



# ibaPDA-Request-S7-UDP

Request Data Interface to SIMATIC S7 via UDP

Manual  
Issue 1.6

Measurement Systems for Industry and Energy

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

---

## Manufacturer

iba AG  
Koenigswarterstr. 44  
90762 Fuerth  
Germany

## Contacts

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

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

© iba AG 2021, All rights reserved.

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

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

Version	Date	Revision - Chapter / Page	Author	Version SW
1.6	09-2021	Note (5.1.2)	st	7.0.0

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

## Content

<b>1</b>	<b>About this manual.....</b>	<b>5</b>
1.1	Target group and previous knowledge .....	5
1.2	Notations .....	6
1.3	Used symbols.....	7
<b>2</b>	<b>System requirements .....</b>	<b>8</b>
<b>3</b>	<b>ibaPDA-Request-S7-UDP.....</b>	<b>10</b>
3.1	General information .....	10
3.1.1	How does the symbolic Request work? .....	10
3.1.2	How much data can be transferred? .....	11
3.2	Configuration and engineering SIMATIC S7 .....	12
3.2.1	Configuration and engineering SIMATIC S7-300, S7-400 and WinAC .....	12
3.2.1.1	Description of the Request blocks .....	12
3.2.1.2	Configuration in STEP 7.....	22
3.2.2	Configuration and engineering SIMATIC S7-1500 .....	31
3.2.2.1	Description of the Request blocks .....	31
3.2.2.2	Configuration in STEP 7.....	37
3.2.2.3	Device configuration.....	39
3.3	Configuration and engineering ibaPDA.....	41
3.3.1	General interface settings.....	41
3.3.2	Add module .....	42
3.3.3	General module settings.....	43
3.3.4	Connection settings .....	44
3.3.4.1	Connection mode TCP/IP .....	45
3.3.4.2	Connection mode PC/CP .....	47
3.3.4.3	Connection mode TCP/IP S7-1x00 .....	50
3.3.5	Signal configuration .....	53
3.3.5.1	Selection via the absolute address of the operands.....	53
3.3.5.2	Selection via the operand symbols .....	55
3.3.5.3	Measuring the CFC connectors.....	59
3.3.6	Module S7 Request.....	62
3.3.7	Module S7 Request Decoder .....	63
3.3.8	Module diagnostics.....	64

3.3.9	Address books .....	66
<b>4</b>	<b>Diagnostics.....</b>	<b>71</b>
4.1	Checking the license .....	71
4.2	Log files.....	72
4.3	Connection diagnostics with PING.....	73
4.4	Connection table .....	74
4.5	Module diagnostics.....	74
<b>5</b>	<b>Appendix .....</b>	<b>75</b>
5.1	Iba S7 library.....	75
5.1.1	iba S7 library for SIMATIC Manager .....	75
5.1.1.1	Integrating the library.....	76
5.1.1.2	Using the blocks.....	78
5.1.2	iba S7 library for SIMATIC TIA portal.....	79
5.1.2.1	Integrating the library.....	81
5.1.2.2	Using the blocks.....	83
5.2	Application examples.....	84
5.3	S7 cycle time measurements .....	84
5.4	Adaptation to the renumbered system functions.....	86
5.5	Setting PG/PC interface / defining new access point.....	88
5.6	S7 routing .....	92
5.6.1	Routing from Ethernet to Ethernet .....	92
5.6.1.1	Configuration of STEP 7/ NetPro .....	93
5.6.1.2	ibaPDA configuration.....	95
5.6.2	Routing from Ethernet to PROFIBUS .....	96
5.6.2.1	Configuration STEP 7/ NetPro.....	97
5.6.2.2	ibaPDA configuration.....	99
5.7	Error codes of Request blocks .....	100
5.7.1	S7-1500.....	104
5.7.2	S7-300/400 .....	105
5.7.3	More error messages.....	108
<b>6</b>	<b>Support and contact.....</b>	<b>112</b>

# 1 About this manual

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

The product *ibaPDA-Request-S7-UDP* is an extension of *ibaPDA* for free selection of S7 symbols and S7 operands when recording data from SIMATIC S7 CPUs. For transferring data, the UDP<sup>1)</sup> network protocol is used. In this manual, only the extensions and differences are shown. For all other functions and operating options, please see the *ibaPDA* and *ibaPDA-Interface-S7-TCP/UDP* manuals.

---

## Other documentation



This documentation provides supplementary information to the *ibaPDA* and *ibaPDA-Interface-S7-TCP/UDP* manuals.

---

## 1.1 Target group and previous knowledge

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

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

- Windows operating system
- Basic knowledge of *ibaPDA*
- Knowledge of configuration and operation of the relevant control system

<sup>1)</sup> User Datagram Protocol

## 1.2 Notations

In this manual, the following notations are used:

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

## 1.3 Used symbols

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

---

### Danger!



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

- Observe the specified measures.
- 

### Warning!



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

- Observe the specified measures.
- 

### Caution!



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

- Observe the specified measures
- 

### Note



A note specifies special requirements or actions to be observed.

---

### Tip



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

---

### Other documentation



Reference to additional documentation or further reading.

---

## 2 System requirements

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

- *ibaPDA* v7.0.0 or higher
- Basic license for *ibaPDA*
- Additional license for *ibaPDA-Interface-S7-TCP/UDP*
- Additional license for *ibaPDA-Request-S7-UDP*
- SIMATIC S7 controller S7-300, S7-400, S7-400H, S7-1500, WinAC
- In case PC/CP connections are used:
  - SIMATIC STEP 7 or SIMATIC NET, or
  - SIMATIC TIA Portal
- SIMATIC CFC (beginning with V6.0), if signals are to be chosen by drag & drop<sup>2)</sup>

For integrating the Request blocks in the S7 program:

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

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

### System restrictions

- Access to optimized data blocks of S7-1500 controllers is not supported.
- Access to S7-1200 controllers is not supported.
- Connectors of CFC blocks, which have constant values assigned, have no operand address. They are marked as constant in the address book and cannot be selected as signal.
- If FB function blocks are used in CFC, the internal static variables of the FB are also displayed in the address book, because they are treated in exactly the same way as connectors by the compiler. These are to be ignored.
- For data acquisition, *ibaPDA* supports only the following data types:
  - BOOL, BYTE, WORD, DWORD, INT, DINT, REAL, TIME, CHARAll other data types exist in the address book but cannot be entered in the signal list.
- For functions FC with connections of the data type STRING, POINTER, STRUCT or ANY under CFC, the interpretation of the SCL code does not work, as there are no references to the data types available in the source.

<sup>2)</sup> can only be used with SIMATIC STEP 7 v5.x



**Licenses**

Order no.	Name	Description
31.001040	ibaPDA-Interface-S7-TCP/UDP	Extension license for an ibaPDA system by a TCP/IP and UDP/IP interface Number of connections: 64
31.101040	one-step-up-Interface-S7-TCP/UDP	Extension license for the extension of an existing ibaPDA-Interface-S7-TCP/UDP interface by another 64 S7-TCP/UDP connections, a max. of 3 permitted
31.001311	ibaPDA-Request-S7-UDP	Extension license for an ibaPDA system for using the Request functionality with the ibaPDA-Interface-S7-TCP/UDP interface Number of connections: 2
31.101311	one-step-up-Request-S7-UDP	Extension license for extending an existing ibaPDA-Request-S7-UDP by another 2 Request-S7-UDP connections, a max. of 127 permitted.

Table 1: License overview, version ibaPDA v6.35

The use of *ibaPDA-Request-S7-UDP*, requires the existence of an *ibaPDA-Interface-S7-TCP/UDP*-license. For each active Request module, one S7-TCP/UDP connection and one Request-S7-UDP connection are occupied.

## 3 ibaPDA-Request-S7-UDP

### 3.1 General information

The *ibaPDA-Request-S7-UDP* interface is suitable for the measurement data acquisition with free variable selection over standard network cards via UDP. The measurement data is sent actively from the controller to *ibaPDA*. For this purpose, several Request blocks have to be integrated in the S7 program for each connection. These Request blocks are used to cyclically send the current values of the S7 operands selected by the user within *ibaPDA* to be recorded in *ibaPDA*. When modifying the signal selection, no modification in the S7 program is required.

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

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

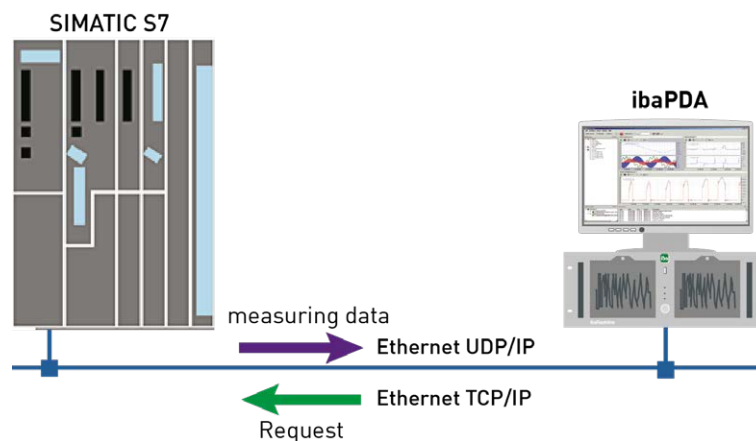


Fig. 1: System topology

On side of the SIMATIC S7, a PROFINET interface integrated on the CPU can be used as well as an additional Ethernet capable communication processor (e.g. CP343-1, CPU343-1 LEAN, CP443-1, ...). The data volume that can be reached depends - besides of many other factors - significantly on the selected interface. Generally, interfaces integrated on the CPU are more powerful than communication processors, since the connection via the backplane bus represents a performance shortage. Especially for the modules of the S7-300 family, there are considerable shortages. For detailed information, please see the Siemens device and system manuals.

#### 3.1.1 How does the symbolic Request work?

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

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

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

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

### 3.1.2 How much data can be transferred?

The amount of transferable data per module is limited by various parameters:

#### 1. Maximum telegram size

A maximum of 1466 bytes of user data can be transferred per telegram and thus per connection.

#### 2. Maximum number of pointers

The data to be transferred is defined using ANY pointers, which are transferred from *ibaPDA* to the controller. Here, connected operands (i.e. successive addresses) are represented by a common pointer.

- Depending on the size of the REQ\_DB, a different number of pointers can be managed. With S7-300, S7-400, WinAC controls the length can be freely selected.
  - 5280 bytes: up to 128 pointers
  - 9120 bytes: up to 512 pointers
  - 14240 bytes: up to 1024 pointers (maximum)
- Only a fixed length is provided for S7-1500 controllers:
  - 9120 bytes: up to 512 pointers

#### Example:

If a REQ\_DB with a length of 9120 bytes is used, up to 512 pointers can be used, which may write a total address space of 1466 bytes. If 512 distributed individual bytes of operands are to be acquired, all 512 pointers are required: Only 512 bytes can be acquired, although the maximum telegram size of 1466 bytes has not yet been reached. If the 512 byte operands are located on successive addresses, only 1 pointer is required. There are 511 pointers available to address the remaining 1466 bytes - 512 bytes = 954 bytes.

The maximum telegram size and the current maximum number of pointers can be viewed in the *S7 Request info* tab under *Diagnostics*.

See also ➔ *Module diagnostics*, page 64.

## 3.2 Configuration and engineering SIMATIC S7

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

You should distinguish whether this configuration is done with the SIMATIC Manager (STEP 7 Version  $\leq$  V5) or with the SIMATIC TIA Portal.

### 3.2.1 Configuration and engineering SIMATIC S7-300, S7-400 and WinAC

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

- Configuration Software (STEP 7 V5):  
Integration of the Request blocks in the S7 program
- Connection configuration:  
If a communication processor CP x43-1 is being used, a programmed connection has to be configured in NetPro. This is not necessary when using a PN interface integrated on the CPU.

#### 3.2.1.1 Description of the Request blocks

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

One set of Request blocks has to be called for each Request module (connection) in *ibaPDA*. The used blocks are part of the iba S7 library (see chapter [Iba S7 library](#), page 75).

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

Request block	CPU with integrated PN interface or WinAC RTX	S7-300 CPU + CP343-1	S7-400 CPU + CP443-1	recommended call-up level
ibaREQ_M (FB140)	X	X	X	OB1
ibaREQ_UDPact (FB145)	X	X	X	OB3x
ibaREQ_UDPint (FB146)	X			OB3x
ibaREQ_UDPext3 (FB147)		X		OB3x
ibaREQ_UDPext4 (FB148)			X	OB3x
ibaREQ_DB (DB15)	X	X	X	-
ibaUDT_UDPact (UDT145)	X	X	X	-

The following blocks always have to be used:

- **ibaREQ\_M (Management)**  
The block realizes the communication with *ibaPDA*. Ideally, the block is called in the OB1. This block has to be called separately in every system configuration for each module in *ibaPDA*.
- **ibaREQ\_UDPact (provides the current signal values)**  
This block provides the current signal values in the call up cycle. The block is called internally in the blocks *ibaREQ\_UDPint*, *ibaREQ\_UDPext3*, or *ibaREQ\_UDPext4*. This is why the block always has to be present in the project, but does not always has to be called separately.
- **ibaREQ\_DB (interface DB)**  
This DB serves as interface to *ibaPDA* and between the different Request blocks.

The following blocks are to be used depending on the S7 system configuration:

- **ibaREQ\_UDPint**  
This block sends the provided current signal values via an integrated PN interface.
- **ibaREQ\_UDPext3**  
This block sends the provided current signal values via an external communication processor CP343-1.
- **ibaREQ\_UDPext4**  
This block sends the provided current signal values via an external communication processor CP443-1.

The blocks *ibaREQ\_UDPint*, *ibaREQ\_UDPext3* and *ibaREQ\_UDPext4* always have to be used alternatively.

### 3.2.1.1.1 ibaREQ\_M (FB140)

#### Description of the formal parameters:

Name	Type	Data type	Description
REQ_DB	IN	BLOCK_DB	DB of the ibaPDA communication interface ibaREQ_DB
RESET	IN	BOOL	FALSE -> no reset (Standard) TRUE -> reset of the communication connection
INP_RANGE	IN	INT	Number of input bytes (evaluation only when initializing), 0: automatic detection (recommended)
OUT_RANGE	IN	INT	Number of output bytes (evaluation only when initializing), 0: automatic detection (recommended)
MARKER_RANGE	IN	INT	Number of marker bytes (evaluation only when initializing), 0: automatic detection (recommended)
ERROR_STATUS	OUT	DWORD	Internal error code

The following SFCs are used internally:

- SFC 20 (BLKMOV)
- SFC 21 (FILL)
- SFC 24 (TEST\_DB)
- SFC 51 (RDSYSST)

#### Detailed description:

##### REQ\_DB

The data is exchanged with *ibaPDA* via this DB. The identical DB must be configured for all related Request blocks.

Length:

5280 Bytes : up to 128 pointers (minimum)

9120 Bytes : up to 512 pointers

14240 Bytes : up to 1024 pointers (maximum)

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

##### RESET

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

**INP\_RANGE**

Limits the number of input bytes to be measured.

For INP\_RANGE = 0 the size of the available processor image of the inputs is determined by the Request FB itself (recommended). The evaluation is performed only during the initialization phase of the function block.

**OUT\_RANGE**

Limits the number of output bytes to be measured.

For OUT\_RANGE = 0 the size of the available processor image of the outputs is determined by the Request FB itself (recommended). The evaluation is performed only during the initialization phase of the function block.

**MARKER\_RANGE**

Limits the number of marker bytes to be measured.

For MARKER\_RANGE = 0 the number of available markers is determined by the Request FB itself (recommended). The evaluation is performed only during the initialization phase of the function block.

**ERROR\_STATUS**

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

A list of all possible error codes can be found in chapter [➤ Error codes of Request blocks](#), page 100.

**3.2.1.1.2 ibaREQ\_UDPact (FB145)****Description of the formal parameters:**

Name	Type	Data type	Description
REQ_DB	IN	BLOCK_DB	DB of the ibaPDA communication interface ibaREQ_DB
Xchange	INOUT	UDT145	Interface for the calling block
ERROR_STATUS	OUT	WORD	Internal error code

The following SFCs are used internally

- SFC 20 (BLKMOV)
- SFC 21 (FILL)
- SFC 36 (MSK\_FLT)
- SFC 37 (DMSK\_FLT)

**Detailed description:****REQ\_DB**

The data is exchanged with *ibaPDA* via this DB. The identical DB must be configured for all related Request blocks.

**Xchange**

Via the parameterized data range, the data are exchanged with the calling block.

**ERROR\_STATUS**

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

A list of all possible error codes can be found in chapter [↗ Error codes of Request blocks](#), page 100.

**3.2.1.1.3 ibaREQ\_UDPint (FB146)****Description of the formal parameters:**

Name	Type	Data type	Description
REQ_DB	IN	BLOCK_DB	DB of the ibaPDA communication interface ibaREQ_DB
CON_ID	IN	INT	Unique connection ID of the sending block (TUSEND)
LOCAL_DEVICE_ID	IN	BYTE	Device ID of the sending block (TUSEND)
LOCAL_PORT	IN	DINT	Used local port number
RESET_CON	IN	BOOL	FALSE -> no reset (standard) TRUE -> reset of the communication connection
ERROR_STATUS	OUT	WORD	Internal error code
ERROR_TCON	OUT	WORD	Error code connection setup of the TCON function block
COUNT_TCON	OUT	WORD	Counter attempts for connecting
ERROR_TSEND	OUT	WORD	Error code of the sending block TUSEND
COUNT_TSEND	OUT	WORD	Counter sent telegrams
LOST_SAMPLES	OUT	WORD	Counter for lost measurement values

The following SFCs are used internally:

- FB145 (ibaREQ\_UDPact)
- FB 65 (TCON)
- FB 66 (TDISCON)
- FB 67 (TUSEND)
- SFB 4 (TON)
- UDT 65 (TCON\_PAR)
- UDT 66 (TADDR\_PAR)
- UDT 145 (ibaUDT\_UDPact)



**Detailed description:****REQ\_DB**

The data is exchanged with *ibaPDA* via this DB. The identical DB must be configured for all related Request blocks.

**CON\_ID**

Unique reference to the connection to be established. Corresponds to the parameter ID of the Siemens standard block TCON.

**LOCAL\_DEVICE\_ID**

ID of the used interface. Corresponds to the parameter `local_device_id` in the CONNECT structure of the Siemens standard block TCON.

Value	Meaning
B#16#01	Communication via IE interface in interface slot 1 (IF1) with WinAC RTX or an IM 151-8 PN/DP CPU.
B#16#02	Communication through the integrated PROFINET interface of the CPU31x-2 PN/DP, CPU314C-2 PN/DP and IM154-8 CPU.
B#16#03	Communication through the integrated PROFINET interface of the CPU319-3 PN/DP, CPU315T-3 PN/DP, CPU317T-3 PN/DP, CPU317TF-3PN/DP.
B#16#04	Communication through SINUMERIK NCU7x0.2 PN with CPU319-3 PN/DP and SINUMERIK NCU7x0.3PN with CPU317-2 PN/DP.
B#16#05	Communication through the integrated PROFINET interface of the CPU412-2 PN, CPU414-3 PN/DP, CPU416-3 PN/DP, CPU412-5H PN/DP (Rack 0), CPU414-5H PN/DP (Rack 0), CPU416-5H PN/DP (Rack 0) and CPU417-5H PN/DP (Rack 0).
B#16#06	Communication via the IE interface in interface slot 2 (IF2) with WinAC RTX
B#16#08	Communication via the integrated PROFINET interface of the CPU410-5H (Rack 0)
B#16#0B	Communication via the IE interface in interface slot 3 (IF3) with WinAC RTX
B#16#0F	Communication via the IE interface in interface slot 4 (IF4) with WinAC RTX
B#16#15	Communication through the integrated PROFINET interface of the CPU412-5H PN/DP (Rack 1), CPU414-5H PN/DP (Rack 1), CPU416-5H PN/DP (Rack 1) and CPU417-5H PN/DP (Rack 1).
B#16#18	Communication via the integrated PROFINET interface of the CPU 410-5H (Rack 1)

Table 2: Valid values of the parameter LOCAL\_DEVICE\_ID

**Other documentation**

You find more information under the following link:

<https://support.industry.siemens.com/cs/document/51339682>

**LOCAL\_PORT**

Number of the locally used port.

**RESET\_CON**

Serves for manually resetting the communication connection.

**ERROR\_STATUS**

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

A list of all possible error codes can be found in chapter ↗ *Error codes of Request blocks*, page 100.

**ERROR\_TCON**

Error code of connection setup. The standard error codes for the TCON function block are output.

0X8yyy error code of TCON block will be passed

**COUNT\_TCON**

Counter for the attempts of connection setup. An increasing value indicates problems when establishing the connection to the *ibaPDA* computer.

**ERROR\_TSEND**

Error code when sending. The standard error codes of the TSEND block are output.

0X8yyy error code of TSEND block will be passed

**COUNT\_TSEND**

Counter for the sent telegrams. Usually, the counter is incremented continuously.

**LOST\_SAMPLES**

The counter is incremented with every call of the block if no new UDP telegram can be sent to *ibaPDA*, as the previous send order has not been finished, yet. A continuously rising value indicates a shortage in the communication performance.

### 3.2.1.1.4 ibaREQ\_UDPext3 (FB147)

#### Description of the formal parameters:

Name	Type	Data type	Description
REQ_DB	IN	Block_DB	DB of the ibaPDA communication interface ibaREQ_DB
ID	IN	Int	Connection ID of the connection configured in NetPro
HW_LADDR	IN	Word	Module starting address of the CP
ERROR_STATUS	OUT	Word	Internal error code
ERROR_SEND	OUT	Word	Error code of the send block AG_SEND
COUNT_SEND	OUT	Word	Telegram counter of the send block
LOST_SAMPLES	OUT	WORD	Counter for lost measurement values

The following SFCs are used internally:

- FB145 (ibaREQ\_UDPact)
- FC 5 (AG\_SEND)
- UDT 145 (ibaUDT\_UDPact)

#### Detailed description:

##### REQ\_DB

The data is exchanged with *ibaPDA* via this DB. The identical DB must be configured for all related Request blocks.

##### ID

Unique reference to the connection to be established. Has to match the ID used in NetPro.

##### HW\_LADDR

Module starting address of the used CP. Has to match the LADDR used in NetPro.

##### ERROR\_STATUS

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

A list of all possible error codes can be found in chapter [↗ Error codes of Request blocks](#), page 100

##### ERROR\_SEND

Error code when sending The standard error codes of the AG\_SEND block are issued.

0X8yyy error code of AG\_SEND block will be passed

##### COUNT\_SEND

Counter for the sent telegrams. Usually, the counter is incremented continuously.

**LOST\_SAMPLES**

The counter is incremented with every call of the block if no new UDP telegram can be sent to *ibaPDA*, as the previous send order has not been finished, yet. A continuously rising value indicates a shortage in the communication performance.

**3.2.1.1.5 ibaREQ\_UDPext4 (FB148)****Description of the formal parameters:**

Name	Type	Data type	Description
REQ_DB	IN	Block_DB	DB of the ibaPDA communication interface ibaREQ_DB
ID	IN	INT	Connection ID of the connection configured in NetPro
HW_LADDR	IN	WORD	Module starting address of the CP
ERROR_STATUS	OUT	WORD	Internal error code
ERROR_SEND	OUT	WORD	Error code of the send block AG_LSEND
COUNT_SEND	OUT	WORD	Telegram counter of the send block
LOST_SAMPLES	OUT	WORD	Counter for lost measurement values

The following SFCs are used internally:

- FB145 (ibaREQ\_UDPact)
- FC 50 (AG\_LSEND)
- UDT 145 (ibaUDT\_UDPact)

**Detailed description:****REQ\_DB**

The data is exchanged with *ibaPDA* via this DB. The identical DB must be configured for all related Request blocks.

**ID**

Unique reference to the connection to be established. Has to match the ID used in NetPro.

**HW\_LADDR**

Module starting address of the used CP. Has to match the LADDR used in NetPro.

**ERROR\_STATUS**

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

A list of all possible error codes can be found in chapter [➤ Error codes of Request blocks](#), page 100

**ERROR\_SEND**

Error code when sending. The standard error codes of the AG\_SEND block are issued.

The following error codes can be displayed:

0X8yyy error code of AG\_LSEND block will be passed

**COUNT\_SEND**

Counter for the sent telegrams. Usually, the counter is incremented continuously.

**LOST\_SAMPLES**

The counter is incremented with every call of the block if no new UDP telegram can be sent to *ibaPDA*, as the previous send order has not been finished, yet. A continuously rising value indicates a shortage in the communication performance.

### 3.2.1.2 Configuration in STEP 7

In the following, we describe the configuration of the Request blocks in STEP 7 V5.

