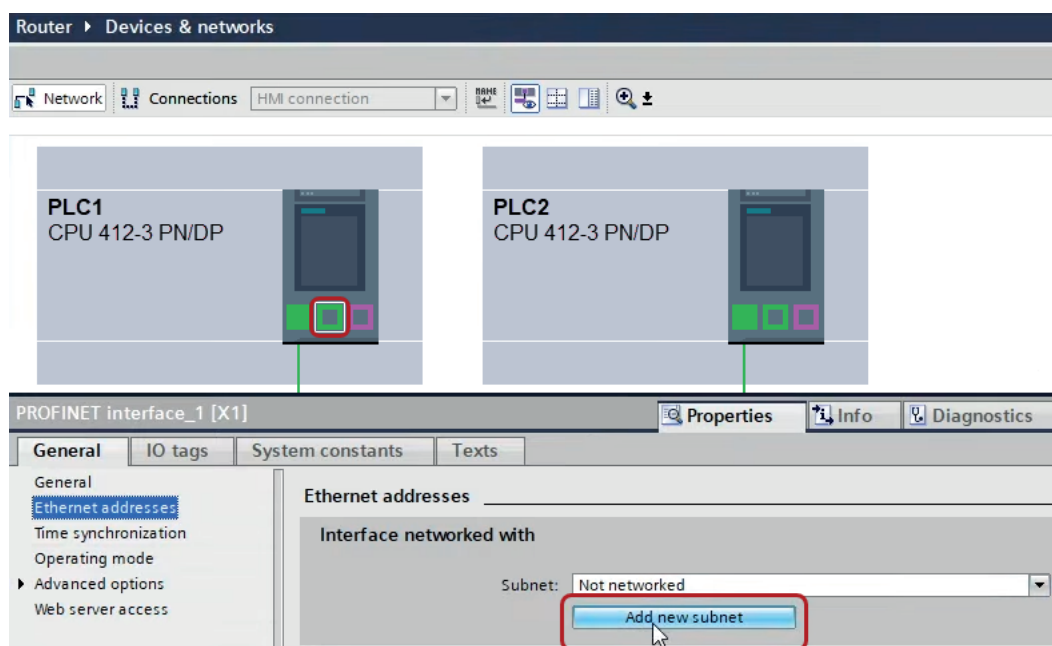
"Passing" the communication through this controller is called *S7 Routing*.

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.

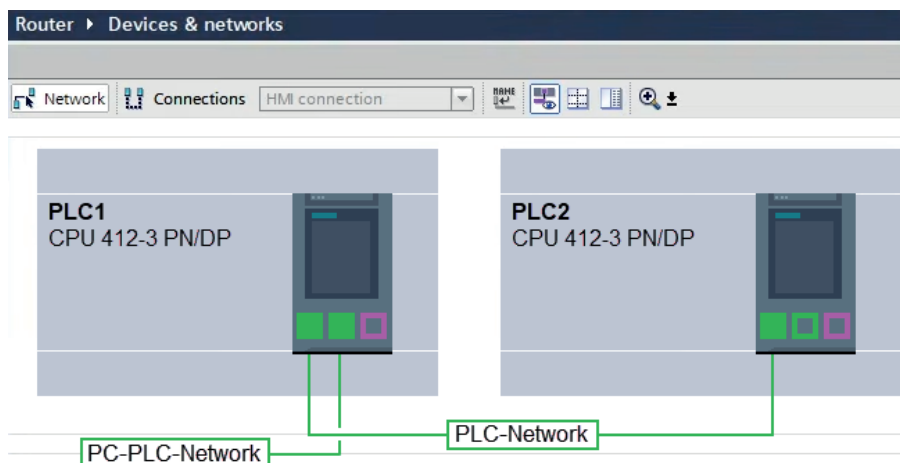
### 7.4.1.1 Configuration of TIA Portal

The following configuration steps are required exclusively for accessing the subordinate control "PLC2" by the programming software TIA 7. For SINUMERIK, SINAMICS, or SIMOTION, you can apply similar steps. For using *ibaPDA*, these steps are not required.

1. Connect both controllers in TIA Portal via the Ethernet ports.
2. Establish a connection with your computer and the first controller "PLC1" by adding a subnet.