---

#### Note



The request blocks do not support multi-instance calls.

---

Copy the required blocks from the iba S7 library (see chapter [↗ Iba S7 library](#), page 75) to the blocks folder of your STEP 7 project.

---

#### Note



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

The blocks ibaREQ\_UDPint, ibaREQ\_UDPext3 and ibaREQ\_UDPext4 have to be adapted when

- for the block ibaREQ\_UDPact another function block number than FB145 is used
- for the ibaUDT\_UDPact data type another number than UDT145 is used
- for the Siemens function blocks of the standard library or the SIMATIC NET CP library other numbers than the standard numbers are used. The relevant Siemens function blocks are

- when using ibaREQ\_UDPint (FB146):  
TCON (FB65), TDISCON (FB66), TUSEND (FB67), TCON\_PAR (UDT65), TADDR\_PAR (UDT66)
- when using ibaREQ\_UDPext3 (FB147):  
AG\_SEND (FC5)
- when using ibaREQ\_UDPext4 (FB148):  
AG\_LSEND (FC50)

For further information on customizing, see chapter [↗ Adaptation to the renumbered system functions](#), page 86.

---

#### 3.2.1.2.1 CPU S7-300 / S7-400 / WinAC with integrated PN interface

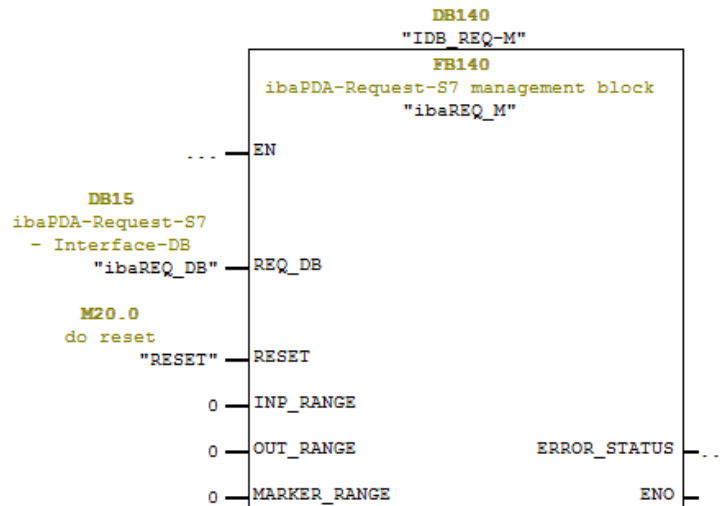
The following blocks are required:

- ibaREQ\_M (FB140)
- ibaREQ\_UDPact (FB145)

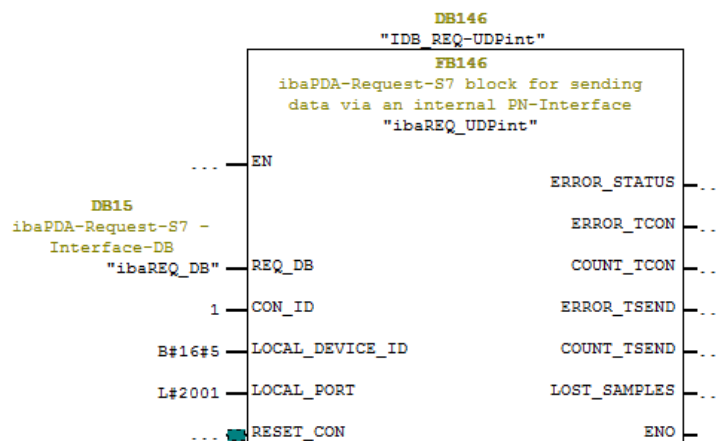
- ibaREQ\_UDPint (FB146)
- ibaREQ\_DB (DB15)
- ibaUDT\_UDPact (UDT145)

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

1. Call the ibaREQ\_M (FB140), preferably within the OB1.



2. Call the ibaREQ\_UDPint (FB146), preferably within a cyclic interrupt OB (OB3x)



#### For each additional Request module:

- In the blocks folder, a data block ibaREQ\_DB (DB15) has to be available for each Request module. Copy the data block and assign a new unique DB number.
- Within the OB1, the ibaREQ\_M (FB140) has to be called once more for each Request module with a new DB number (input REQ\_DB).
- Within a cyclic interrupt OB (OB3x), the ibaREQ\_UDPint (FB146) has to be called for each Request module once more with the new DB number (input REQ\_DB).
- Please consider that all instance data blocks have to be unique and that the values for the CON\_ID and LOCAL\_PORT parameters are assigned uniquely.

**Finish:**

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

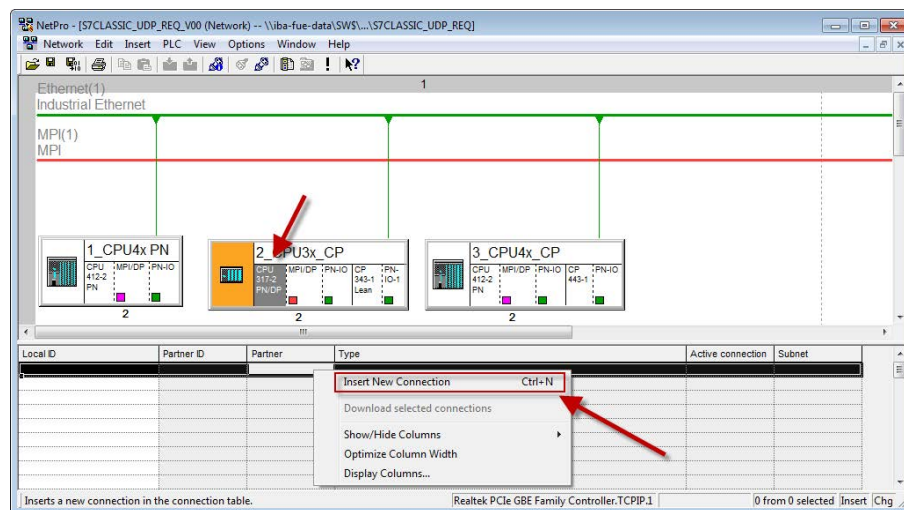
**3.2.1.2.2 CPU S7-300 with CP343-1**

The following function blocks are required:

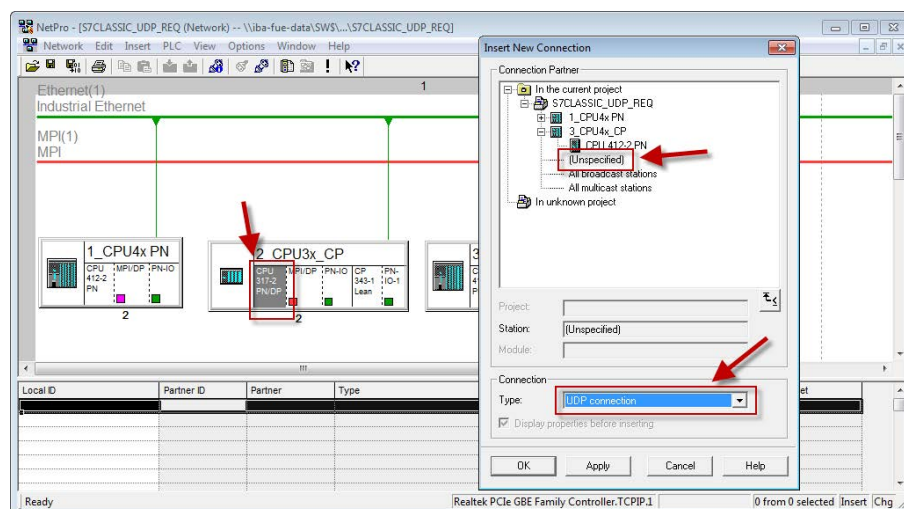
- ibaREQ\_M
- ibaREQ\_UDPackt
- ibaREQ\_UDPext3
- ibaREQ\_DB
- ibaUDT\_UDPackt (UDT145)

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

1. Configure a new connection in NetPro.

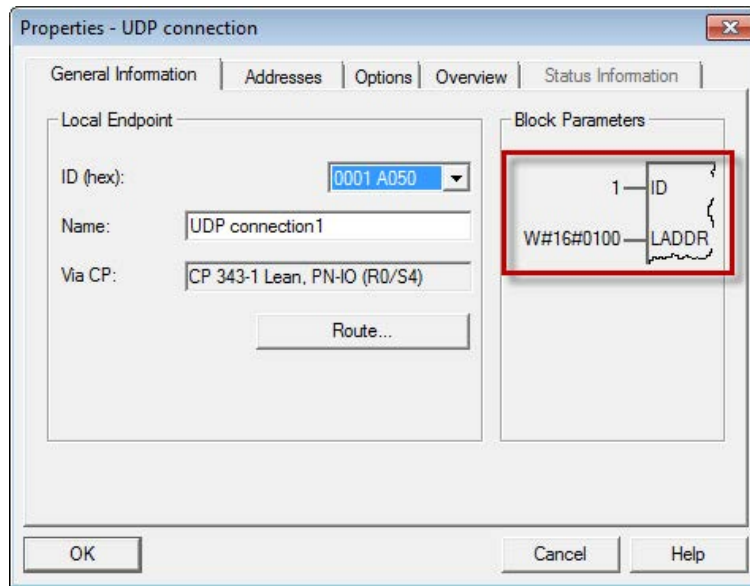


2. Select the connection partner "unspecified" and connection type "UDP connection".

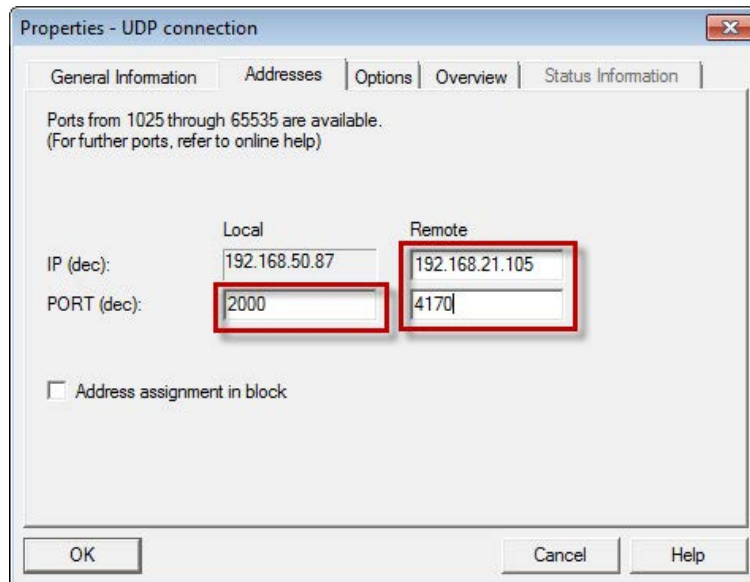


3. The automatically assigned function block parameter for the connection ID (ID) and the hardware starting address (LADDR) are needed later in step 7.

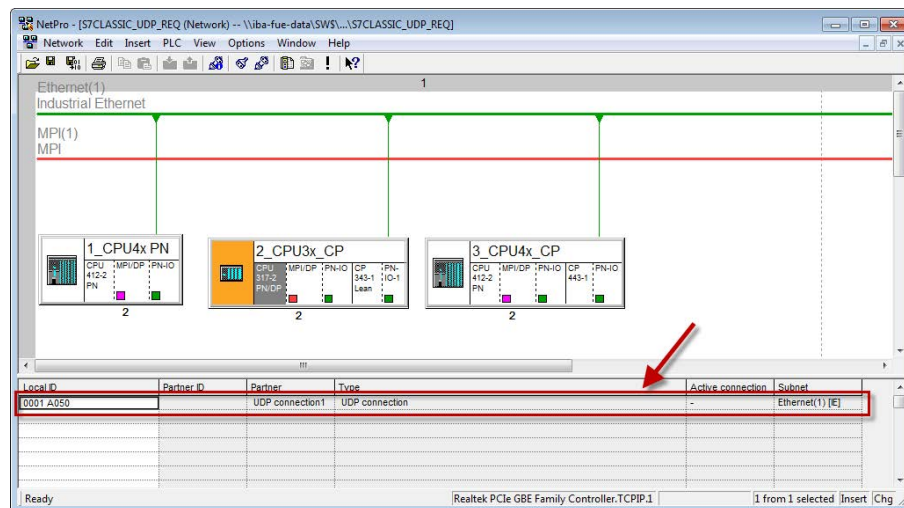




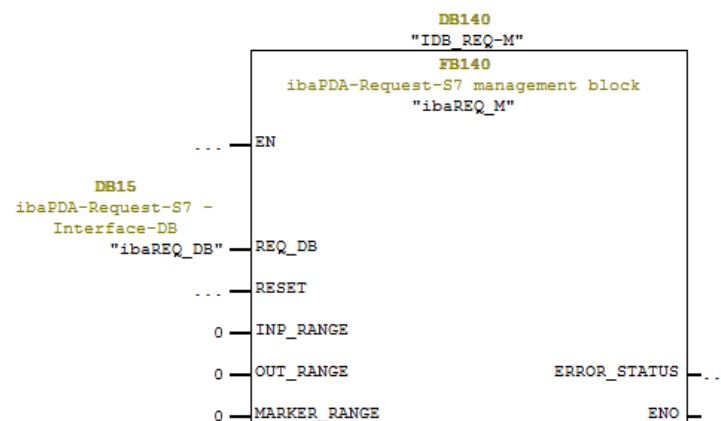
4. Enter the IP address of the *ibaPDA* computer as partner IP address and the configured port number (standard: 4170) and select a unique local port number.



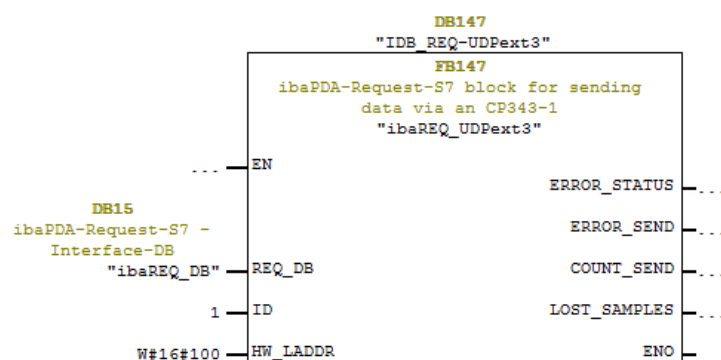
5. The connection table of the CPU now displays the newly configured connection.



6. Call of ibaREQ\_M (FB140), preferably within the OB1.



7. Call of ibaREQ\_UDPext3 (FB147), preferably within a cyclic interrupt OB (OB3x)



#### For each additional Request module:

- In the blocks folder, a data block ibaREQ\_DB (DB15) has to be available for each Request module. Copy the data block and assign a new unique DB number.
- Configure for each Request module a separate connection (steps 1-4). Assign different local port numbers.
- Within the OB1 the ibaREQ\_M (FB140) has to be called once more for each Request module with a new DB number (input REQ\_DB) (step 6).

- Within a cyclic interrupt alarm call OB (OB3x), the ibaREQ\_UDPext3 (FB147) has to be called once more for each Request module with the new DB number (input REQ\_DB) (step 7).
- Please consider that all instance data blocks have to be unique and that the values for the ID and HW\_LADDR parameters are to be assigned uniquely.

#### Finish:

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

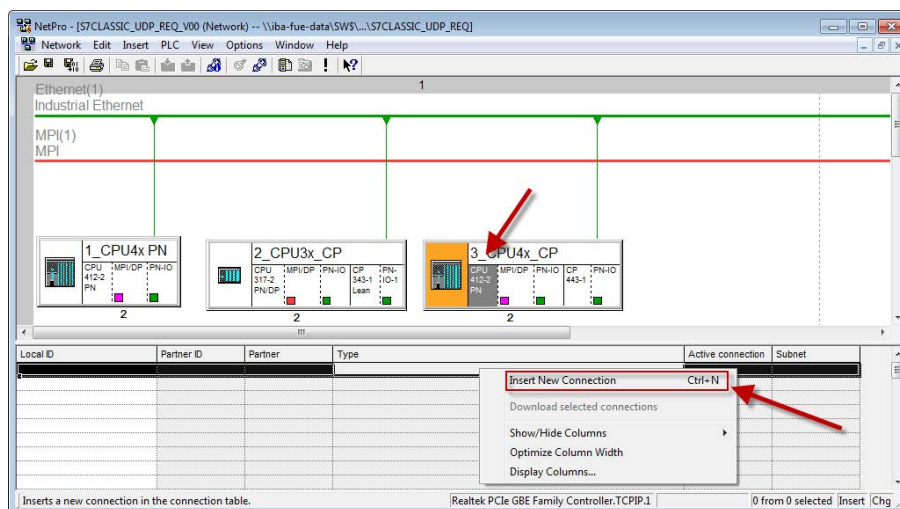
### 3.2.1.2.3 CPU S7-400 with CP443-1

The following blocks are required:

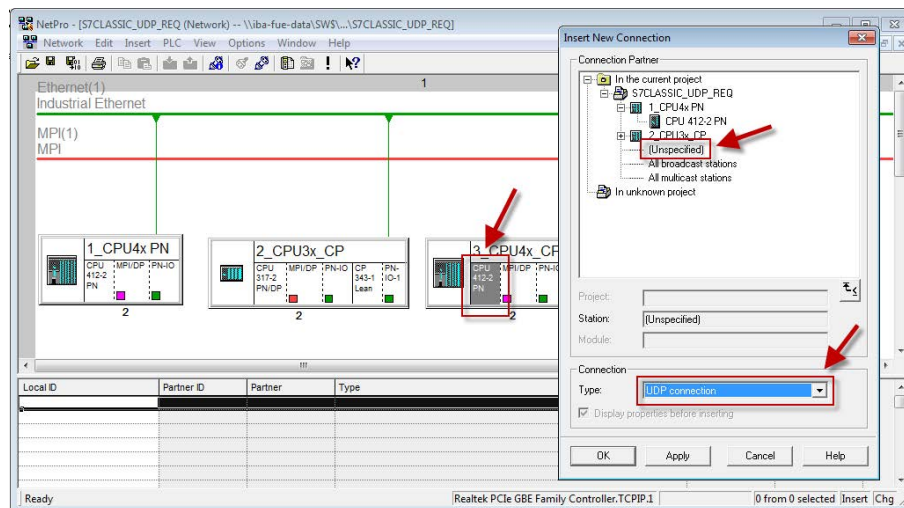
- ibaREQ\_M
- ibaREQ\_UDPact
- ibaREQ\_UDPext4
- ibaREQ\_DB
- ibaUDT\_UDPact (UDT145)

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

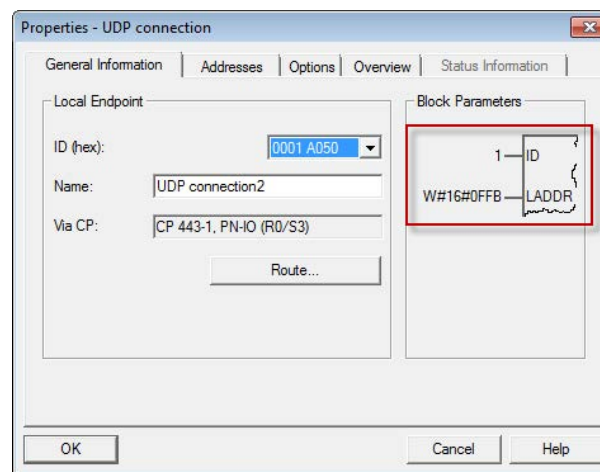
1. Configure a new connection in NetPro.



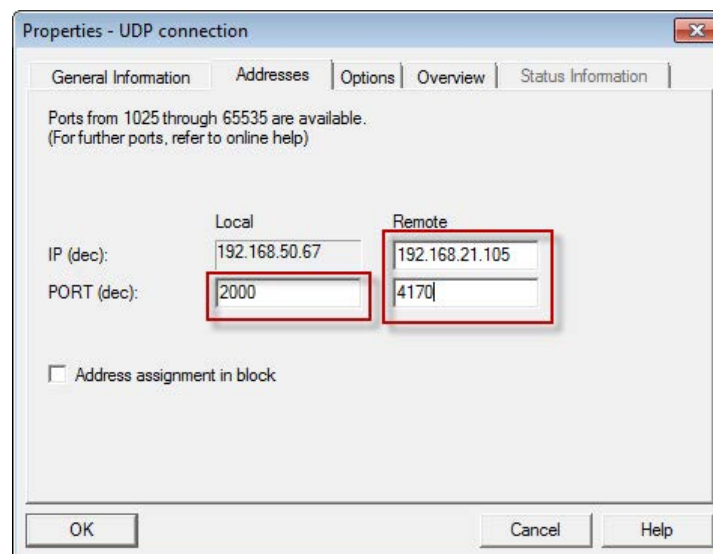
2. Select the connection partner "unspecified" and connection type "UDP connection".



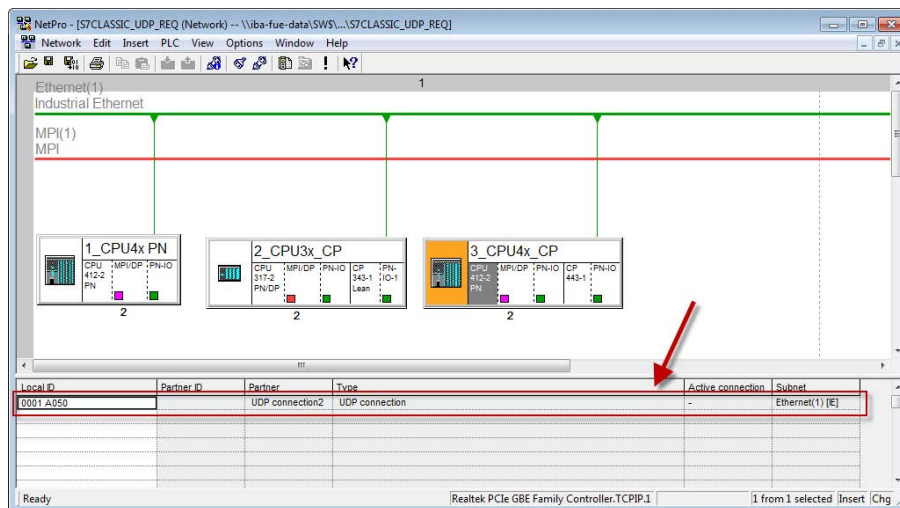
3. The automatically assigned function block parameters are needed later in step 7.



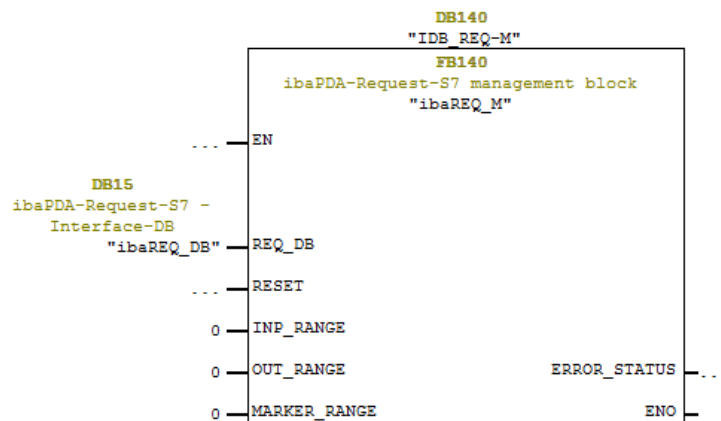
4. Enter the IP address of the *ibaPDA* computer as partner IP address and the configured port number (standard: 4170) and select a unique local port number.



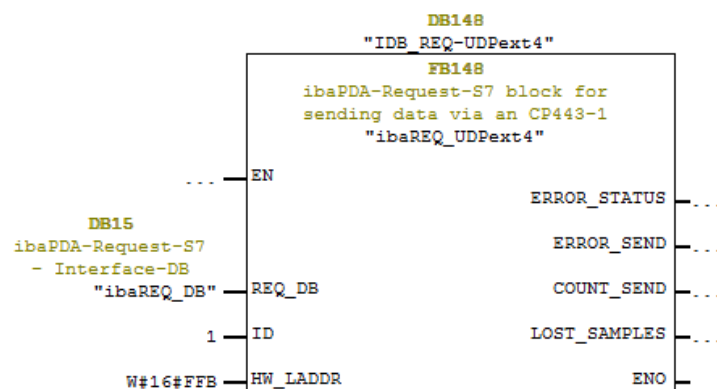
5. The connection table of the CPU now displays the newly configured connection.