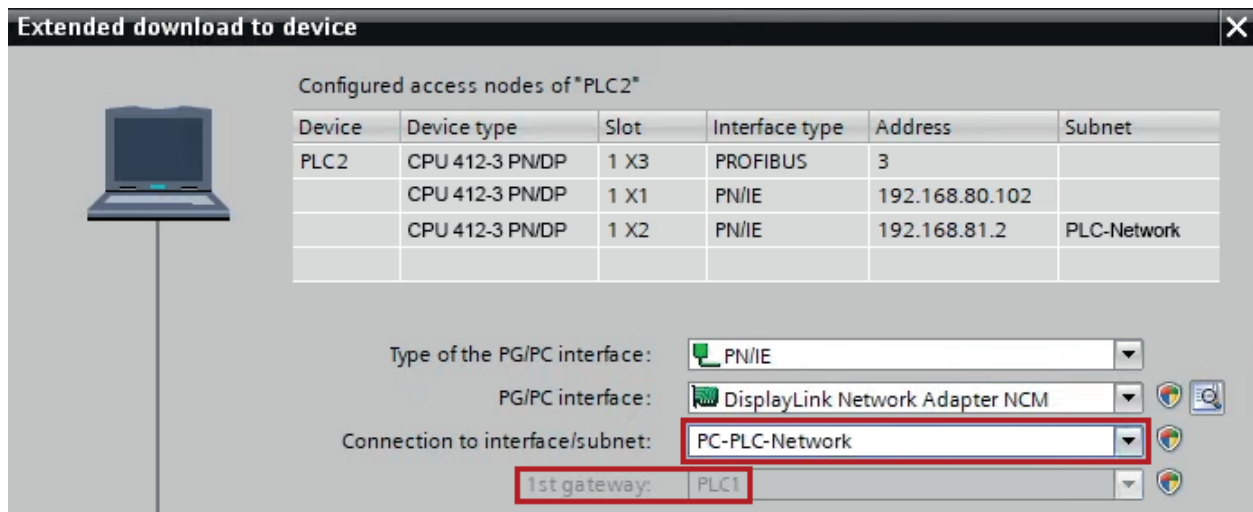


→ The connections between the controllers and to the computer are shown in TIA Portal.



3. Download the programming of the controller "PLC1" and then the programming of the controller "PLC2".

4. In the controller "PLC2", set the connection from "PLC1" to the computer as the connection to the interface/subnet.

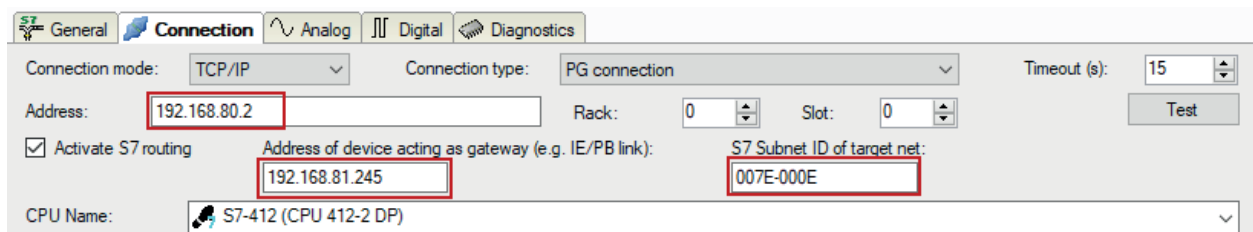


→ "PLC1" appears as the first gateway.

→ "PLC2" is now connected to the computer via S7 routing via "PLC1".

### 7.4.1.2 Configuration of ibaPDA

Configure the following settings.



#### 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

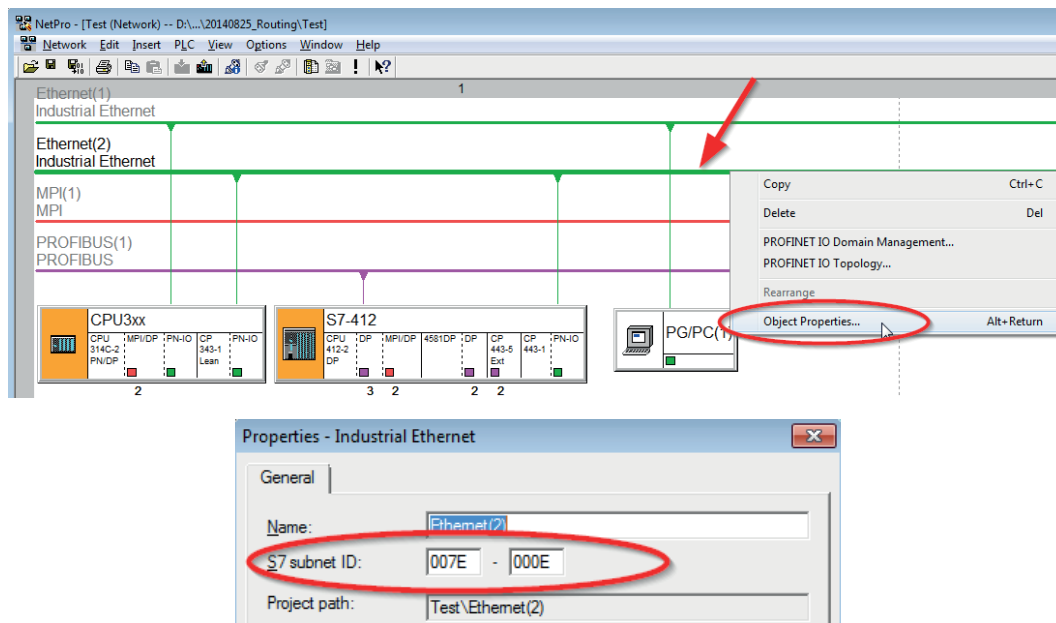
Enter the subnet ID from STEP 7 NetPro or TIA Portal.



## Identifying the S7 subnet ID in NetPro

You can identify the S7 subnet ID in NetPro.

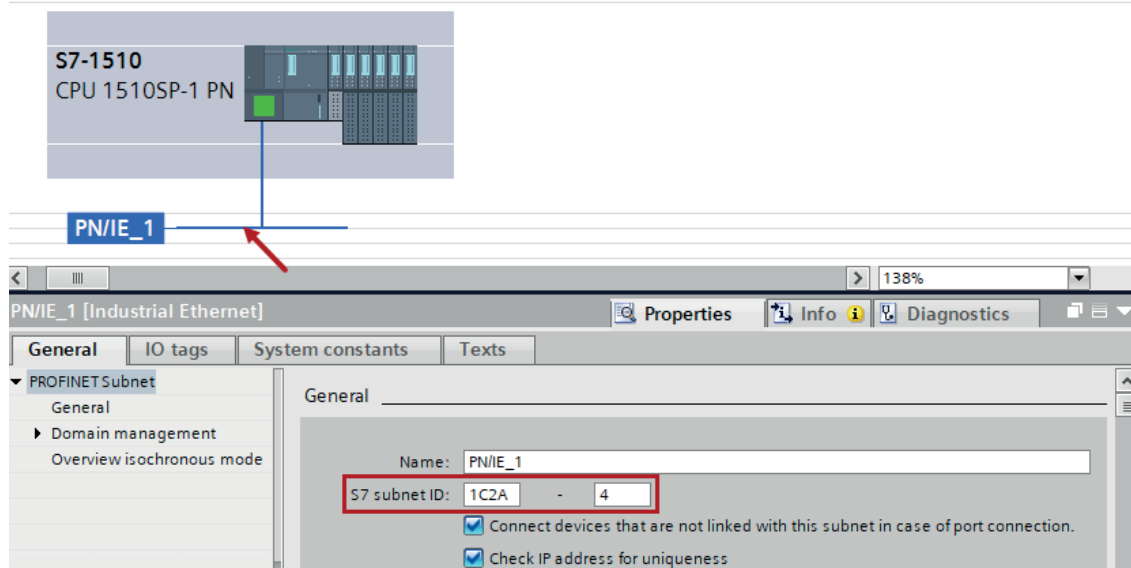
Right-click on the secondary bus system and open the *Object Properties*.