6. Call the ibaREQ\_M (FB140), preferably within the OB1.



7. Call the ibaREQ\_UDPext4 (FB148), preferably in the context of a cyclic interrupt OB (OB3x)



#### For each additional Request module:

- In the blocks folder, a data block ibaREQ\_DB (DB15) has to be available for each Request module. Copy the data block and assign a new unique DB number.
- Configure for each Request module a separate connection (steps 1-4). Assign different local port numbers.
- Within the OB1, the ibaREQ\_M (FB140) has to be called once more for each Request module with the new DB number (input REQ\_DB) (step 6).

- Within a cyclic interrupt OB (OB3x), the ibaREQ\_UDPext4 (FB148) has to be called once more for each Request module with a new DB number (input REQ\_DB) (step 7).
- Please consider that all instance data blocks have to be unique and that the values for the ID and HW\_LADDR parameters are to be assigned uniquely.

**Finish:**

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

### 3.2.2 Configuration and engineering SIMATIC S7-1500

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

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

#### 3.2.2.1 Description of the Request blocks

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

A set of Request blocks has to be called for each Request module (connection).

The used blocks are part of the iba S7 library (see chapter [Iba S7 library](#), page 75).

Request block	S7-1500 CPU with integrated PN interface	S7-1500 CPU with CP1543-1 or CM1542-1	recommended call up level
ibaREQ_M (FB1400)	X	X	OB1
ibaREQ_UDP2 (FB1406)	X	X	OB3x
ibaREQ_UDPact (FB 1410)	X	X	OB3x
ibaREQ_UDP (FB1405)	X	X	OB3x
ibaREQ_DB (DB15)	X	X	-
ibaREQ_DB Interface	X	X	-

Table 3: Combinations of Request blocks

- **ibaREQ\_M (Management)**  
The block realizes the communication with *ibaPDA*. Ideally, the block is called in the OB1.
- **ibaREQ\_UDP2 (provides and sends the current signal values)**  
The block provides the actual signal values in the send cycle. Ideally, the block is called in a cyclic interrupt OB.
- **ibaREQ\_UDPact**  
The block is used internally by ibaREQ\_UDP2.
- **ibaREQ\_UDP**  
Similar function to ibaREQ\_UDP2. The block is only available for compatibility reasons. It is recommended to use the block ibaREQ\_UDP2.
- **ibaREQ\_DB (interface DB)**  
This DB serves as interface to *ibaPDA* and between the different Request blocks.

### 3.2.2.1.1 ibaREQ\_M (FB1400)

#### Description of the formal parameters:

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

The following SIMATIC standard blocks are used internally:

#### ■ GET\_IM\_DATA (FB801)

#### Detailed description:

##### REQ\_DB

The data is exchanged with *ibaPDA* via this DB. The identical DB must be configured for all related Request blocks.

The length of the data block is fixed.

##### RESET

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

##### CPU\_HW\_ID

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

##### ERROR\_STATUS

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

A list of all possible error codes can be found in chapter [↗ Error codes of Request blocks](#), page 100

### 3.2.2.1.2 ibaREQ\_UDP2 (FB1406)

#### Description of the formal parameters:

Name	Type	Data type	Description
INTERFACE_ID	IN	HW_ANY	Hardware identifier of the used interface
CON_ID	IN	CONN_OUC	Unique connection ID of the send block (TSEND_C)
LOCAL_PORT	IN	UINT	Local port number
RESET_CON	IN	BOOL	TRUE -> reset of the communication connection



Name	Type	Data type	Description
REQ_DB	INOUT	DB_ANY	DB of the ibaPDA communication interface ibaREQ_DB
ERROR_STATUS	OUT	WORD	Internal error code
LOST_SAMPLES	OUT	UNIT	Counter for lost measurement values

The following SIMATIC standard blocks are used internally:

- TCON
- TUSEND
- TDISCON

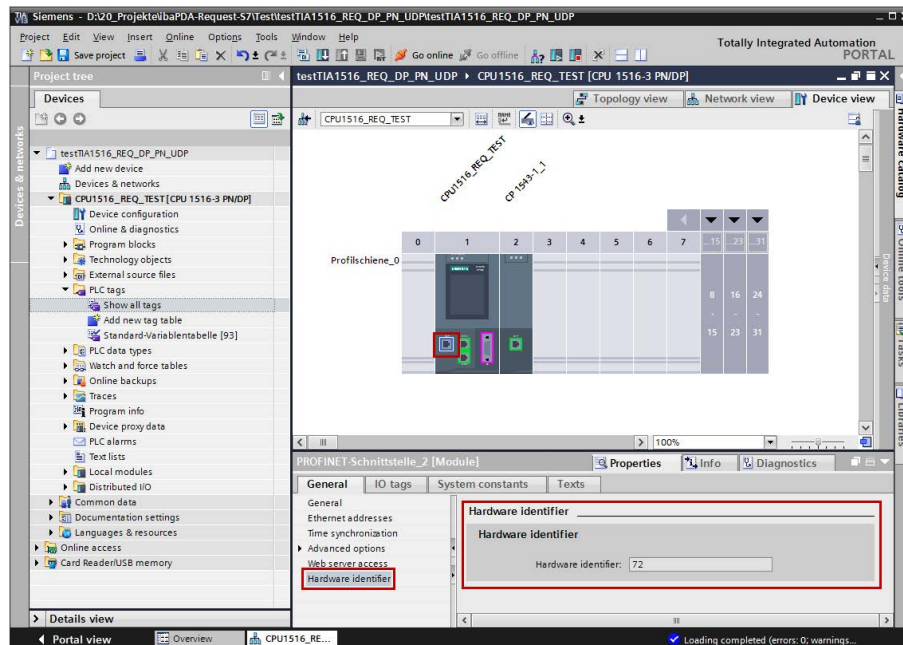
**Detailed description:**

**INTERFACE\_ID**

Hardware identifier of the used interface.

**Tip**

You find the hardware identifier of the marked interface under *Properties - General - Hardware identifier*



The hardware identifier could be configured as a numerical value or as a system constant of the type `Hw_Interface`. You find the system constant under *Properties - System constants*. Always use the system constant of the interface and not of a port, or of the IO system.

PROFINET-Schnittstelle_2 [Module]				
Properties				
System constants				
Name	Type	Hardware identi.	Comment	
Local-PROFINET-Schnittstelle_2	Hw_Interface	72		
Local-PROFINET-Schnittstelle_2-Port_1	Hw_Interface	73		

**CON\_ID**

Unique reference to the connection to be set up, value range: 1 to 4095.

**LOCAL\_PORT**

Number of the locally used port

**RESET\_CON**

Serves for manually resetting the communication connection.

**REQ\_DB**

The data are exchanged with *ibaPDA* via this DB. The identical DB must be configured for all related Request blocks.

**ERROR\_STATUS**

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

A list of all possible error codes can be found in chapter [➤ Error codes of Request blocks](#), page 100

### LOST\_SAMPLES

The counter is incremented with every call of the block if no new UDP telegram can be sent to *ibaPDA*, as the previous send order has not been finished, yet. A continuously rising value indicates a shortage in the communication performance.

#### 3.2.2.1.3 ibaREQ\_UDP (FB1405)

**Description of the formal parameters:**

Name	Type	Data type	Description
INTERFACE_ID	IN	HW_ANY	Hardware identifier of the used interface
CON_ID	IN	CONN_OUC	Unique connection ID of the send block (TSEND_C)
LOCAL_PORT	IN	UINT	Local port number
RESET_CON	IN	BOOL	TRUE -> reset of the communication connection
REQ_DB	INOUT	DB_ANY	DB of the ibaPDA communication interface ibaREQ_DB
ERROR_STATUS	OUT	WORD	Internal error code
ERROR_TSEND	OUT	WORD	Error code send block
COUNT_TSEND	OUT	WORD	Telegram counter of the send block
LOST_SAMPLES	OUT	UNIT	Counter for lost measurement values

The following SIMATIC standard blocks are used internally:

#### ■ TSEND\_C (FB1030)

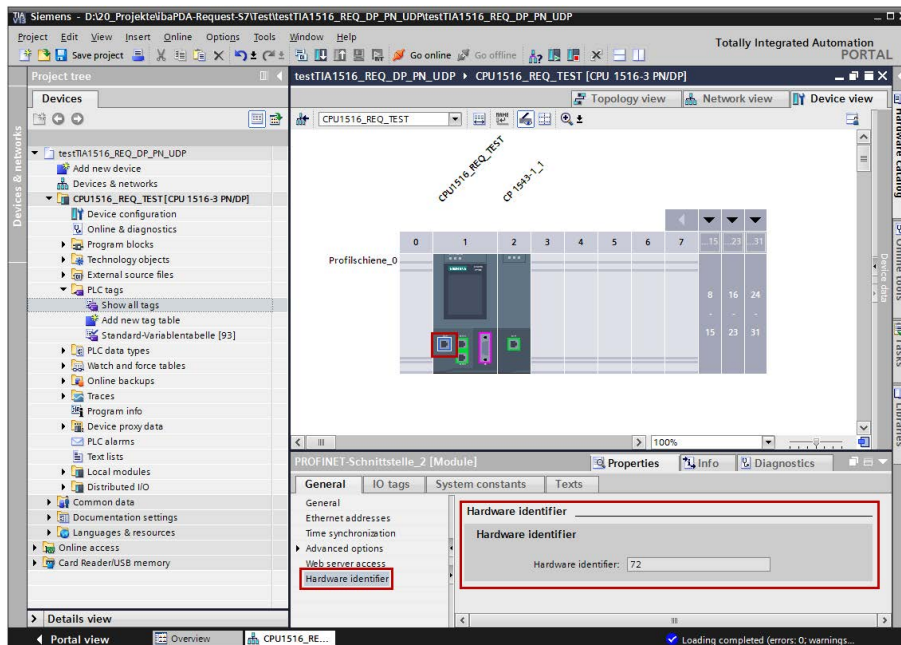
**Detailed description:**

#### INTERFACE\_ID

Hardware identifier of the used interface.

**Tip**

You find the hardware identifier of the marked interface under *Properties - General - Hardware identifier*



The hardware identifier could be configured as a numerical value or as a system constant of the type `Hw_Interface`. You find the system constant under *Properties - System constants*. Always use the system constant of the interface and not of a port, or of the IO system.

PROFINET-Schnittstelle_2 [Module]				
Properties				
System constants				
Name	Type	Hardware identi.	Comment	
Local-PROFINET-Schnittstelle_2	Hw_Interface	72		
Local-PROFINET-Schnittstelle_2-Port_1	Hw_Interface	73		

**CON\_ID**

Unique reference to the connection to be set up, value range: 1 to 4095.

**LOCAL\_PORT**

Number of the locally used port

**RESET\_CON**

Serves for manually resetting the communication connection.

**REQ\_DB**

Via this DB, the data are exchanged with *ibaPDA*. The identical DB must be configured for all related Request blocks.

**ERROR\_STATUS**

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

A list of all possible error codes can be found in chapter [➤ Error codes of Request blocks](#), page 100.

### ERROR\_TSEND

Error code when sending Standard error codes of the TSEND\_C block are issued.

0X8yyy errorcode of TSEND\_C block will be passed.

### LOST\_SAMPLES

The counter is incremented with every call of the block if no new UDP telegram can be sent to *ibaPDA*, as the previous send order has not been finished, yet. A continuously rising value indicates a shortage in the communication performance.

## 3.2.2.2 Configuration in STEP 7

In the following, we describe the configuration of the Request blocks in STEP 7.

### Note



The request blocks do not support multi-instance calls.

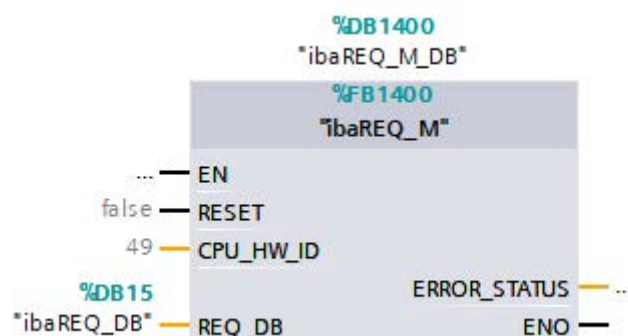
Copy the required blocks from the iba S7 library (see chapter [➤ Iba S7 library](#), page 75) to the blocks folder of your STEP 7 project.

The following blocks are required:

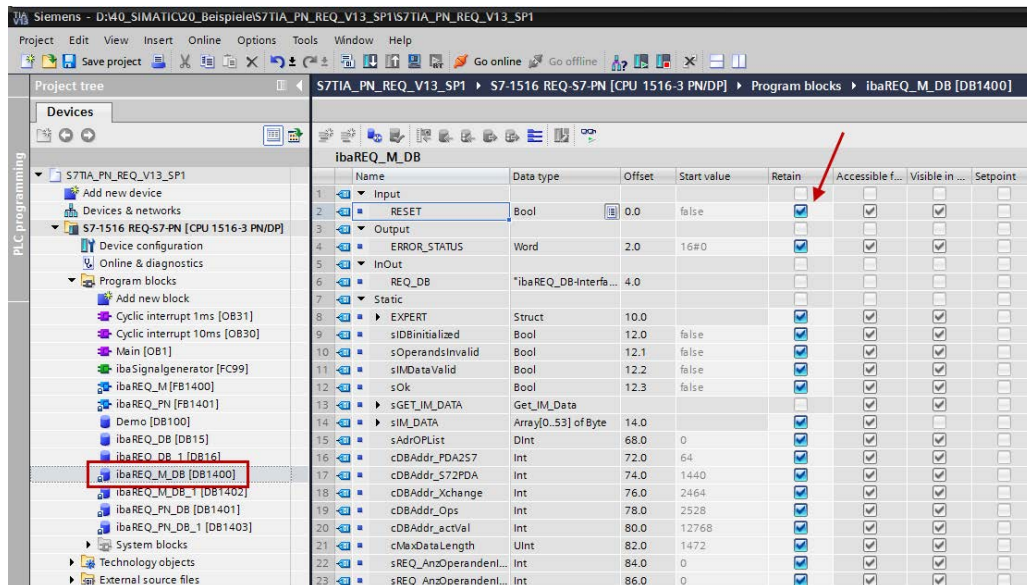
- ibaREQ\_M (FB1400)
- ibaREQ\_UDP2 (FB1406)
- ibaREQ\_UDPact (FB1410)
- ibaREQ\_DB (DB15)
- ibaREQ\_DB-Interface (PLC data type)

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

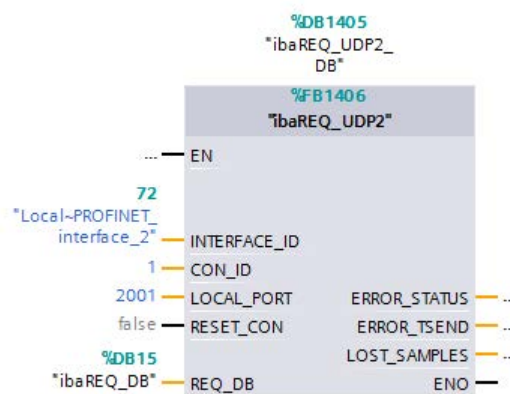
1. Call the ibaREQ\_M (FB1400) preferably within the OB1



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



3. Call the ibaREQ\_UDP2 (FB1406), preferably within a cyclic interrupt OB (OB3x)



#### For each additional Request module:

- In the blocks folder, a data block ibaREQ\_DB (DB15) has to be available for each Request module. Copy the data block and assign a new unique DB number.
- Within the OB1, the ibaREQ\_M (FB1400) has to be called for each Request module with the new DB number once more.
- Within the cyclic interrupt OB (OB3x), the ibaREQ\_UDP2 (FB1406) has to be called for each Request module with the new DB numbers once more.
- Please consider that all instance data blocks have to be unique and that the values for the CON\_ID and LOCAL\_PORT are assigned uniquely.

#### Finish:

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

### 3.2.2.3 Device configuration

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

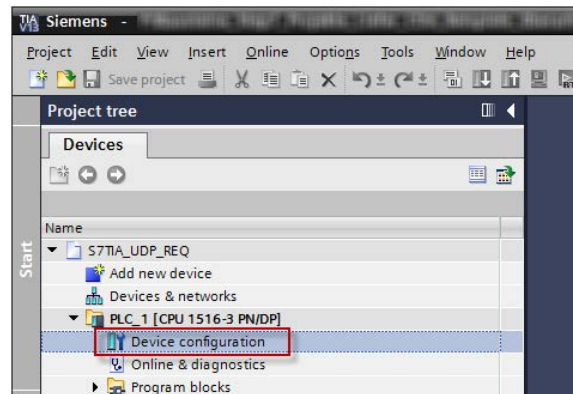
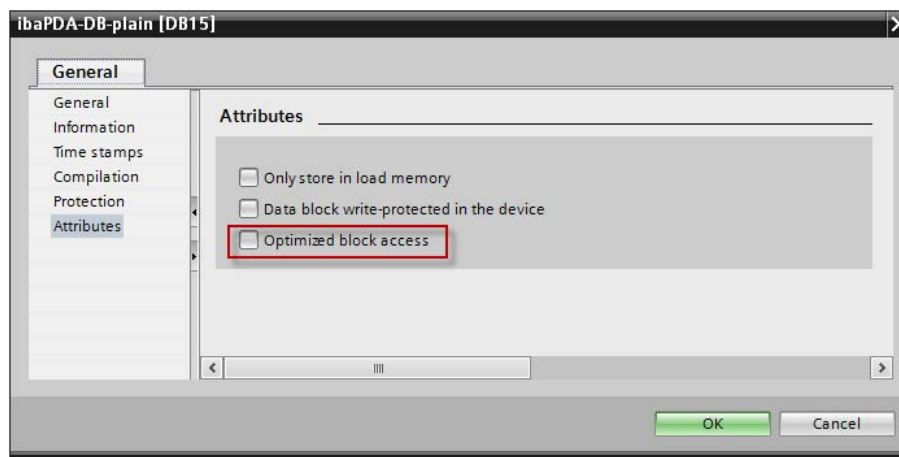


Fig. 2: Device configuration

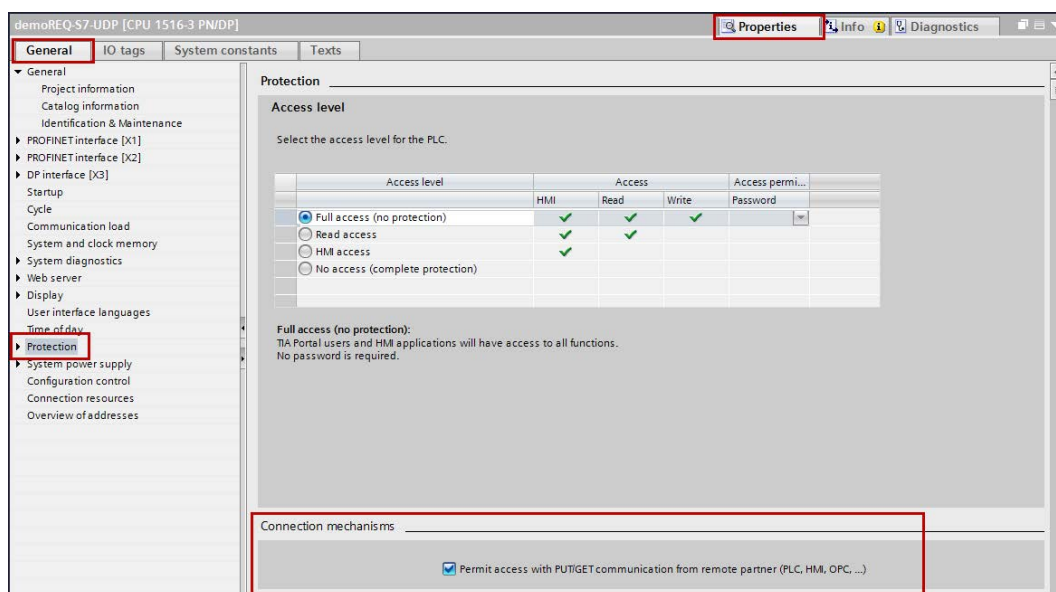
- Disable ibaREQ\_DB (DB15) Properties – Attributes – Optimized block access



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

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





### Access protection S7-1500

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

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

#### Note



The configuration of the UDP connection is carried out programm-controlled in the Request block. It is not allowed to configure manually a UDP connection and use it for this purpose.



## 3.3 Configuration and engineering ibaPDA

### 3.3.1 General interface settings

The interface ibaPDA-Interface-S7-TCP/UDP is configured in the ibaPDA "I/O Manager". If all system requirements are met, the "S7 TCP/UDP" data interface is displayed in the interface tree. *ibaPDA-Request-S7-UDP* is a module ("S7 Request") of this interface.

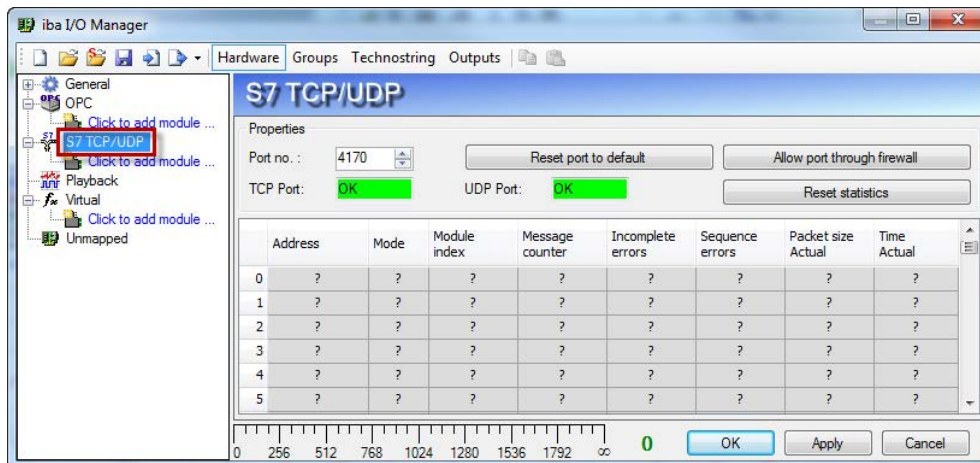


Fig. 3: Interface in the I/O Manager

The interface has the following functions and configuration options:

#### Port no.

Port used on the PC side

The port no. has to be used identically in the S7 connection configuration.

With <Reset to default port> the setting under Port no. can be reset to the default setting.

Via <Allow port through firewall>, the Firewall exceptions can be reactivated. This is necessary, for example, after changing the port number. By default, the settings are already made during installation.

With <Reset statistics>, you can reset the counter.

All active connections are entered in the table.

With a double-click on a row, you jump to the configuration of the corresponding module.

#### Other documentation



For more information about the interface *ibaPDA-Interface-S7-TCP/UDP*, see the corresponding manual.

For information about the connection diagnostics, see chapter [↗ Diagnostics](#), page 71.

### 3.3.2 Add module

Add a module by clicking below the interface. Select the desired module type and click on <OK>.

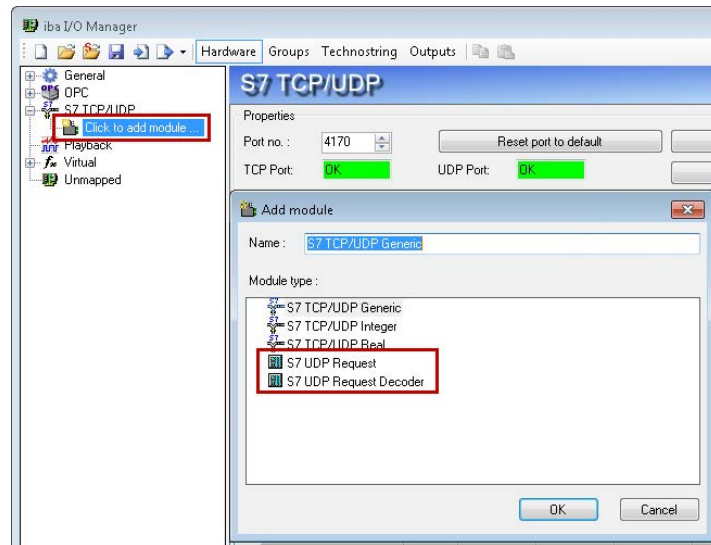


Fig. 4: Add module

Module name	Description
S7 UDP Request	Request module for a max. of 1024 analog and 1024 digital signals.
S7 UDP Request Decoder	Request module for a max. of 11728 signals which are transmitted as 733 words (1466 Byte).

Table 4: Module overview of the Request-S7-UDP interface

### 3.3.3 General module settings

All modules have the following common setting options:

The screenshot shows the 'S7 UDP Request (0)' configuration window. It has a title bar with the text 'S7 UDP Request (0)' and a tabbed interface with tabs for 'General', 'Connection', 'Analog', 'Digital', and 'Diagnostics'. The 'General' tab is selected, showing a tree view with sections: 'Basic', 'Connection', 'Module Layout', and 'S7'. The 'Basic' section contains fields for 'Module Type' (S7 UDP Request), 'Locked' (False), 'Enabled' (True), 'Name' (S7 UDP Request), 'Module No.' (0), 'Timebase' (10 ms), and 'Use name as prefix' (False). The 'Connection' section contains 'Module Index' (200) and 'Auto enable/disable' (False). The 'Module Layout' section contains 'No. analog signals' (32) and 'No. digital signals' (32). The 'S7' section contains 'CPU Name' with a 'No addressbook' button. Below the tree view is a 'Name' field with the description 'The name of the module.' and two links: 'Select S7 operands' and 'Manage addressbooks'.

Fig. 5: General module settings

#### Basic settings

##### Module type

Display of the module type (read only)

##### Locked

A module can be locked for preventing accidental or unauthorized changes of the module settings.

##### Enabled

Disabled modules are excluded from signal acquisition.

##### Name

Here, the clear text name of the module designation has to be entered.

##### Module No.

Sequential number that is assigned by *ibaPDA* in ascending order but can be changed by the user. This number serves to identify signals unambiguously, e.g. in expressions and *ibaAnalyzer*.

##### Timebase

All signals of this module are acquired with this timebase.

**Use name as prefix**

Puts the module name in front of the signal name.

**Module Layout****No of analog/digital signals**

Defines the number of configurable analog/digital signals in the signal tables. A maximum of 1024 signals for each are allowed.

**S7****CPU Name**

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

This requires that address books have already been generated. Otherwise, the selection list is empty. When clicking on „Create address book...” in the selection list, you get directly to the address book generator (see chapter [↗ Selection via the operand symbols](#), page 55).

**Connection****Module index**

Internal reference number of the module (read only).

**Auto enable/disable**

If TRUE, the acquisition is started even when no connection to the S7-CPU can be established. The module is disabled. During the acquisition, *ibaPDA* tries to reconnect to the S7-CPU. When it succeeds, it will automatically restart the acquisition.

If FALSE, the acquisition is not started, in case no connection could be established to the configured S7-CPU.

**3.3.4 Connection settings**

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

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

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

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

### 3.3.4.1 Connection mode TCP/IP

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

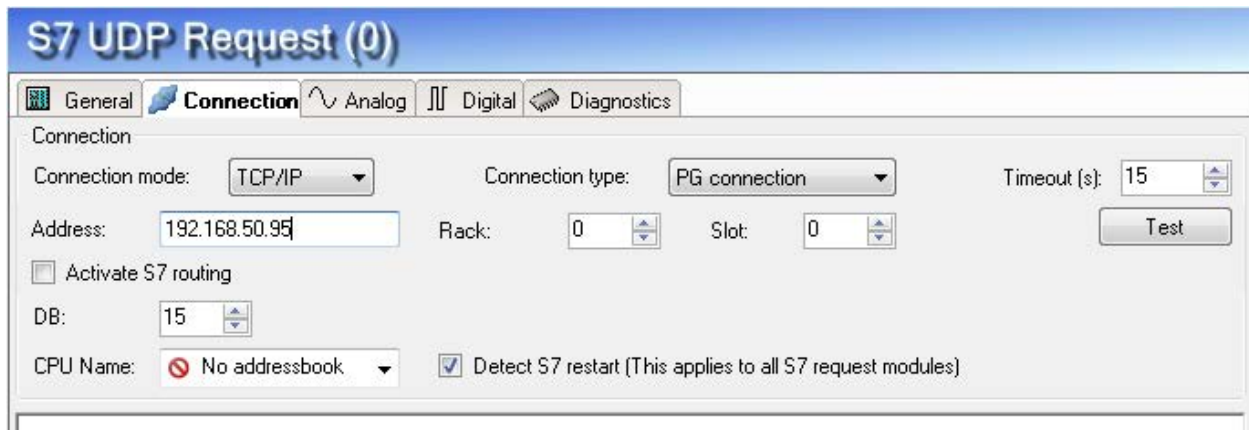


Fig. 6: Connection configuration TCP/IP

#### CPU Name

Selection of the linked address book

#### Connection mode

Selection of the TCP/IP connection mode

#### Timeout

Timeout of the connection

#### Connection type

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

#### Address

IP address of the controller

#### Rack

Rack number of the controller (default: 0)

#### Slot

Number of the slot of the controller in the rack (for S7-1500 CPUs: 0)

#### Test

Connection test to the CPU and available diagnostic data are issued

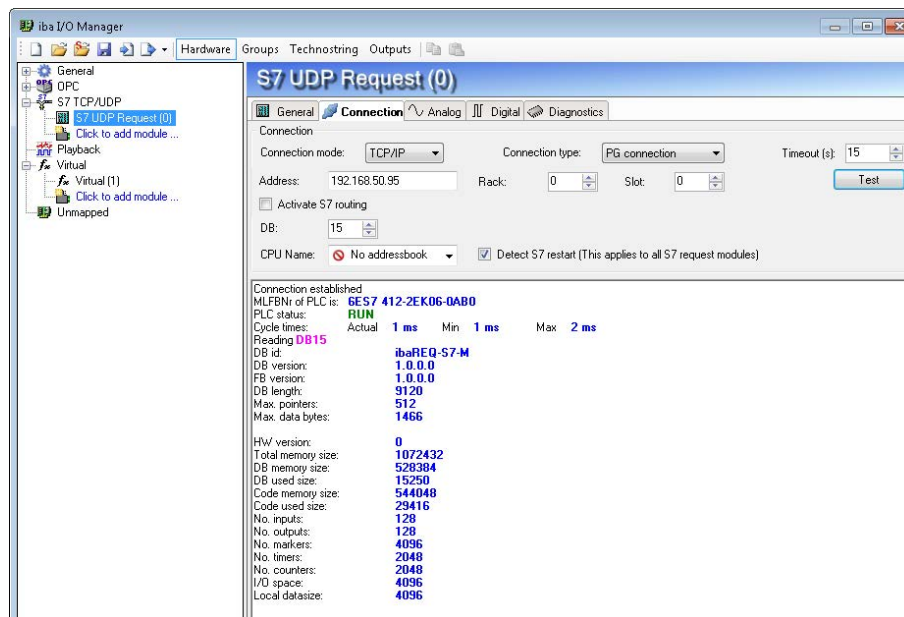


Fig. 7: Connection test / diagnostic data

### Tip



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

Please check if...

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

### Activate S7 routing

(see chapter [↗ S7 routing](#), page 92):

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

Two additional input fields appear

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

### Detect S7 restart:

The current request request configuration is stored in a data block on the CPU. In case the "Detect S7 restart" option is enabled, *ibaPDA* can detect if this data block has been deleted or overwritten, e. g. as a result of loading the offline program or due to a cold restart and will restart the data

acquisition. The configuration data are transferred again. This does not affect a warm restart of the CPU.

#### Note



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

### 3.3.4.2 Connection mode PC/CP

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

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

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

#### Note



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

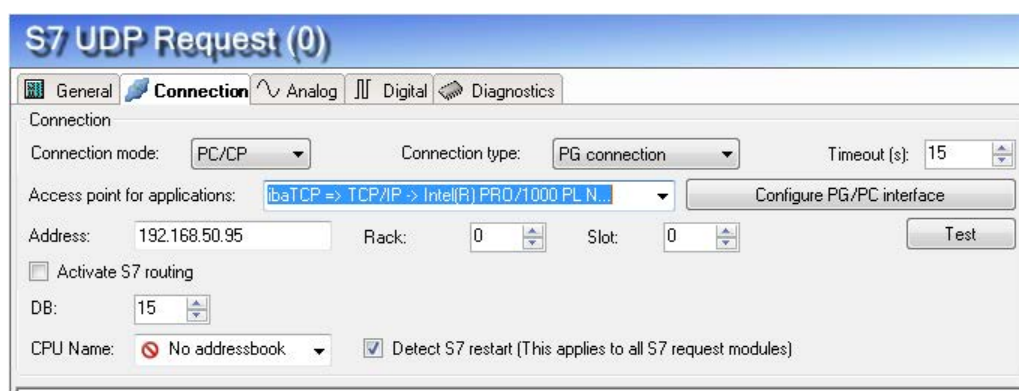


Fig. 8: Connection configuration PC/CP

#### CPU Name

Selection of the linked address book

**Connection mode**

Selection of the PC/CP connection mode

**Connection type**

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

**Timeout**

Timeout of the connection

**Access point for applications**

Selection of the access point that is to be used

For notes on creating and adapting an access point ➔ *Setting PG/PC interface / defining new access point*, page 88

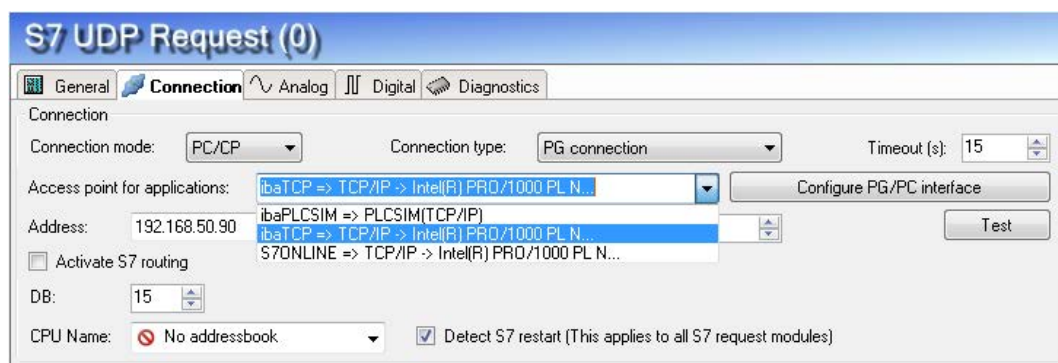


Fig. 9: Select access point

**Note**

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

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

**Configure PG/PC interface**

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

**Address**

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

**Rack**

Rack number of the controller (default: 0)



## Slot

Number of the slot of the controller in the rack (for S7-1500 CPUs: 0)

## Test

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

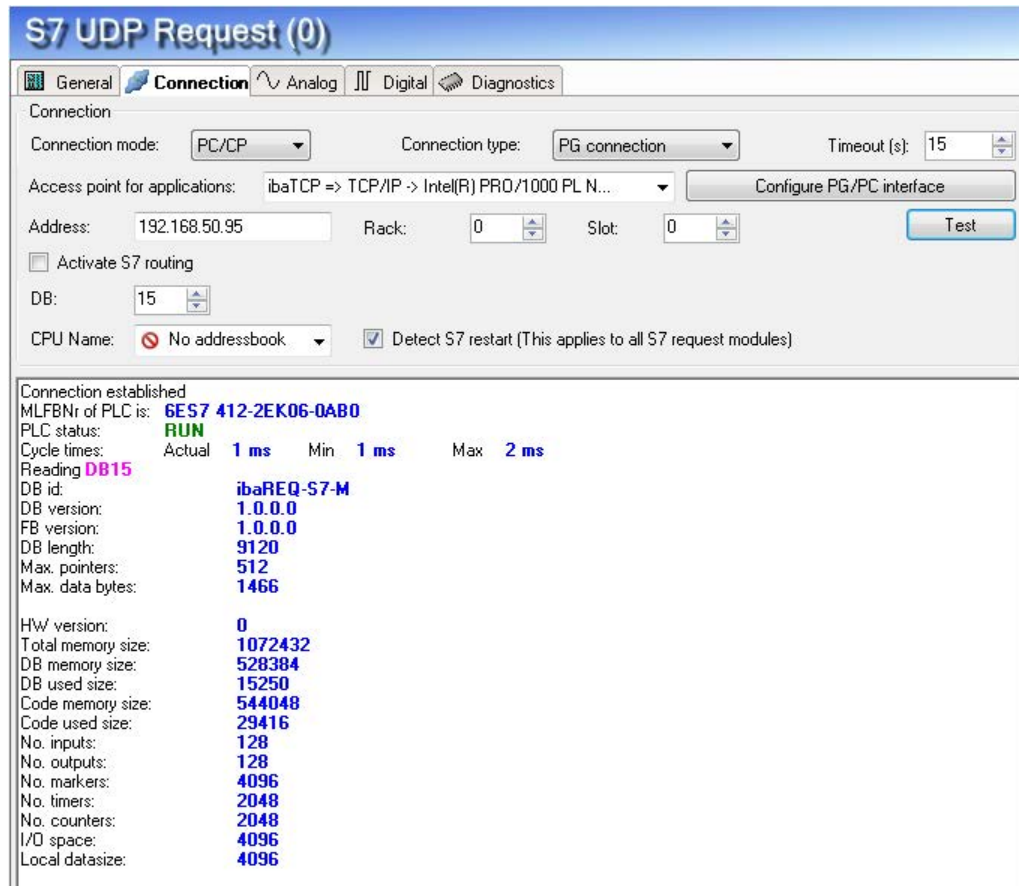


Fig. 10: Connection test / diagnostic data

## Tip



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

Please check if...

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

### Activate S7 routing

(see chapter ↗ *S7 routing*, page 92):

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

Two additional input fields appear

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

### Detect S7 restart

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

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

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

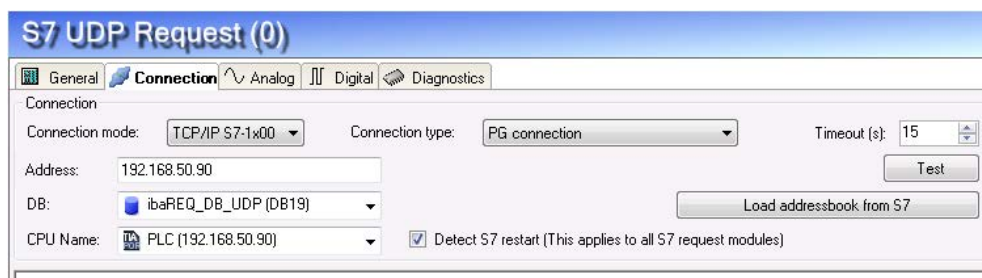


Fig. 11: Connection configuration TCP/IP S7-1x00

#### CPU Name

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

#### Connection mode

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

#### Connection type

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

#### Timeout

Timeout of the connection

#### Address

IP address of the controller

#### Test

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

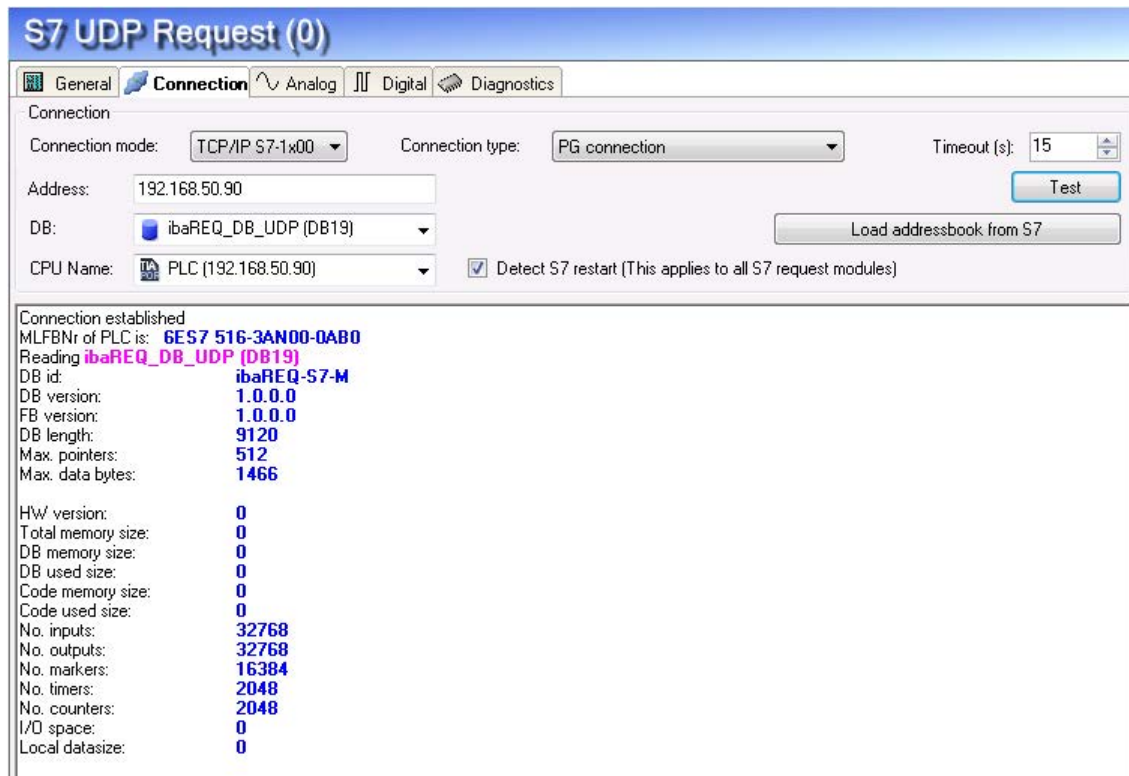
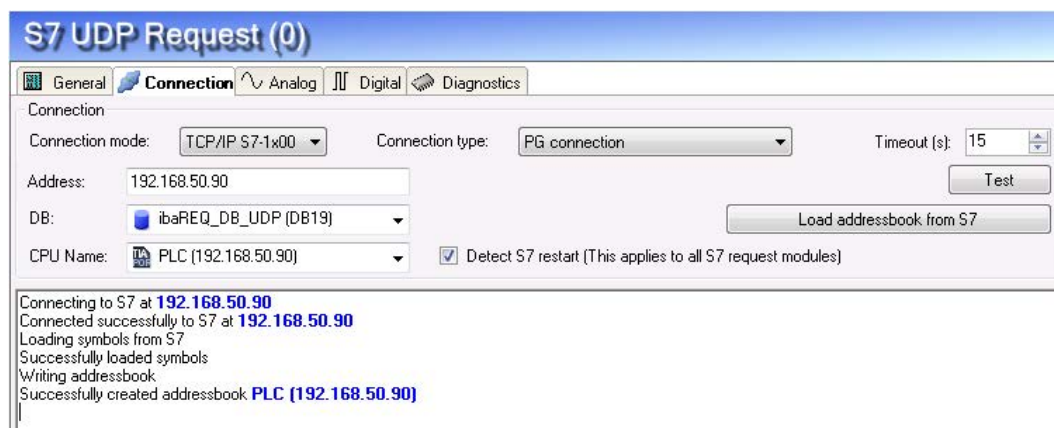


Fig. 12: Connection test / diagnostic data

### Load addressbook from S7

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



---

**Tip**

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

Please check if...

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

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

For further information, see ➤ *Device configuration*, page 39

**Detect S7 restart:**

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

---

**Note**

The "TCP/IP S7-1x00" connection mode does not support the use of absolute operands and addresses.

---

---

**Note**

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

---

### 3.3.5 Signal configuration

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

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

There are 3 options for selecting measurement values:

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

SIMATIC CPU	Access via absolute address	Access via symbol	Access via CFC connectors <sup>3)</sup>
S7-300	X	X	X
S7-400	X	X	X
WinAC	X	X	X
S7-1500	X	X	

Table 5: Possible types of selections

Supported operand ranges:

Operand ranges	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 <sup>4)</sup>

Table 6: Supported operand ranges

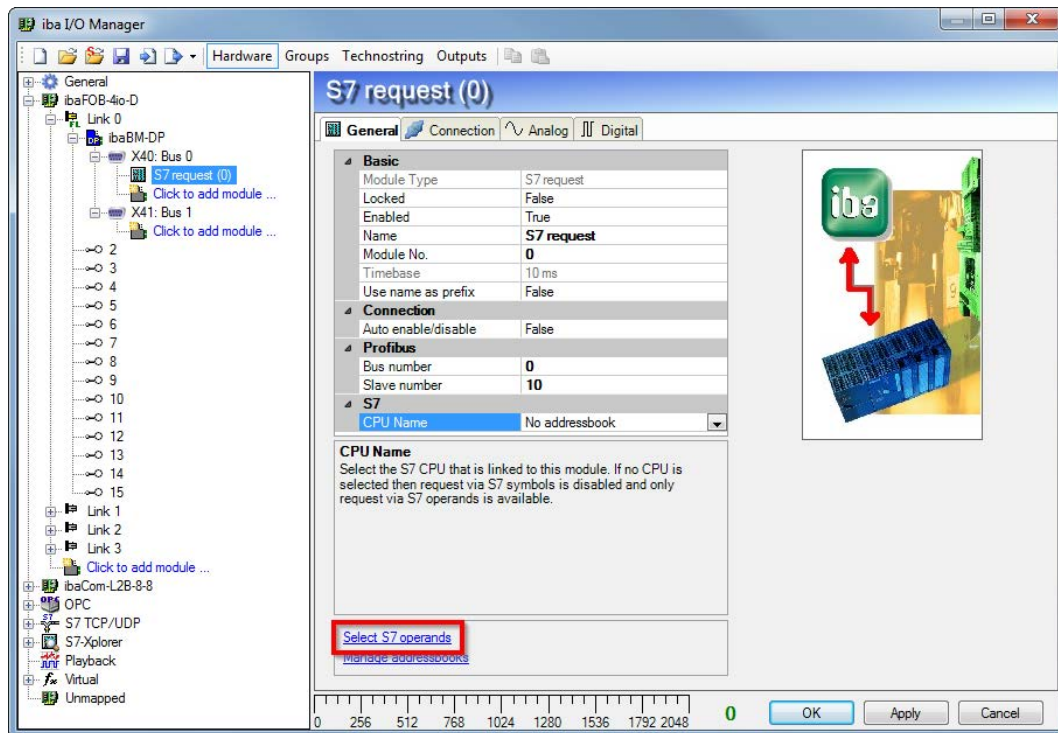
#### 3.3.5.1 Selection via the absolute address of the operands

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

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


<sup>3)</sup> Required is the use of the SIMATIC STEP 7 options package S7-CFC

<sup>4)</sup> The access to optimized data blocks of S7-1500 controllers is not supported.



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

	Name	Unit	Gain	Offset	S7 Operand	S7 DataType	Active
0	DB 100.DBD 0			1	0 DB 100.DBD 0	DWORD	<input checked="" type="checkbox"/>
1	DB 100.DBD 4			1	0 DB 100.DBD 4	DWORD	<input checked="" type="checkbox"/>
2	DB 100.DBD 8			1	0 DB 100.DBD 8	DWORD	<input checked="" type="checkbox"/>
3				1		INT	<input type="checkbox"/>
4				1		INT	<input type="checkbox"/>
5				1		INT	<input type="checkbox"/>

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

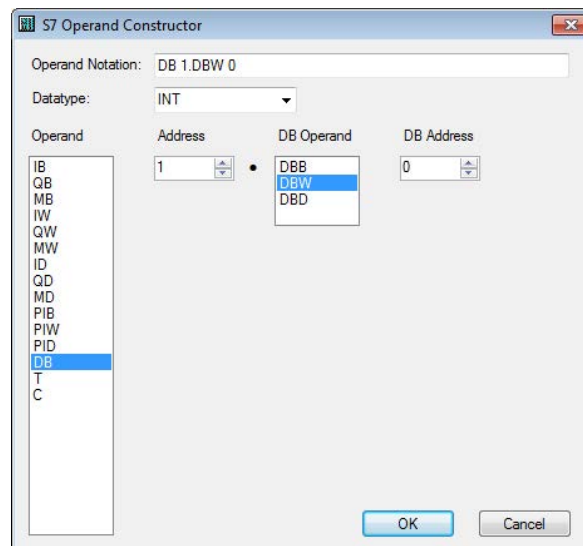


Fig. 13: S7 operand constructor

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

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

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

Fig. 14: Operand names

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

### 3.3.5.2 Selection via the operand symbols

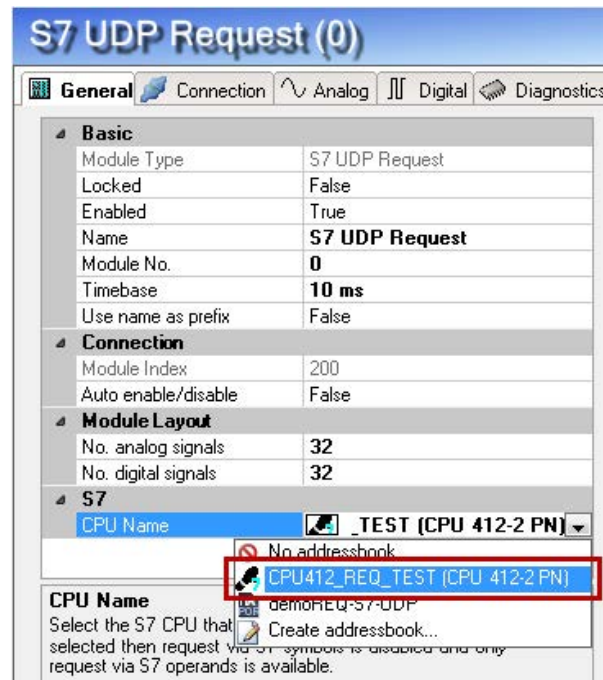
This method of access requires that the signals to be measured already have an entry in the S7 symbol table, the PLC variable list or in a data block and that an address book has been created (see chapter [Address books](#), page 66).