## Identifying the S7 subnet ID in TIA Portal

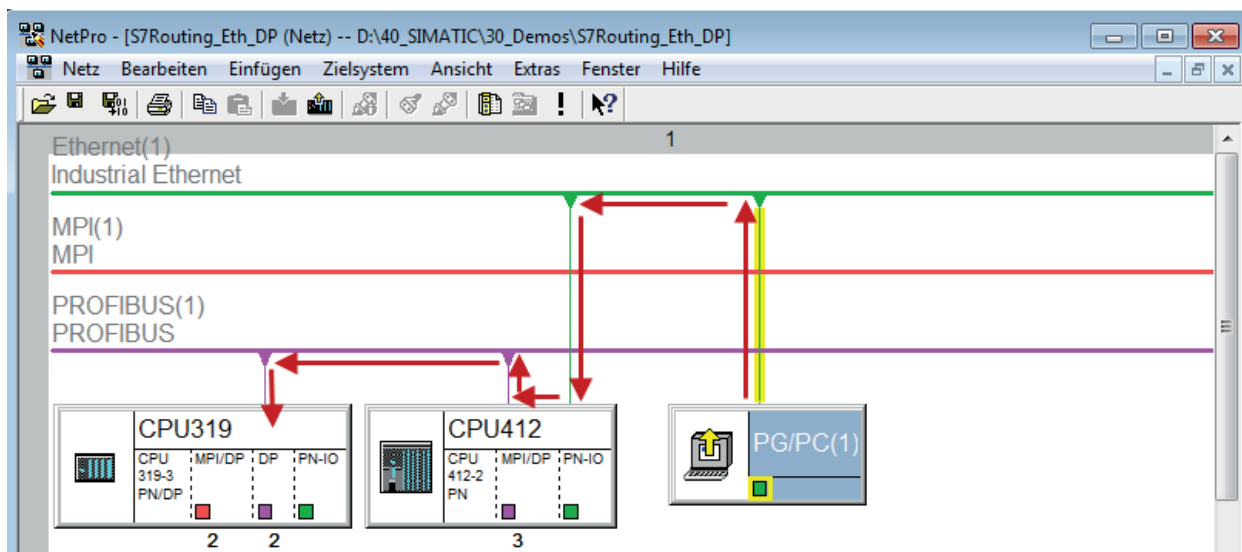
You can identify the S7 subnet ID in TIA Portal.

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



## 7.4.2 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*.

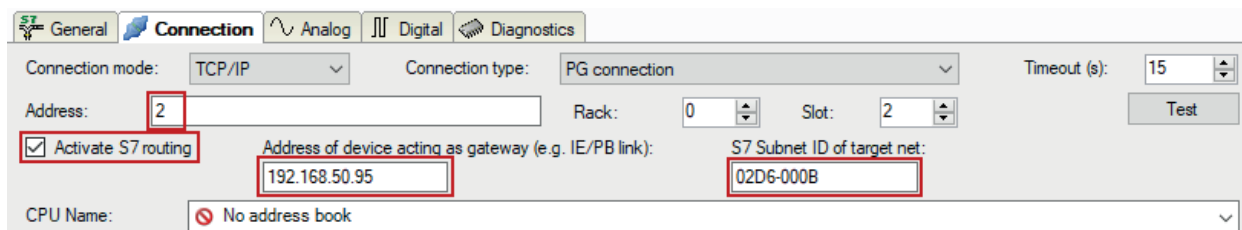
### 7.4.2.1 Configuration of TIA Portal

The configuration steps are required exclusively for accessing the subordinate control "PLC2" by the programming software TIA Portal. For SINUMERIK, SINAMICS, or SIMOTION, you can apply similar steps.

For the configuration of PROFIBUS proceed as described for Ethernet, see [↗ Configuration of TIA Portal](#), page 95.

### 7.4.2.2 Configuration of ibaPDA

Configure the following settings.