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

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

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



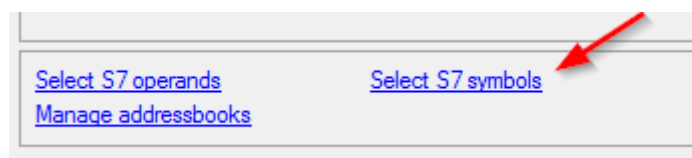


2. An additional "S7 symbol" column is displayed on the *Analog* and *Digital* tab.

Name	Unit	Gain	Offset	S7 Symbol	S7 Operand	S7 D...	Ac...
0		1	0			INT	<input type="checkbox"/>
1		1	0			DWORD	<input type="checkbox"/>
2		1	0			REAL	<input type="checkbox"/>
3		1	0			REAL	<input type="checkbox"/>
4		1	0			INT	<input type="checkbox"/>

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






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



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



S7 UDP Request (0)

<div><div><div> General</div><div> Connection</div><div> Analog</div><div> Digital</div><div> Diagnostics</div></div></div>								
	Name	Unit	Gain	Offset	S7 Symbol	S7 Operand	S7 DataType	Active
0	counter_16bit		1	0	SYMBOL\{counter_16bit	MW 100	WORD	<input checked="" type="checkbox"/>
1	counter_32bit		1	0	SYMBOL\{counter_32bit	MD 104	DWORD	<input checked="" type="checkbox"/>
2	sinus		1	0	SYMBOL\{sinus	MD 112	REAL	<input checked="" type="checkbox"/>
3	cosinus		1	0	SYMBOL\{cosinus	MD 116	REAL	<input checked="" type="checkbox"/>
4			1	0			INT	<input type="checkbox"/>

In the symbol browser, you have the following options:

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

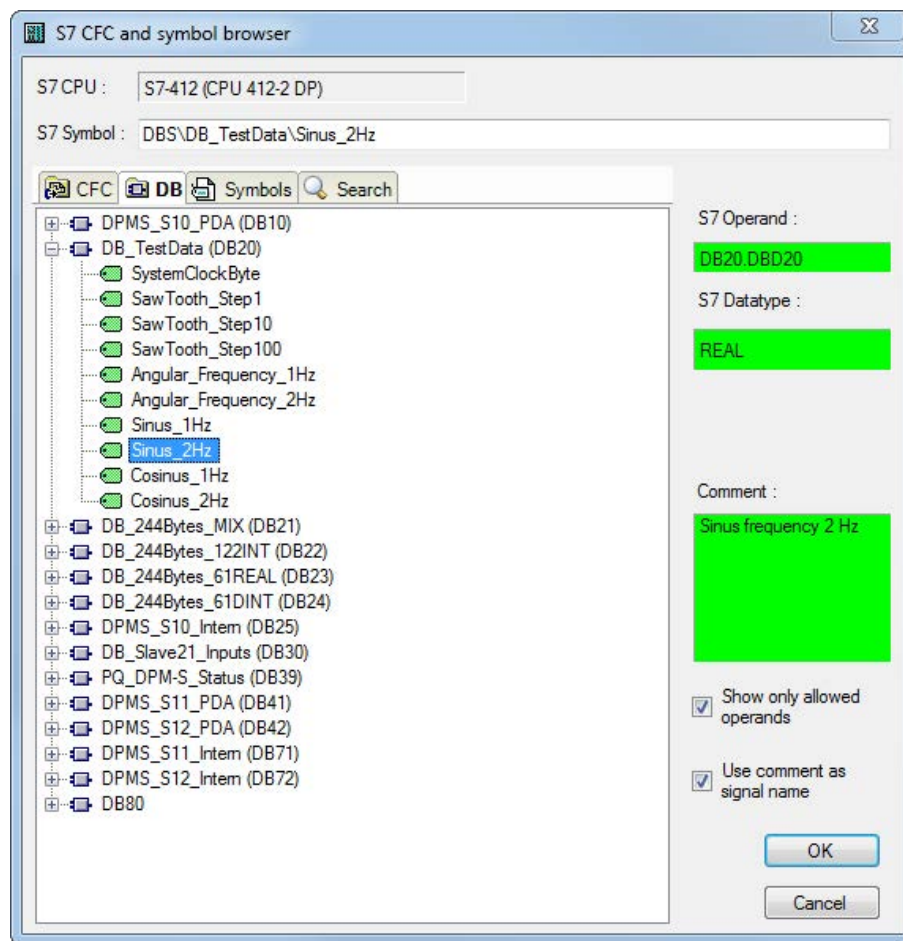


Fig. 15: S7-CFC and symbol browser

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

The variables are shown in the following colors:

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

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

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



### Show only allowed operands

If you enable this option, only the operands or symbols allowed for the current signal table from which you opened the browser are displayed in the tree structure (no red and no yellow operands). This improves clarity.

### Signal name, signal comment 1 and 2

Usually, the symbolic signal name from STEP 7 is adopted as signal name in the I/O Manager. With these three combo boxes you have the possibility to modify the signal name and the two comments.

Select the desired alternative from the available options. If a signal name or comment in the signal table should not be changed, select "Do not change".

### Lookup symbols in the signal grid

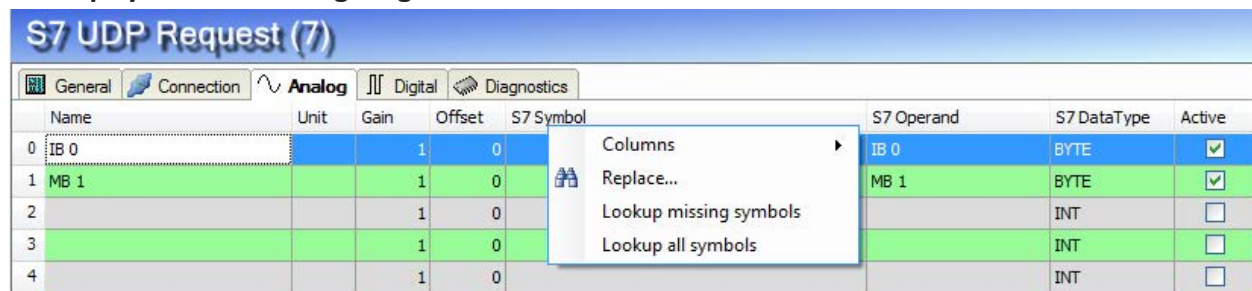


Fig. 16: Lookup symbols

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

### 3.3.5.3 Measuring the CFC connectors

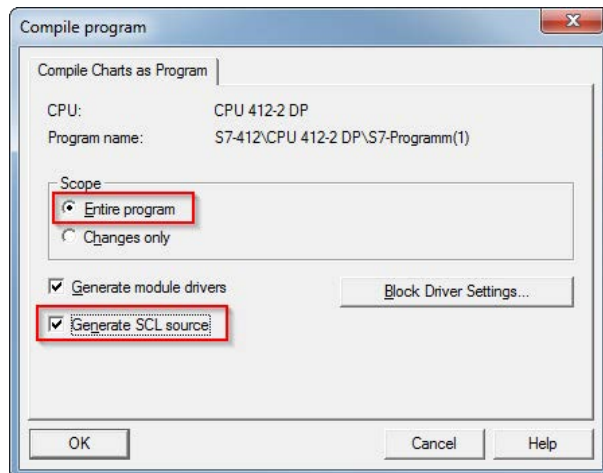
For selecting CFC connectors for the measurement process, proceed basically as described in chapter [↗ Selection via the operand symbols](#), page 55, until opening the S7-CFC and symbol browser.

---

**Note**

In case the connectors are not displayed on the *CFC* tab, then possibly the SCL sources have not been translated in the STEP 7 project.

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



Afterwards, go back and generate the address books again.

---

**Note**

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

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

---

In the *S7-CFC and symbol browser* you can select the *CFC* tab instead of the *Symbol* tab and select the signals. The connectors are listed hierarchically by chart name, module name and connector name:

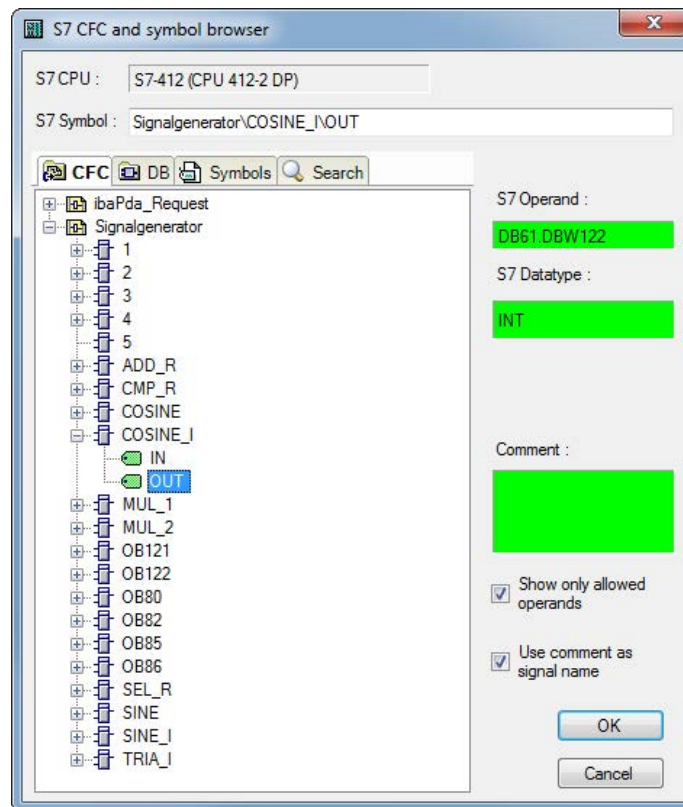


Fig. 17: S7-CFC and symbol browser

### Special function Drag & Drop

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

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

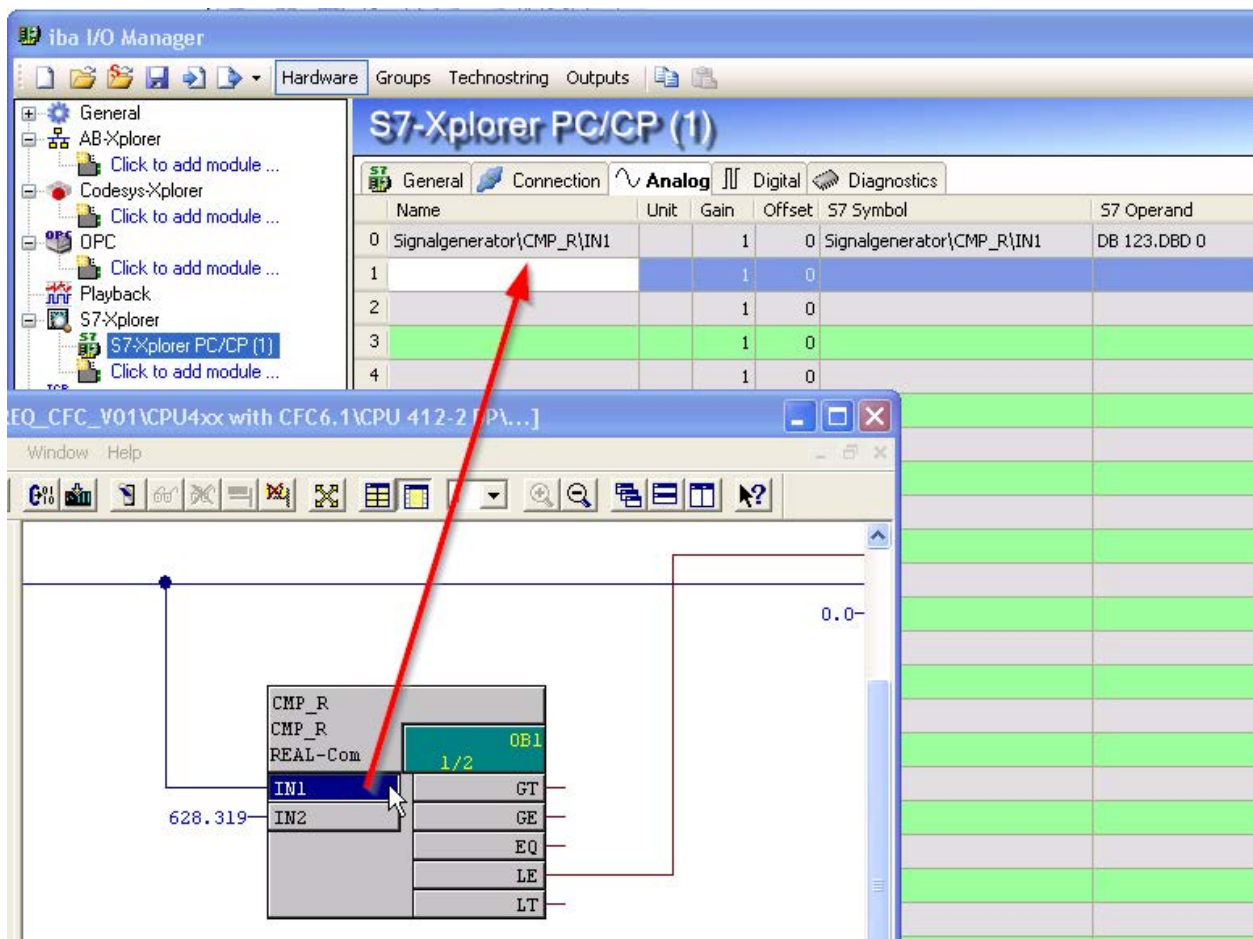


Fig. 18: Selecting signals via drag &amp; drop

Click the connector in the CFC chart, hold the mouse key down, drag it onto the I/O Manager and drop it in the desired signal row. Now, the CFC connector has been entered as measurement signal in the signal table of the Request module.



Fig. 19: Added signal

### 3.3.6 Module S7 Request

With the "S7 Request" module, analog and digital signals can be acquired.

In *ibaPDA*, up to 1024 analog and 1024 digital signals can be defined. Together, these may occupy a max. of 1466 Byte (max. length of the user data of an UDP telegram).



### 3.3.7 Module S7 Request Decoder

With the "S7 Request Decoder" module, up to 11728 digital signals can be acquired that are sent in form of a max. of 733 words (1466 Byte).

The operand to be addressed (absolute or symbolic) is of the WORD data type. Individual digital signals cannot be addressed.

On the *Digital* tab, you can directly enter the words as basic signals for the decoding via absolute S7 operands. S7 symbols can be used too by using address books.

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

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


# S7 request decoder (2)

General				Connection				Digital							
Connector				S7 Operand				DataType				Active			
0 + PEW 1				PEW 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"/>			
Q Name												Active			
0 Digital Signal 0												<input checked="" type="checkbox"/>			
1 Digital Signal 1												<input checked="" type="checkbox"/>			
2 Digital Signal 2												<input checked="" type="checkbox"/>			
3 Digital Signal 3												<input checked="" type="checkbox"/>			
4 Digital Signal 4												<input checked="" type="checkbox"/>			
5 Digital Signal 5												<input checked="" type="checkbox"/>			
6 Digital Signal 6												<input checked="" type="checkbox"/>			
7 Digital Signal 7												<input checked="" type="checkbox"/>			
8 Digital Signal 8												<input checked="" type="checkbox"/>			
9 Digital Signal 9												<input checked="" type="checkbox"/>			
10 Digital Signal 10												<input checked="" type="checkbox"/>			
11 Digital Signal 11												<input checked="" type="checkbox"/>			
12 Digital Signal 12												<input checked="" type="checkbox"/>			
13 Digital Signal 13												<input checked="" type="checkbox"/>			
14 Digital Signal 14												<input checked="" type="checkbox"/>			
15 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"/>			
5 +								WORD				<input type="checkbox"/>			
6 +								WORD				<input type="checkbox"/>			

Fig. 20: Module S7 Request Decoder

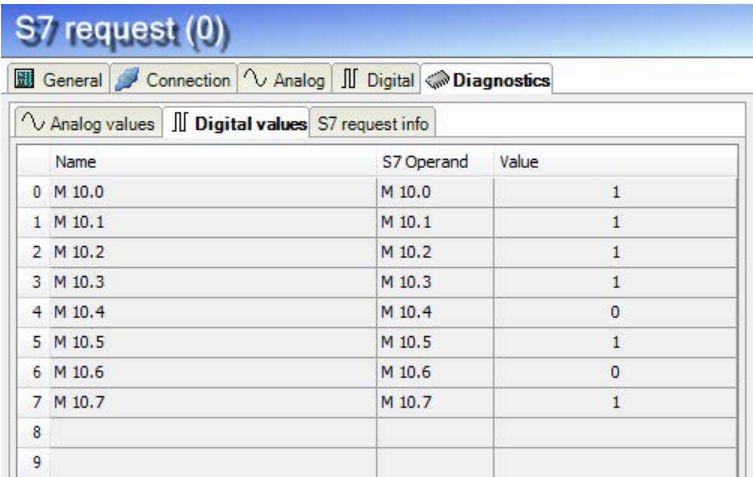
### 3.3.8 Module diagnostics

On the *Diagnostics* tab all configured operands are listed as a table with their data type and actual value.



Name	S7 Operand	Datatype	Value	Displaytype
0 DB 100.DBD 0	DB 100.DBD 0	REAL	20325	DEC
1 DB 100.DBD 4	DB 100.DBD 4	REAL	325	DEC
2 DB 100.DBD 8	DB 100.DBD 8	REAL	0,8910338	DEC
3 DB 100.DBD 12	DB 100.DBD 12	REAL	-0,4539368	DEC
4		INT		DEC
5		INT		DEC

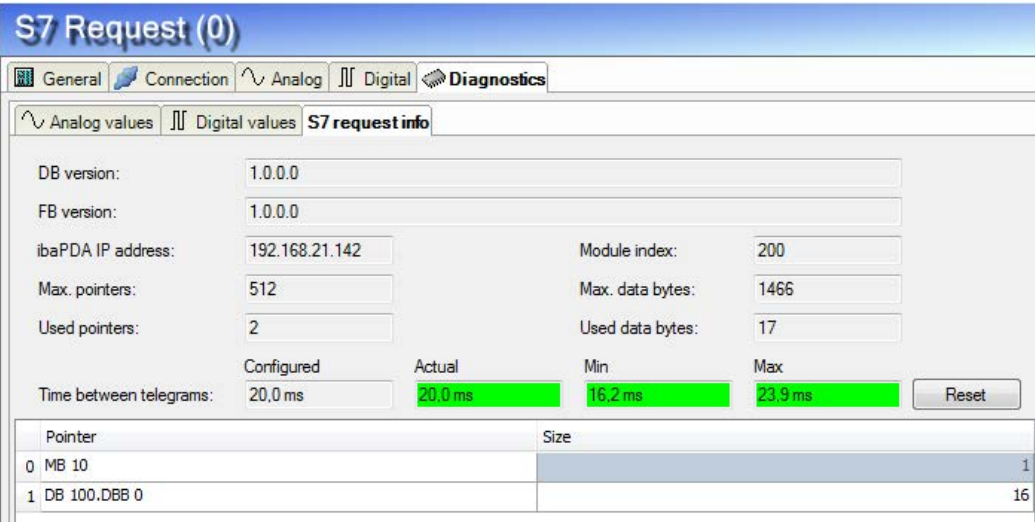
Fig. 21: Module diagnostics analog values



Name	S7 Operand	Value
0 M 10.0	M 10.0	1
1 M 10.1	M 10.1	1
2 M 10.2	M 10.2	1
3 M 10.3	M 10.3	1
4 M 10.4	M 10.4	0
5 M 10.5	M 10.5	1
6 M 10.6	M 10.6	0
7 M 10.7	M 10.7	1
8		
9		

Fig. 22: Module diagnostics digital values

On the *S7 request info* tab the data sent to the S7-CPU, and the data that are reported back as well as the general diagnostics data are displayed.



DB version: 1.0.0.0  
 FB version: 1.0.0.0  
 ibaPDA IP address: 192.168.21.142  
 Module index: 200  
 Max. pointers: 512  
 Max. data bytes: 1466  
 Used pointers: 2  
 Used data bytes: 17  
 Time between telegrams: Configured 20,0 ms, Actual 20,0 ms, Min 16,2 ms, Max 23,9 ms  
 Reset

Pointer	Size
0 MB 10	1
1 DB 100.DBB 0	16

Fig. 23: Module diagnostics S7 request info



**DB version**

Version of the data block used in the CPU

**FB version**

Version of the function block used in the CPU

**ibaPDA IP address**

IP address of the *ibaPDA* computer sent to the S7-CPU.

**Module index**

Module index sent to the S7-CPU (see also ↗ *General module settings*, page 43)

**Max. pointers**

The max. number of used pointers (depends on the size of the data block *ibaREQ\_DB*).

**Used pointers**

Currently used number of pointers.

**Max. data bytes**

Max. size of the user data in the data telegrams to *ibaPDA*

**Used data bytes**

Currently used bytes in the user data of the data telegrams.

**Time between telegrams:**

Configured: Corresponds to the setting "Timebase" on the *General* tab

Actual: Time between the two last received telegrams

Min: shortest time

Max: longest time

<Reset> Resetting the Min and Max values

The background color of the values "Actual", "Min" and "Max" provides additional information:

Color	Meaning
Green	The timespan between two telegrams is shorter than the double of the defined timebase.
Orange	The timespan between two telegrams is higher or equals the double of the defined timebase.

**Pointer table**

Currently required data pointer with address and length

For optimizing the communication performance, signals with consecutive addresses are requested and transferred as a block (pointer).

### 3.3.9 Address books

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

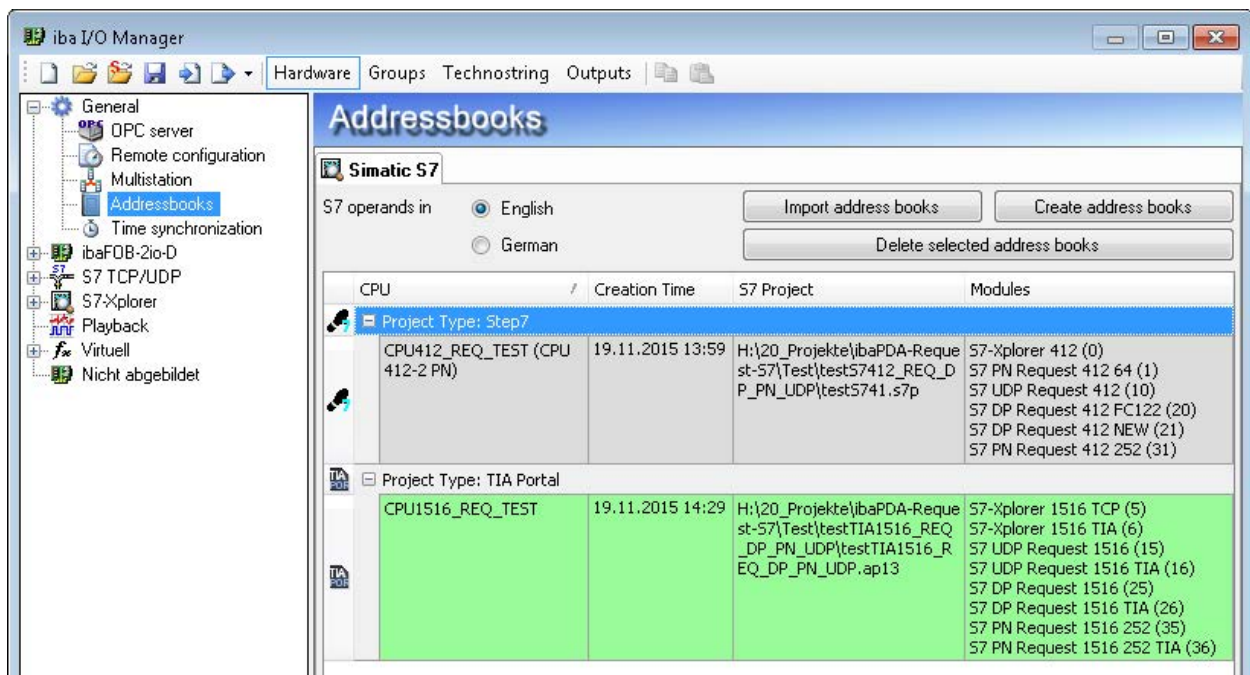


Fig. 24: S7 address book

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

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

#### S7 operands in English / German

Selection of the signal mnemonics

#### Import address books

Import of address books which have already been created

#### Create addressbooks

Creates new address books from STEP 7 projects

#### Delete selected address books

Deletes address books from a table

#### Table

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

#### Creating an address book

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

## Offline from S7 project

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

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

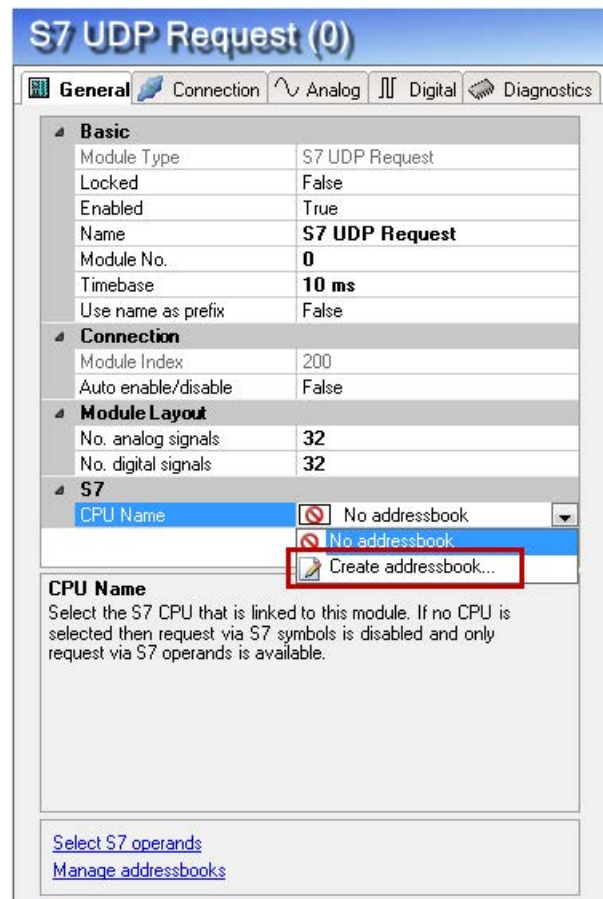
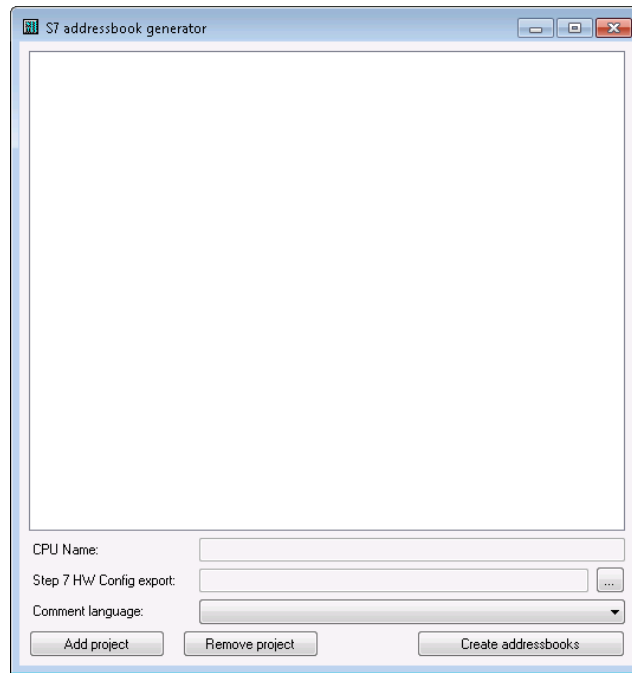


Fig. 25: Create S7 address book from module configuration

How to proceed in the address book generator:

1. Click on the <Add project> button

**CPU Name**

CPU designation

**Step 7 HW Config export**

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

**Comment language**

Selection of the comment language to be imported (only available for SIMATIC TIA Portal projects)

**Add project**

Adds a new project to the list

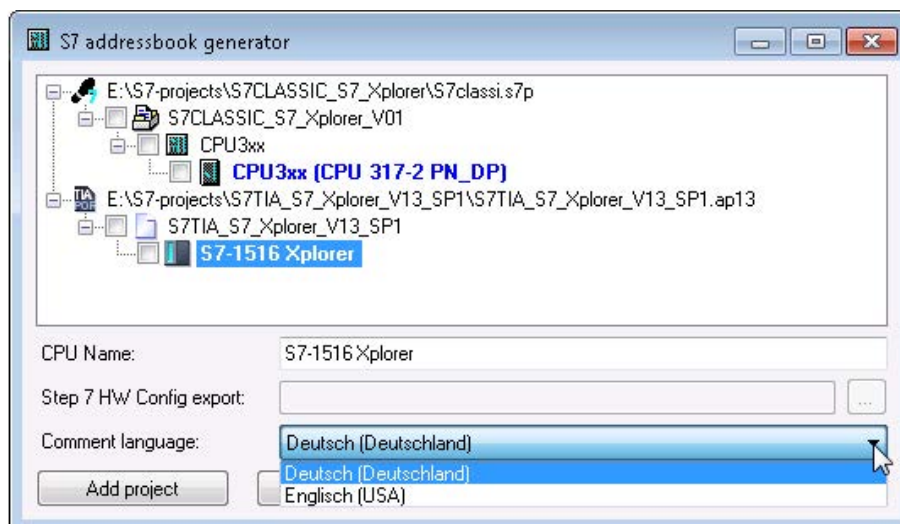
**Remove project**

Removes the marked project from the list

**Create addressbooks**

Creates address books from the selected projects

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



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

#### Note



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

#### Note



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

#### Online from S7-1200 / 1500 CPU

Online address books can be created from S7-1200 or S7-1500 CPUs 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 the button <Load addressbook from S7> to load the address book.

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

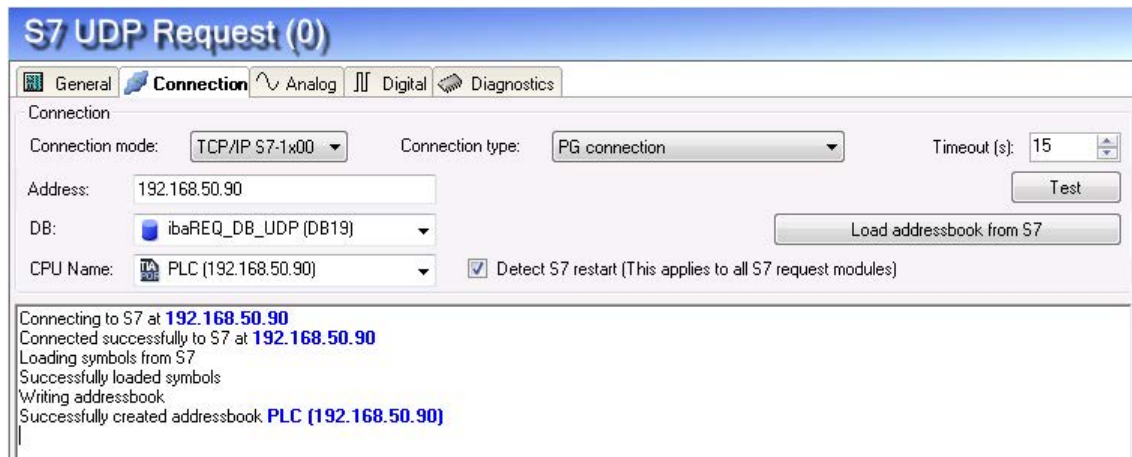


Fig. 26: Load address book online from CPU

Address books read out online also contain operand address information and can therefore also be used in TCP/IP connection mode. Just change the connection mode after creation of the address book.

## 4 Diagnostics

### 4.1 Checking the license

If the "S7 UDP Request" modules are not available in the signal tree, you can either check in *ibaPDA* in the I/O Manager under *General - Settings - License Info* or in the *ibaPDA* service status application if your license is detected properly. The number of licensed connections is shown in brackets.

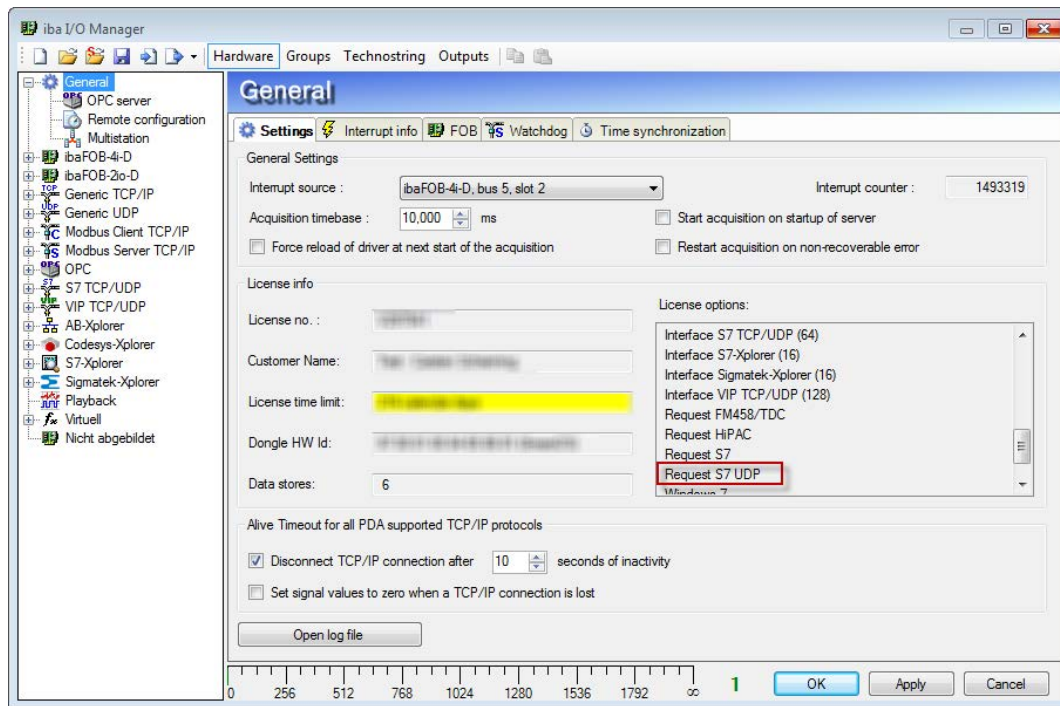


Fig. 27: Checking the license

## 4.2 Log files

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

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

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

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

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

Examples:

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



### 4.3 Connection diagnostics with PING

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

Open a Windows command prompt.



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

With an existing connection you receive several replies.

```
Administrator: C:\Windows\system32\cmd.exe

C:\Users>ping 192.168.21.120

Pinging 192.168.21.120 with 32 bytes of data:
Reply from 192.168.21.120: bytes=32 time<1ms TTL=128
Reply from 192.168.21.120: bytes=32 time<1ms TTL=128
Reply from 192.168.21.120: bytes=32 time=1ms TTL=128
Reply from 192.168.21.120: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.21.120:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\Users>
```

Fig. 28: PING successful

With no existing connection you receive error messages.

```
Administrator: C:\Windows\system32\cmd.exe

C:\Users>ping 192.168.21.121

Pinging 192.168.21.121 with 32 bytes of data:
Reply from 192.168.21.104: Destination host unreachable.
Reply from 192.168.21.104: Destination host unreachable.
Reply from 192.168.21.104: Destination host unreachable.
Reply from 192.168.21.104: Destination host unreachable.

Ping statistics for 192.168.21.121:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

C:\Users>
```

Fig. 29: PING unsuccessful

## 4.4 Connection table

For every Ethernet-based interface, there is a table available in the I/O manager which shows the status of each connection. Each line represents one connection. The following figure shows, as an example, the connection table of the Codesys-Xplorer interface:

Name	Error count	Update time Actual	Response time Actual	Response time Average	Response time Min	Response time Max
0 Codesys V2 - ...	0	1,0 ms	0,0 ms	0,0 ms	0,0 ms	6,0 ms
1 Codesys V3 - ...	0	1,3 ms	0,0 ms	0,4 ms	0,0 ms	16,0 ms
2 ?	?	?	?	?	?	?
3 ?	?	?	?	?	?	?

Fig. 30: Connection table, example Codesys-Xplorer

The connected target systems (controllers) are identified by their name or IP address in the first (left) column.

Depending on the interface type the table shows error counters, read counters and/or data sizes, as well as the cycle times, refresh times and/or update times of the different connections during the data acquisition. Click the <Reset counters> button to reset the error counters and the calculation of the response times.

Additional information is provided by the background color of the table rows:

Color	Meaning
Green	The connection is OK and the data are read.
Yellow	The connection is OK, however the data update is slower than the configured update time.
Red	The connection has failed.
Gray	No connection configured.

Table 7: Meaning of background colors

## 4.5 Module diagnostics

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

See chapter [Module diagnostics](#), page 64.

## 5 Appendix

### 5.1 Iba S7 library

The iba S7 library is available in two versions:

- SIMATIC Manager : STEP 7  $\geq$  V5.5
- SIMATIC TIA Portal STEP 7  $\geq$  V14 SP1

#### 5.1.1 iba S7 library for SIMATIC Manager

The iba S7 library for SIMATIC Manager („ibaS7LibCLASSIC\_Vx\_y") is suitable for the use with SIMATIC Manager V5.5 or higher. It contains the Request blocks described in the manual, which are required for the use of *ibaPDA-Request-S7-UDP*.

You find the iba S7 library as an archived file on the DVD "iba Software & Manuals" under [\04\\_Libraries\\_and\\_Examples\10\\_Libraries\01\\_SIMATIC\\_S7\](#)

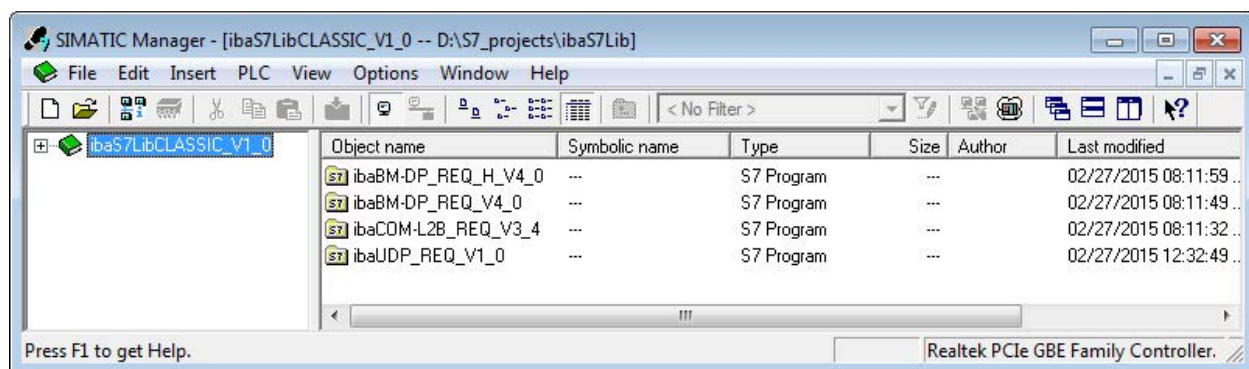


Fig. 31: iba S7 library for SIMATIC Manager

The following components are included:

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

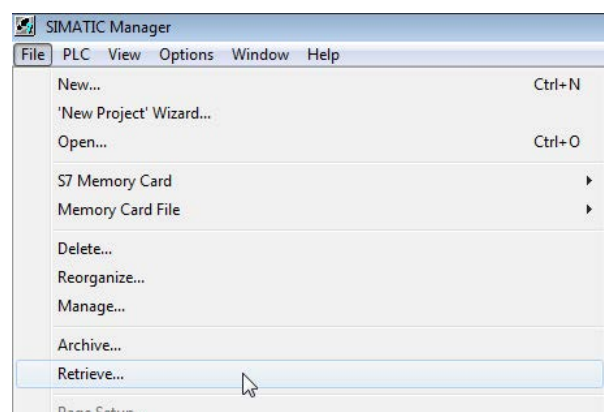
iba connection	Block name	Block no.	Note
ibaCom-L2B	ibaL2B_Init	FC111	formerly FC23 and FC101
	ibaL2B_Req	FC112	formerly FC22 and FC100
	ibaL2B_Req_CP	FC113	formerly FC26 and FC102 only necessary when using a CP342-5 instead of the FC112
	ibaL2B_DB_work	DB22	
	ibaL2B_DB_Struct	UDT22	
	ibaL2B_CP_SNDRCV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

Table 8: ibaS7LibCLASSIC block overview

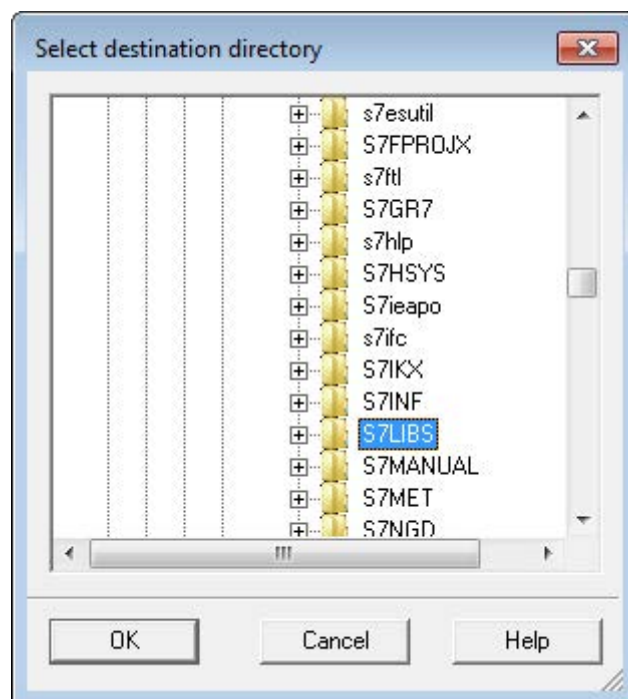
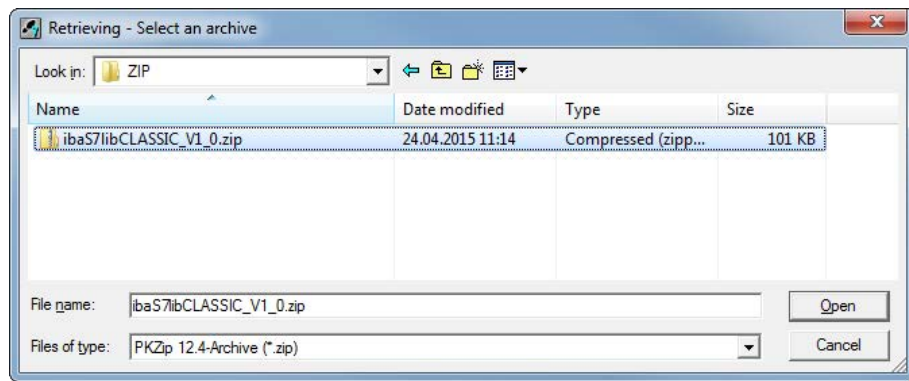
### 5.1.1.1 Integrating the library

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

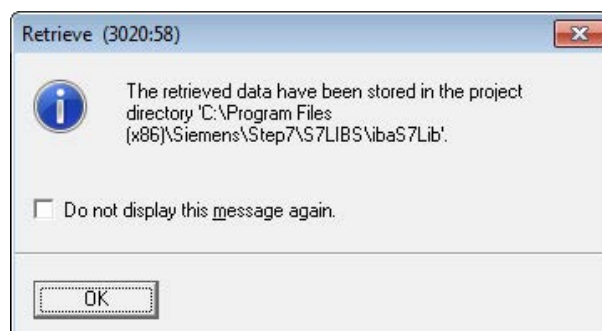
- Select the menu *File – Retrieve...*



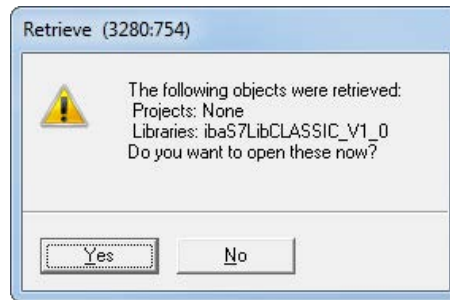
- Choose the archive file of the iba S7 library and select a storage location for the extracted library in a next step:



- Confirm the message for a successful extraction.



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

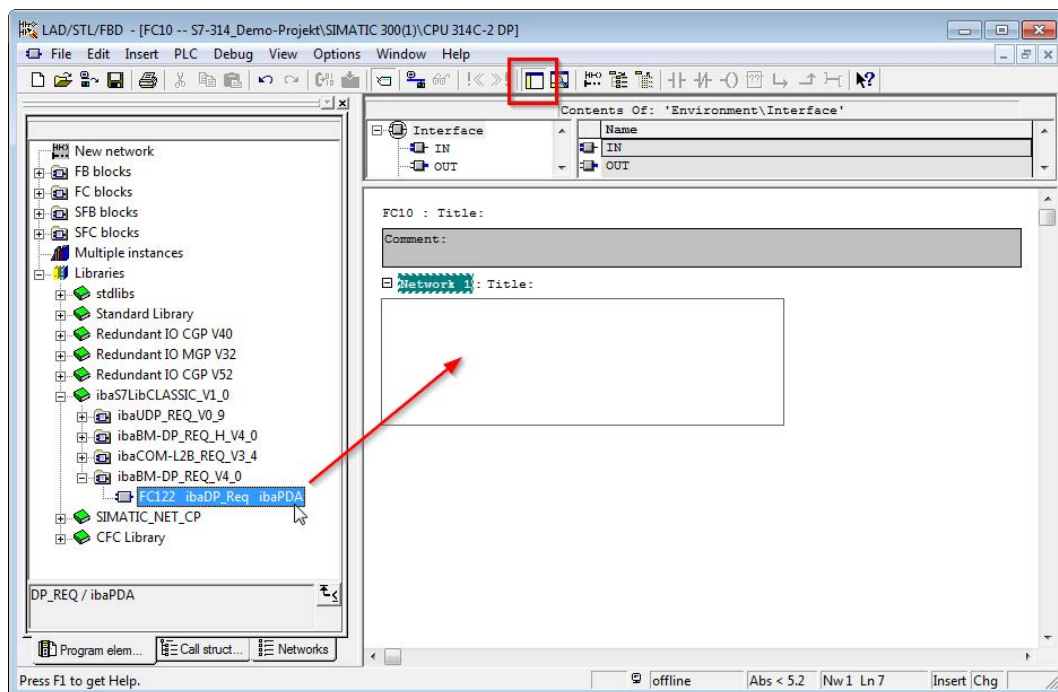


Now, the library is integrated and can be closed again.

### 5.1.1.2 Using the blocks

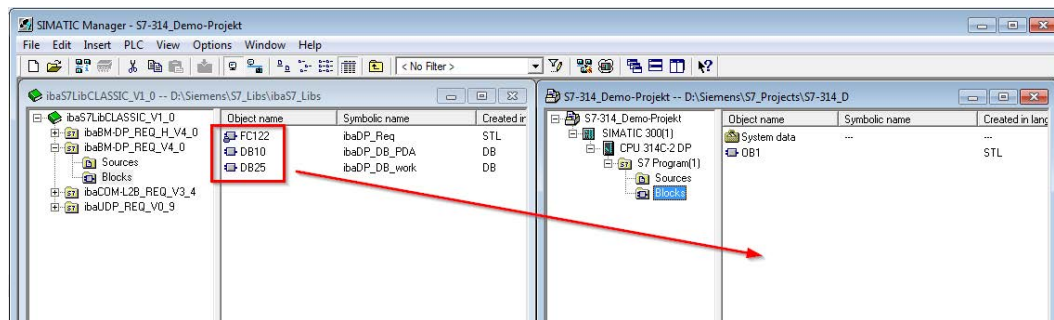
There are two options for using the blocks from the library.

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



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





The blocks may now be used in the destination project.

### 5.1.2 iba S7 library for SIMATIC TIA portal

The iba S7 library for SIMATIC TIA Portal („ibaS7LibTIA\_Vx\_y") is suited for the use with SIMATIC TIA Portal. It contains the Request blocks described in the manual, which are required for the use of ibaPDA-Request-S7-UDP.