#### 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).

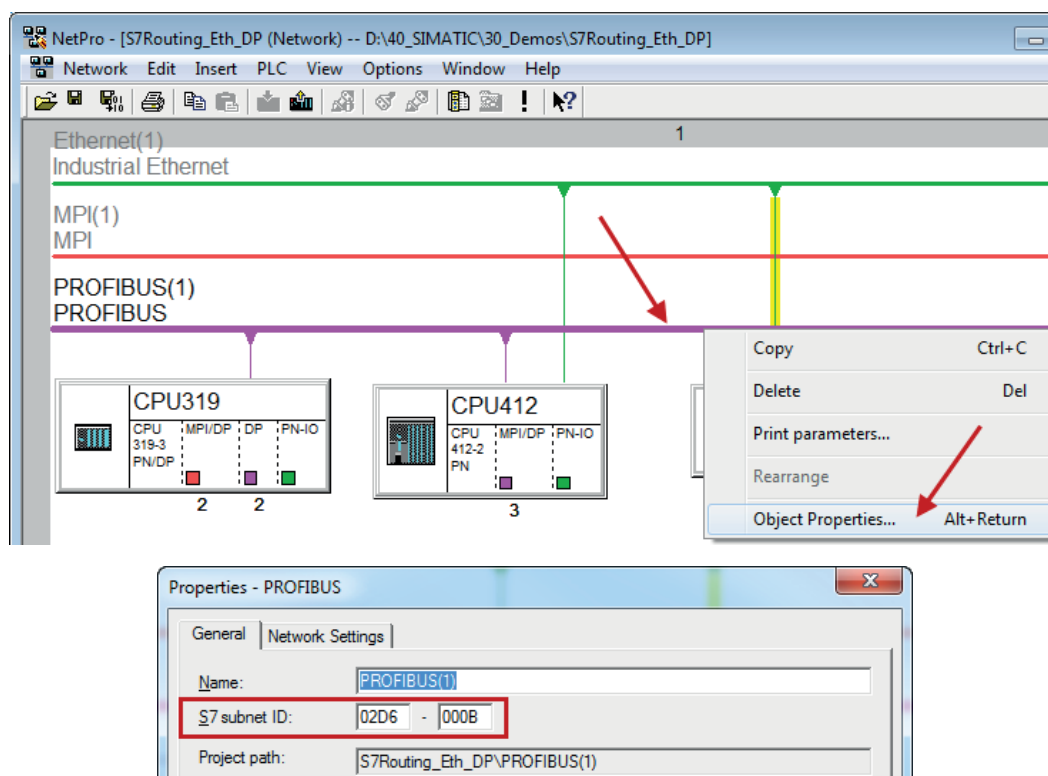
### S7 subnet ID of target net

Enter the subnet ID from STEP 7 NetPro.

### Identifying the S7 subnet ID in NetPro

You can identify the S7 subnet ID in NetPro.

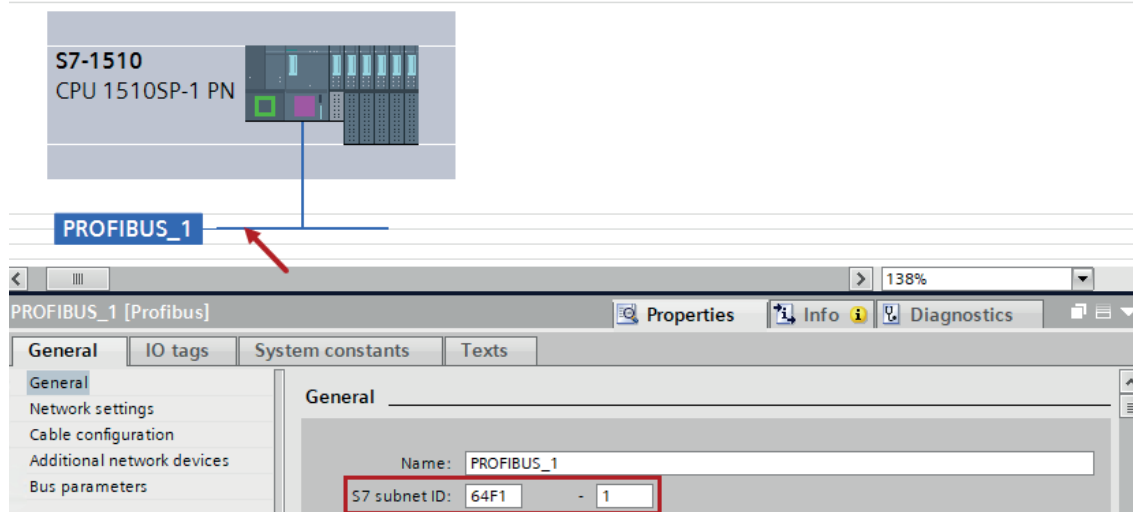
Right-click on the secondary bus system and open the *Object Properties*.



### 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*.



## 7.5 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
32	initialization not completed
41	too many pointers (ibaREQ_DB too small)
42	too many pointers in one command (>128)
44	invalid command id
45	operand invalid (not defined)
46	operand invalid (datatype)
47	operand invalid (memory area)
200	no connection to PN device / DP slave
300	version of ibaREQ_UDPack does not match with ibaREQ_M (ID)
301	version of ibaREQ_UDPack does not match with ibaREQ_M (FB)
302	version of ibaREQ_UDPack does not match with ibaREQ_M (DB)
303	type of transmit agent does not match with configured request type in <i>ibaPDA</i>
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

Value ERROR_STATUS	Description
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 4: 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)

Value ERROR_STATUS	Description
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 5: 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

Value ERROR_STATUS	Description
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
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 6: Error codes of Request block FC123

## 8 Support and contact

### Support

Phone: +49 911 97282-14

Email: [support@iba-ag.com](mailto:support@iba-ag.com)

---

#### Note



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](mailto: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](http://www.iba-ag.com)**