You find the iba S7 library as project archive on the DVD "iba Software & Manuals" under  
 \04\_Libraries\_and\_Examples\10\_Libraries\01\_SIMATIC\_S7\

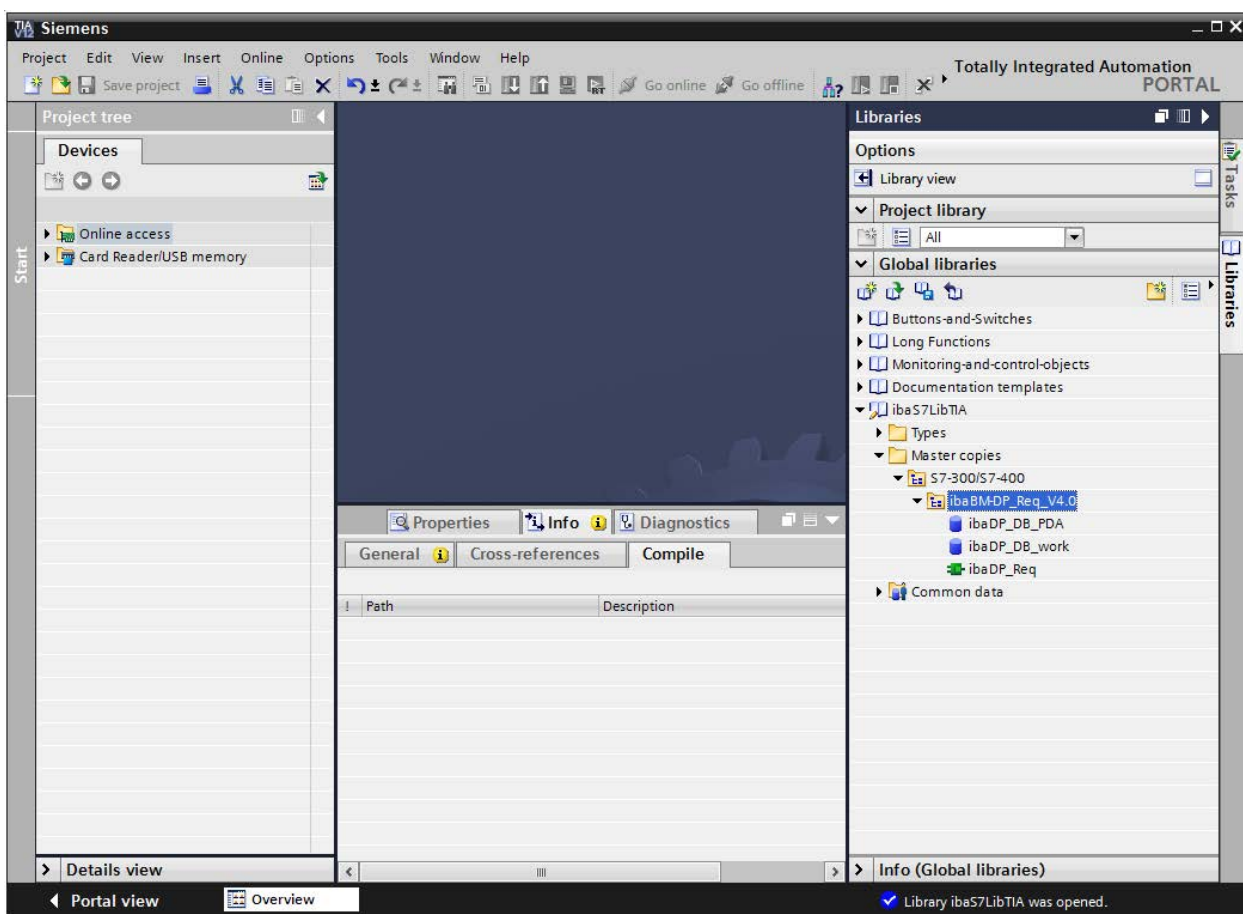


Fig. 32: iba S7 library for SIMATIC TIA portal

The following components are included:

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

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



iba connection	Block name	Block no.	Note
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		

### Note

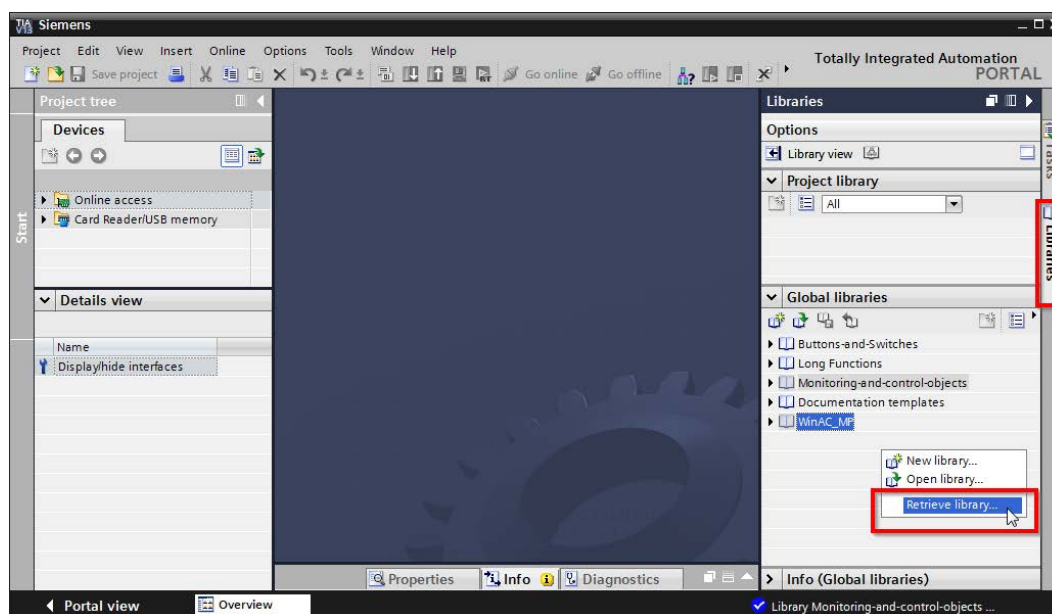


TIA portal libraries are version-dependent. There may be upward compatibility depending on the TIA portal version.

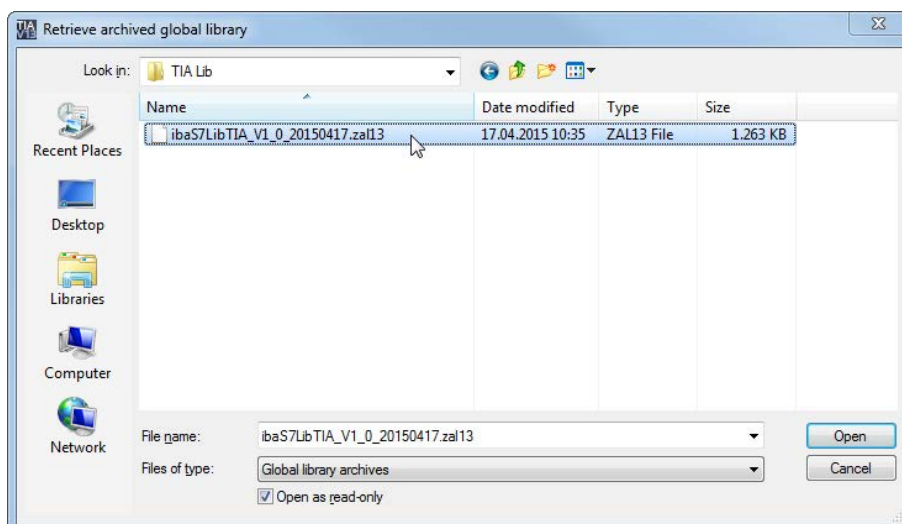
#### 5.1.2.1 Integrating the library

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

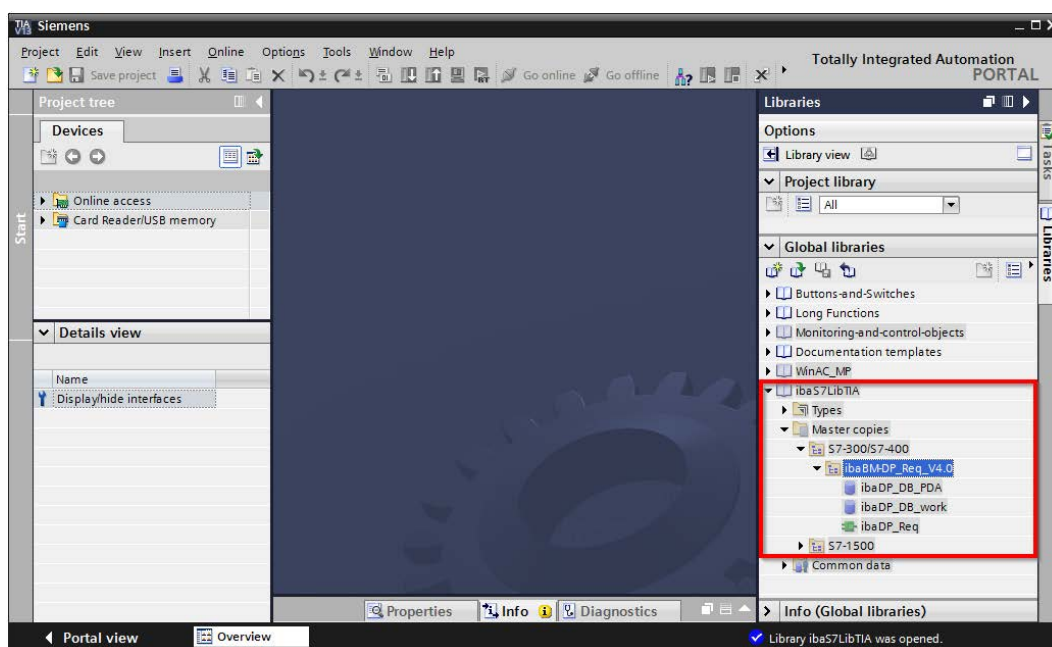
- Select the *Retrieve library...* command on the *Libraries* tab in the context menu (right mouse-button).



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



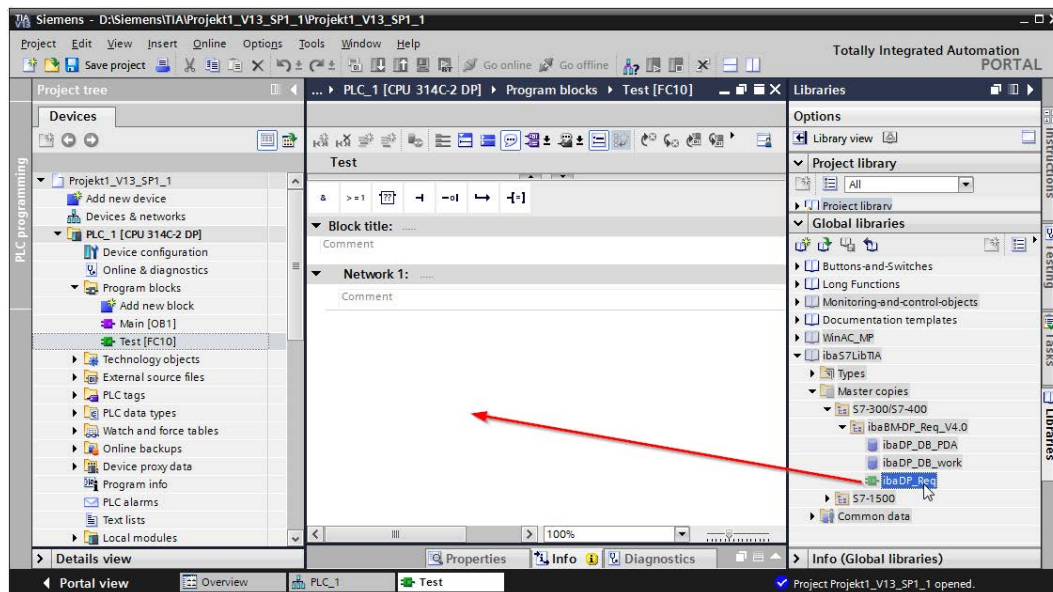
- Now, the library is integrated.



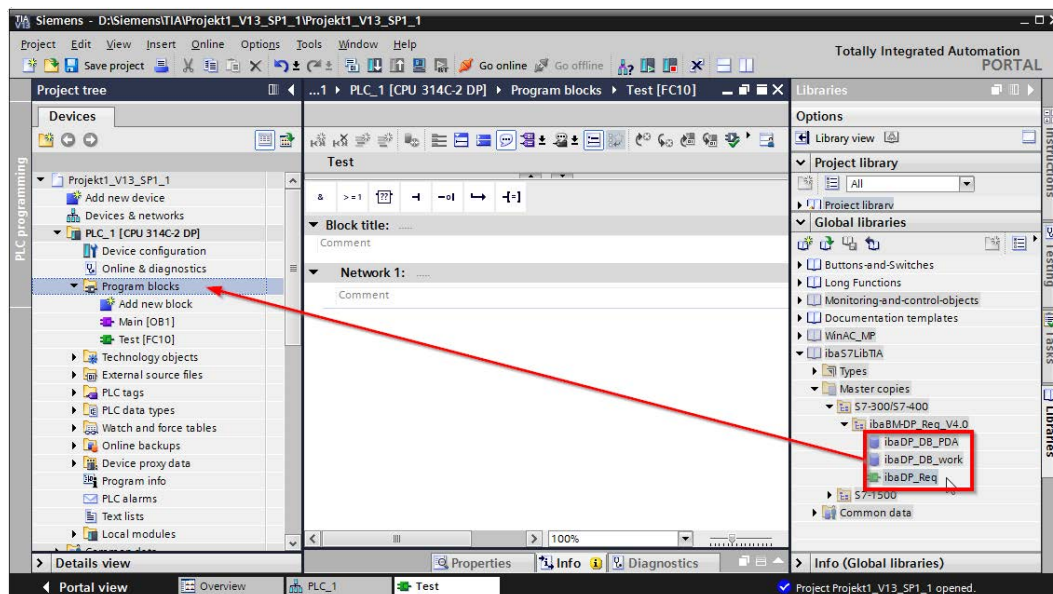
### 5.1.2.2 Using the blocks

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

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

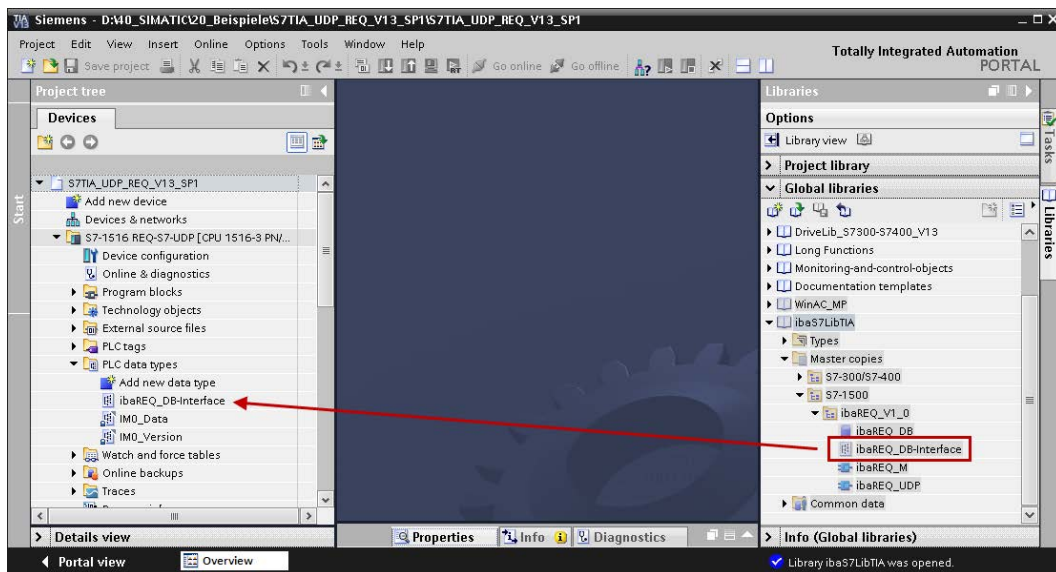


2. Drag or copy the blocks to the program blocks directory in the project navigation.



The blocks can now be called within a project block.

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



## 5.2 Application examples

You find application examples for several different configurations on the DVD "iba Software & Manuals" under:

\\04\_Libraries\_and\_Examples\\50\_ibaPDA-Interface-S7-TCP\_UDP\\Request-S7\\

iba	S7-CPU	S7 project	ibaPDA project
ibaPDA-... ...Interface-S7-TCP/UDP + ...Request-S7-UDP	S7-300 PN	S7CLASSIC_ UDP_REQ_Vxx.zip	ibaPDA_S7CLASSIC_ UDP_REQ_Vxx.zip
	S7-300 + CP343-1 LEAN		
	S7-400 + CP443-1		
	S7-1500	S7TIA_UDP_REQ_ Vx_SPx_Vyy.zip	ibaPDA_S7TIA_UDP_ REQ_Vyy.zip

Table 9: Application examples on the DVD

## 5.3 S7 cycle time measurements

The following tables show which code run times are needed by the Request blocks. The measurement values have been determined in a test environment and only serve as reference points. The values may deviate in other system environments.

SIMATIC S7-CPU	Number of signals	Data amount	ibaREQ_M FB140	ibaREQ_UDPint FB146
CPU412-2 PN 6ES7 412-2EK06-0AB0	1 INT + 0 BOOL (1 Pointer)	2 Byte	128 µs	510 µs
	59 REAL + 64 BOOL (2 Pointer)	244 Byte	132 µs	595 µs
	59 REAL + 64 BOOL (123 Pointer)	244 Byte	132 µs	1100 µs
	122 INT + 0 BOOL (1 Pointer)	244 Byte	132 µs	560 µs
	122 INT + 0 BOOL (122 Pointer)	244 Byte	132 µs	1112 µs
	512 INT + 512 BOOL (2 Pointer)	1088 Byte	132 µs	684 µs
	512 INT + 512 BOOL (1024 Pointer)	1088 Byte	132 µs	5502 µs
	366 REAL + 0 BOOL (1 Pointer)	1464 Byte	132 µs	700 µs
	366 REAL + 0 BOOL (366 Pointer)	1464 Byte	132 µs	2434 µs

SIMATIC S7-CPU	Number of signals	Data amount	ibaREQ_M FB1400	ibaREQ_UDP FB1405
CPU1516-3 PN/DP 6ES7 516-3AN00-0AB0	1 INT + 0 BOOL (1 Pointer)	2 Byte	195 µs	402 µs
	59 REAL + 64 BOOL (2 Pointer)	244 Byte	189 µs	421 µs
	59 REAL + 64 BOOL (123 Pointer)	244 Byte	195 µs	792 µs
	122 INT + 0 BOOL (1 Pointer)	244 Byte	189 µs	413 µs
	122 INT + 0 BOOL (122 Pointer)	244 Byte	195 µs	795 µs
	512 INT + 512 BOOL (2 Pointer)	1088 Byte	189 µs	431 µs
	512 INT + 512 BOOL (1024 Pointer)	1088 Byte	192 µs	2028 µs
	366 REAL + 0 BOOL (1 Pointer)	1464 Byte	189 µs	431 µs
	366 REAL + 0 BOOL (366 Pointer)	1464 Byte	196 µs	1586 µs

## 5.4 Adaptation to the renumbered system functions

This procedure is only necessary when using the SIMATIC Manager (STEP 7 ≤ V5), if a different block number was assigned to one of the following used embedded block numbers:

Symbolic name	Standard numbering	Origin
ibaREQ_UDPact	FB145	ibaAG, ibaS7Lib
TCON	FB65	Siemens, Standard Library
TDISCON	FB66	Siemens, Standard Library
TUSEND	FB67	Siemens, Standard Library
AG_SEND	FC5	Siemens, SIMATIC_NET_CP
AG_LSEND	FC50	Siemens, SIMATIC_NET_CP
ibaUDT_UDPact	UDT145	ibaAG, ibaS7Lib

Table 10: Subordinate blocks

- Copy the following block sources from the iba S7 library to the source folder of your STEP 7 project.

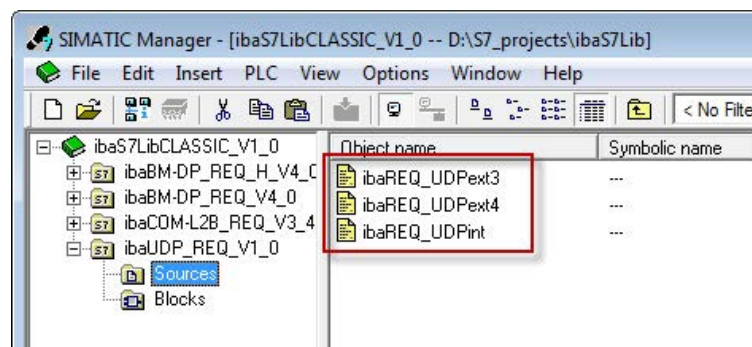


Fig. 33: Function block sources in the ibaS7 function block library.

- Make a new translation for all sources of the function blocks you use.

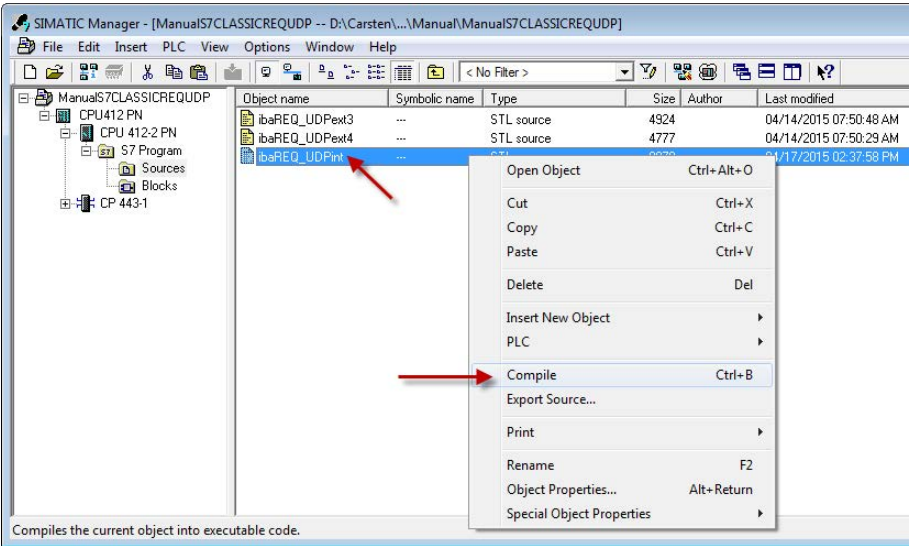


Fig. 34: Translating the function block sources

**Note**



It is absolutely essential that the symbolic designation of the subordinate blocks is not modified (see table above).



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

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

There are 2 possible remedies:

### Changing the interface with remaining access point name

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

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

### Adding a special access point for *ibaPDA*

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

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

The setting for the SIMATIC Manager will also be changed.

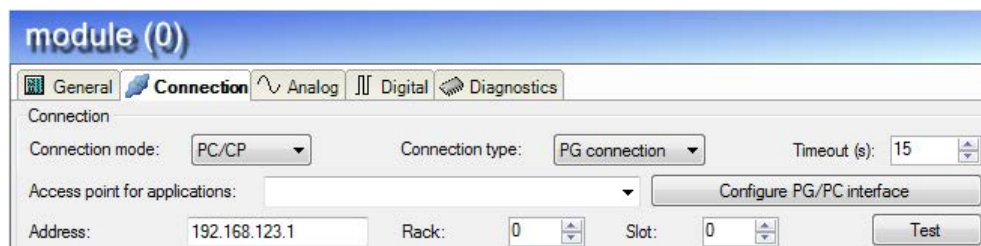
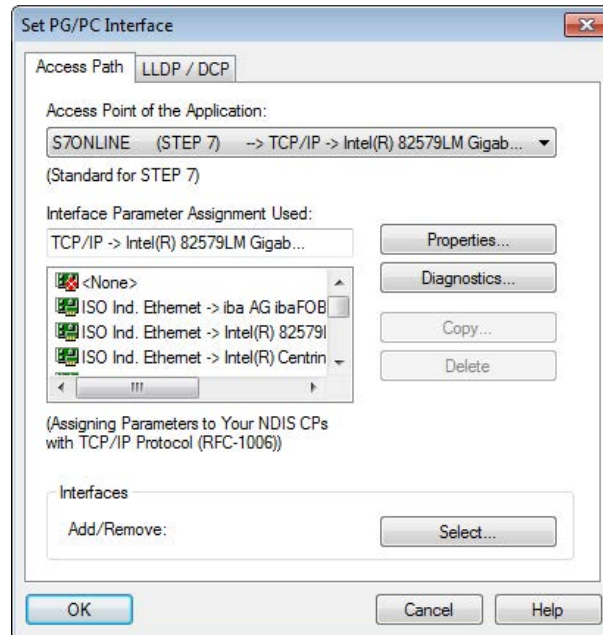


Fig. 35: Configure PG/PC interface

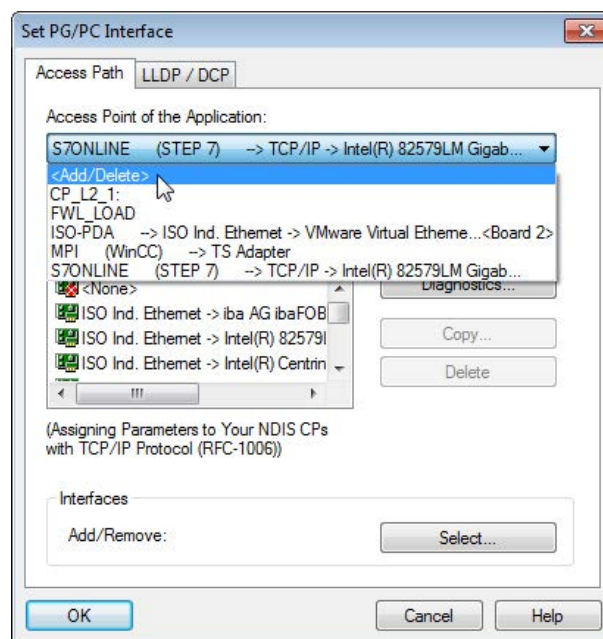


## Procedure

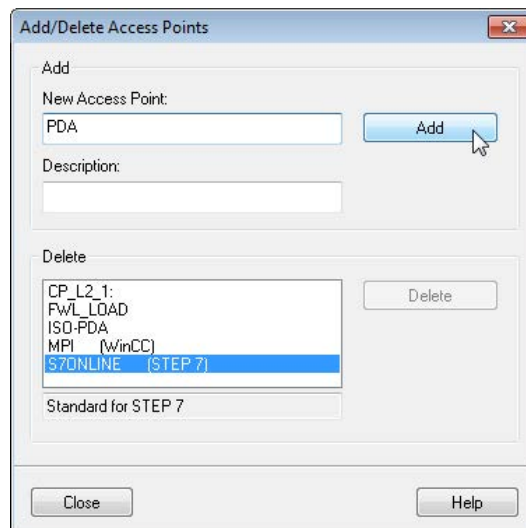
1. Open the dialog box with the <Configure PG/PC interface> button.



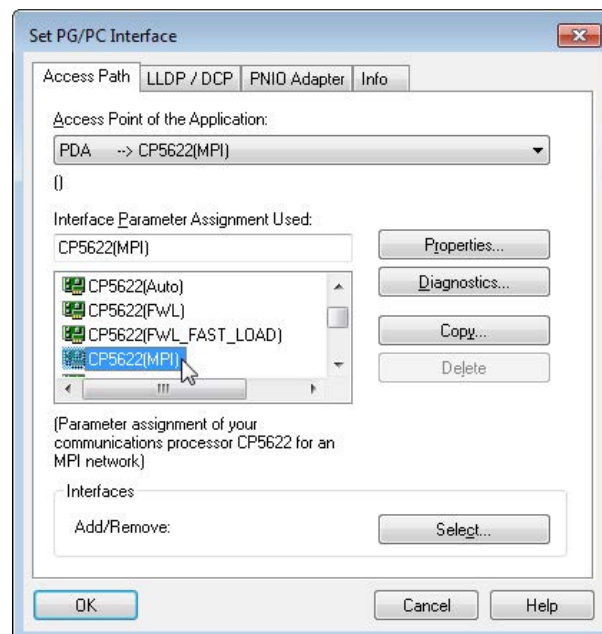
2. Select the row <Add/Delete> under "Access Point of the Application".



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



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



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

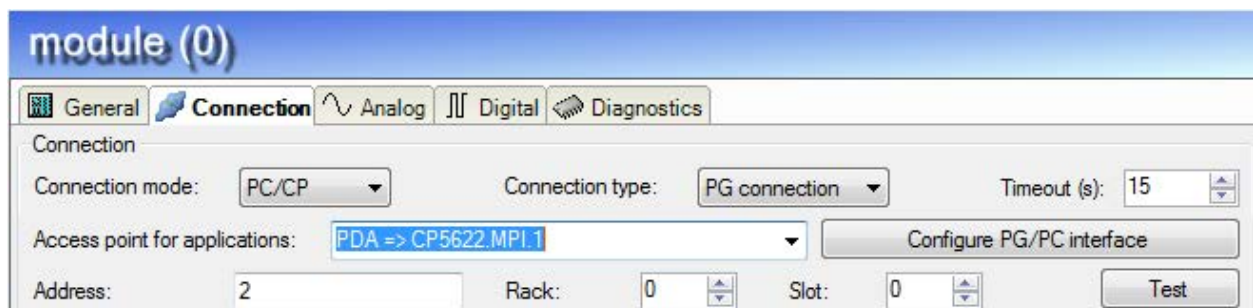


Fig. 36: Set access point

**Notes on the different access points**

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

Basically, there are 3 types of access points:

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

**TCP/IP**

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

**ISO**

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

**Bus system (PROFIBUS or MPI)**

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

## 5.6 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).

### 5.6.1 Routing from Ethernet to Ethernet

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

The following constellation will make this clear:

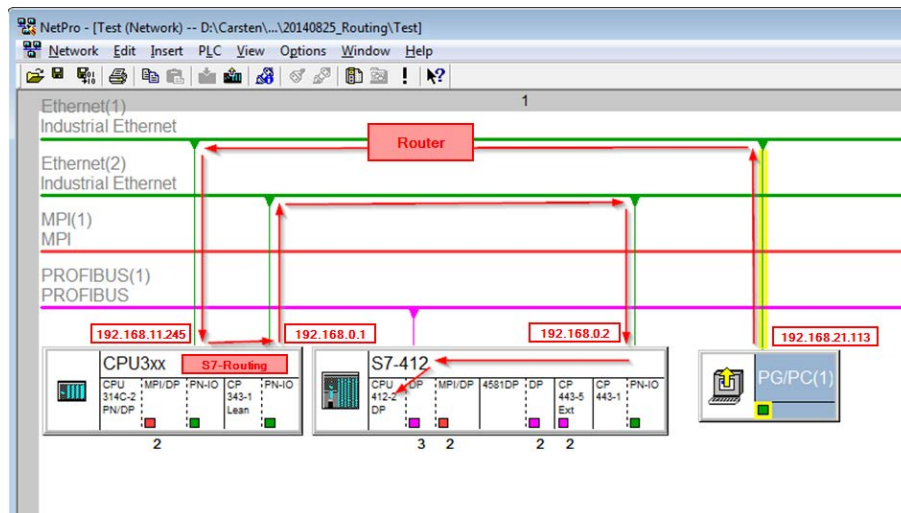


Fig. 37: S7 Routing, example system topology

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

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

### 5.6.1.1 Configuration of STEP 7/ NetPro

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

Inserting a PG/PC station:

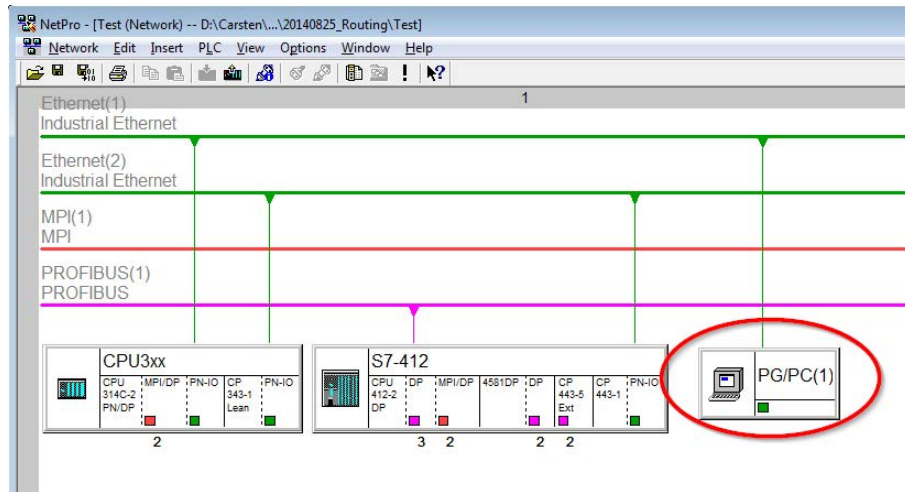


Fig. 38: NetPro configuration

Assigning an interface (network interface card):

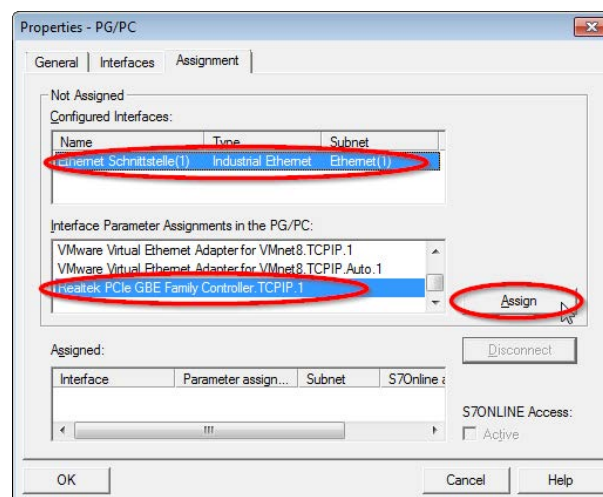


Fig. 39: PG/PC interface assigned

Result:

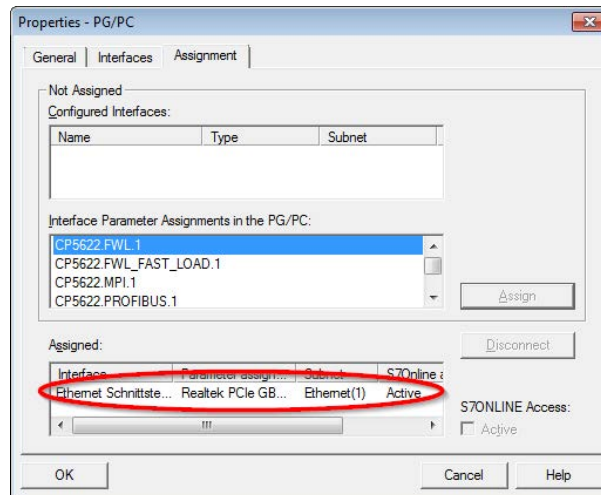


Fig. 40: Interface assigned

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

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

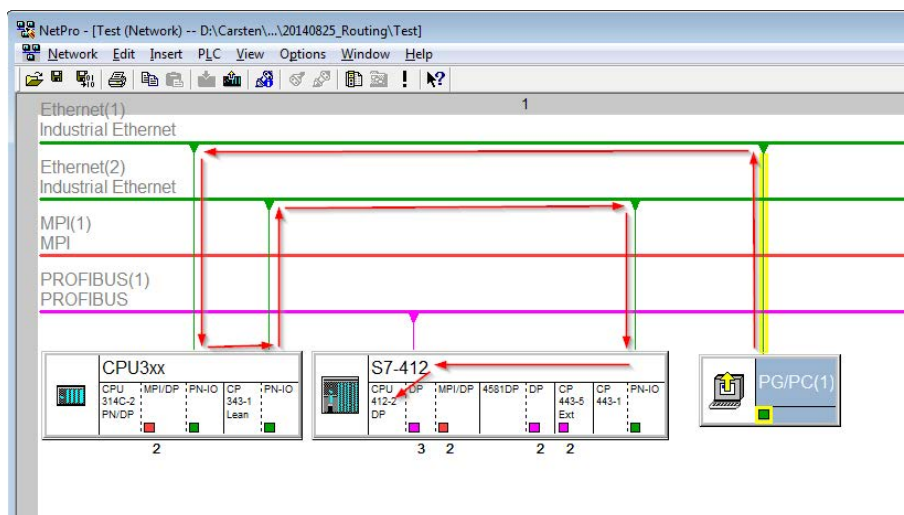


Fig. 41: Communication channel

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

### 5.6.1.2 ibaPDA configuration

The following entries have to be made:

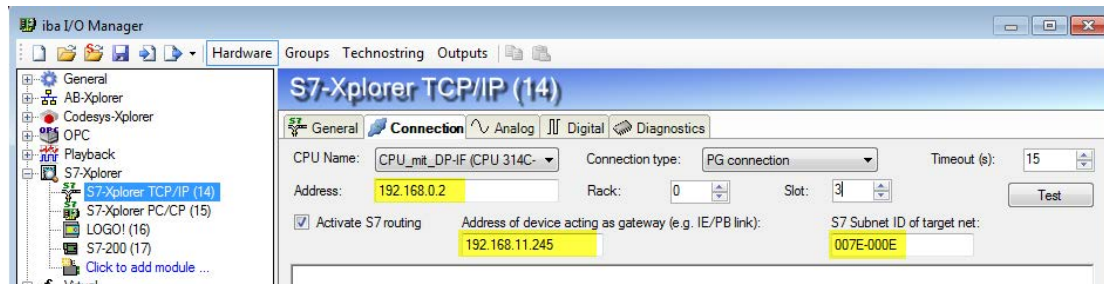


Fig. 42: Enable S7 routing

#### Activate S7 routing

Enable to use S7 routing

#### Address

Address of the target control (here CPU412)

#### Address of device acting as gateway

Enter address of the gateway (here CPU314C)

#### S7 subnet ID of target net

Enter subnet ID from STEP 7 NetPro

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

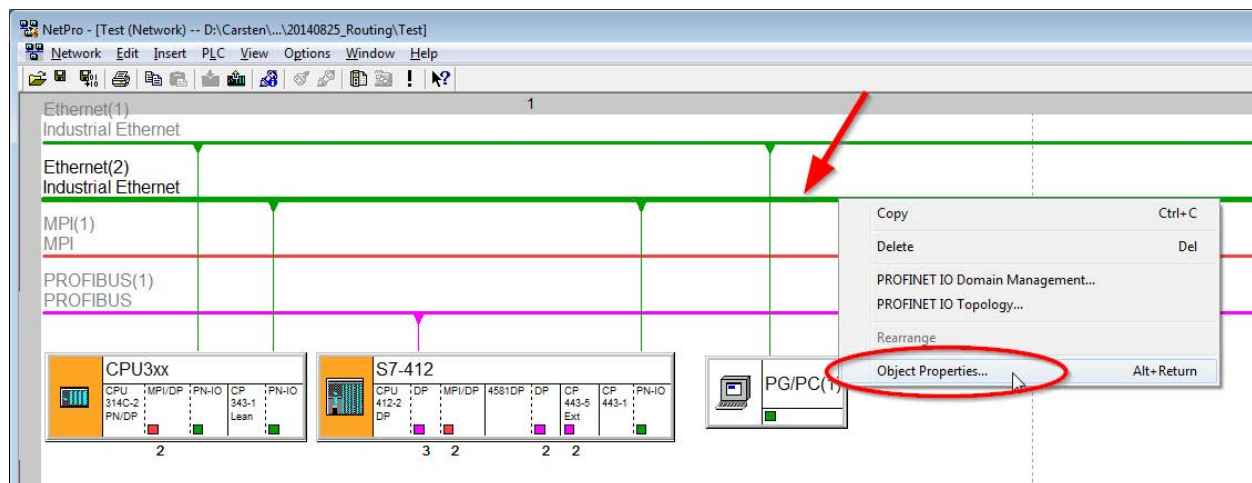


Fig. 43: Determine S7 subnet ID



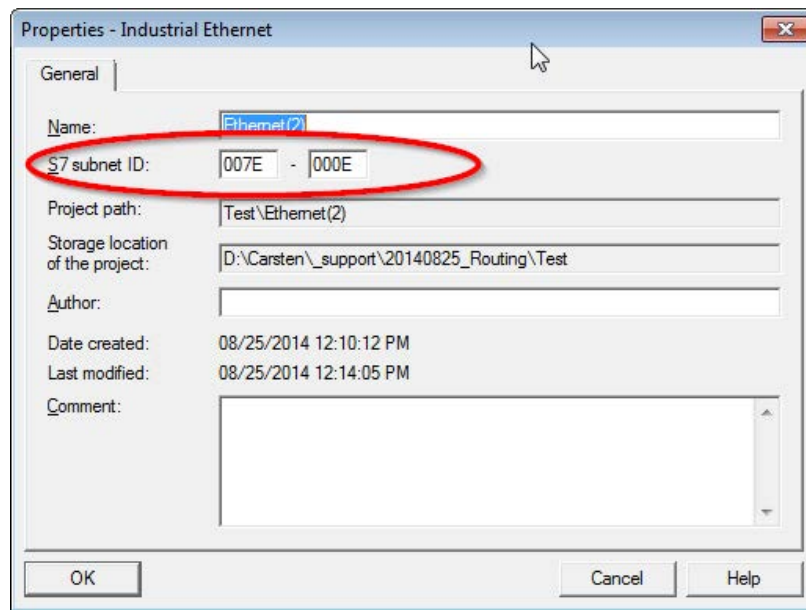


Fig. 44: S7 subnet ID

### 5.6.2 Routing from Ethernet to PROFIBUS

We want to implement the following way of access:

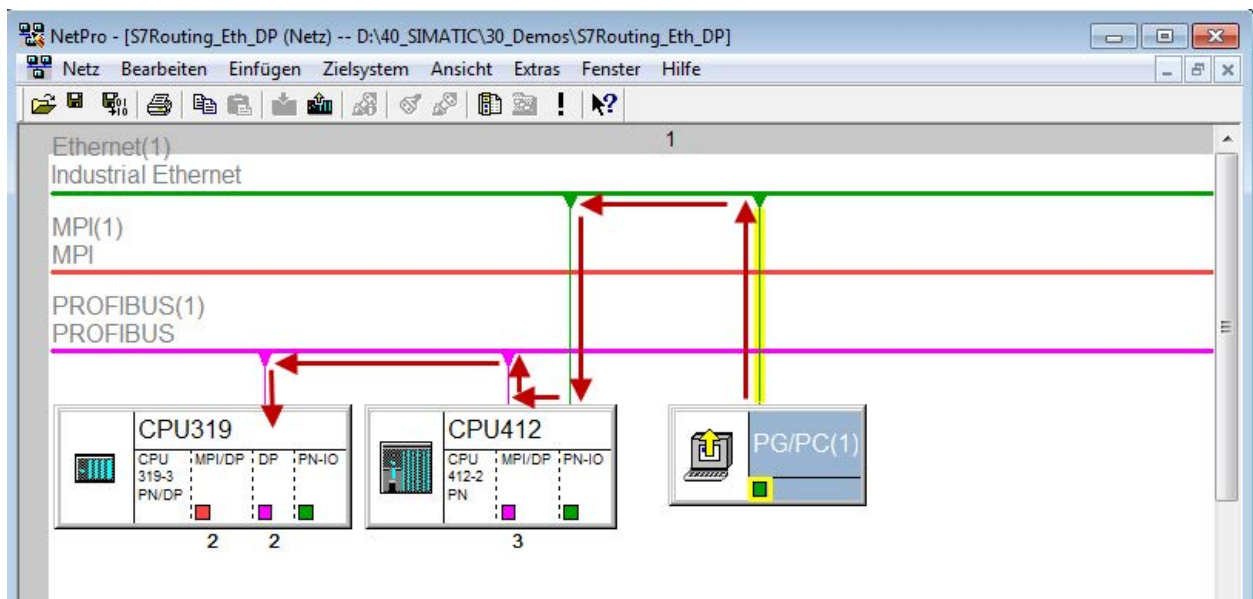


Fig. 45: S7 Routing, example system topology Ethernet PROFIBUS

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



### 5.6.2.1 Configuration STEP 7/ NetPro

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

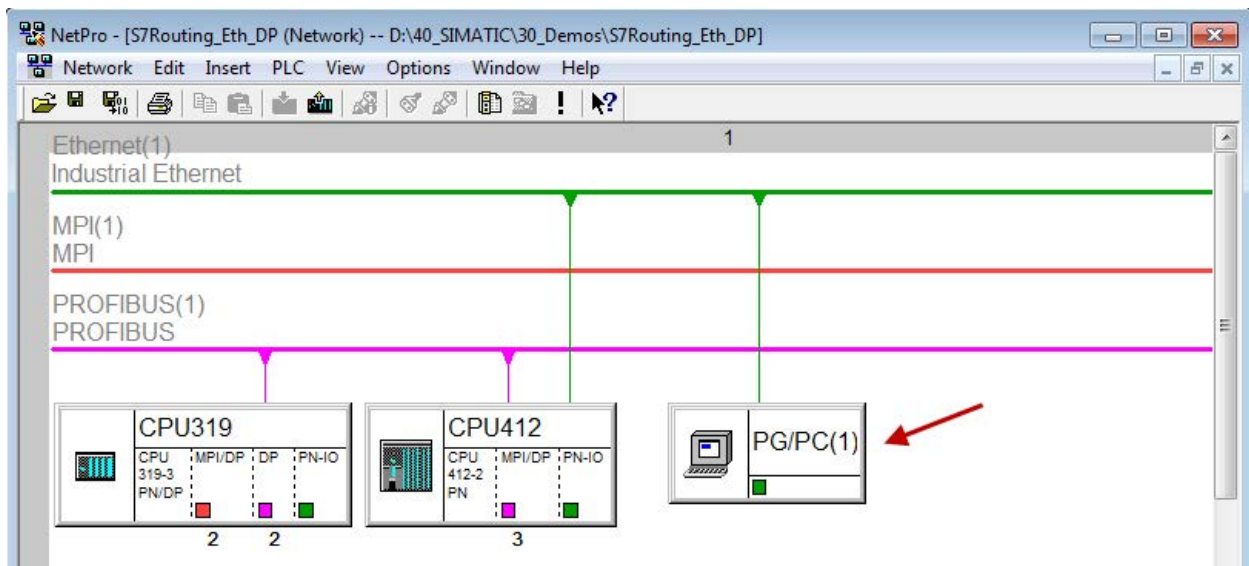


Fig. 46: Configuration NetPro

Assigning an interface (network card):

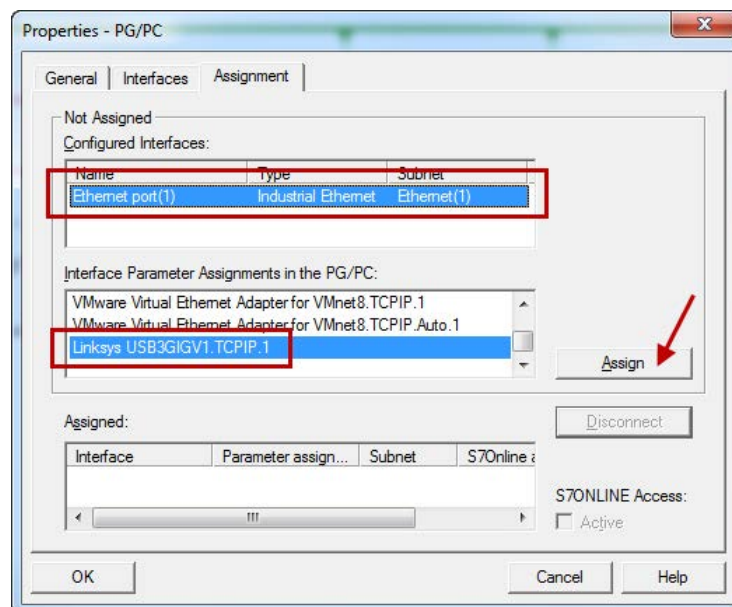


Fig. 47: Assign PG/PC interface

Result:

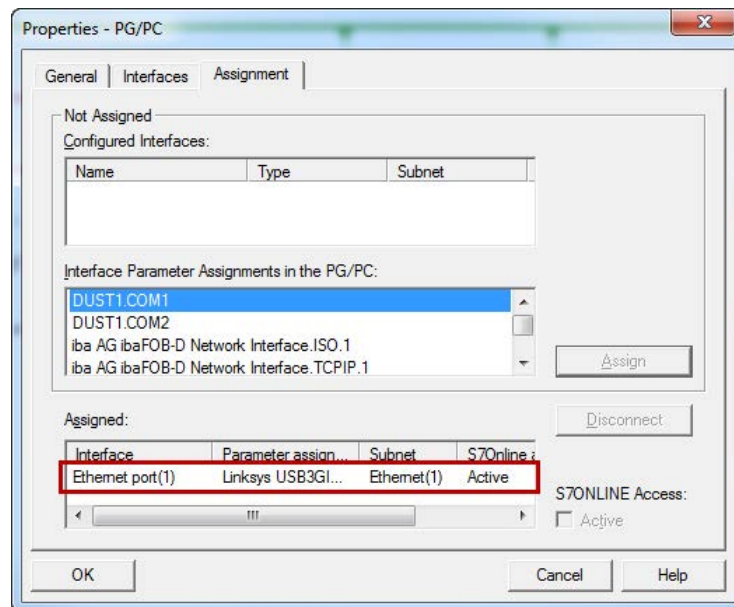


Fig. 48: Interface has been assigned

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

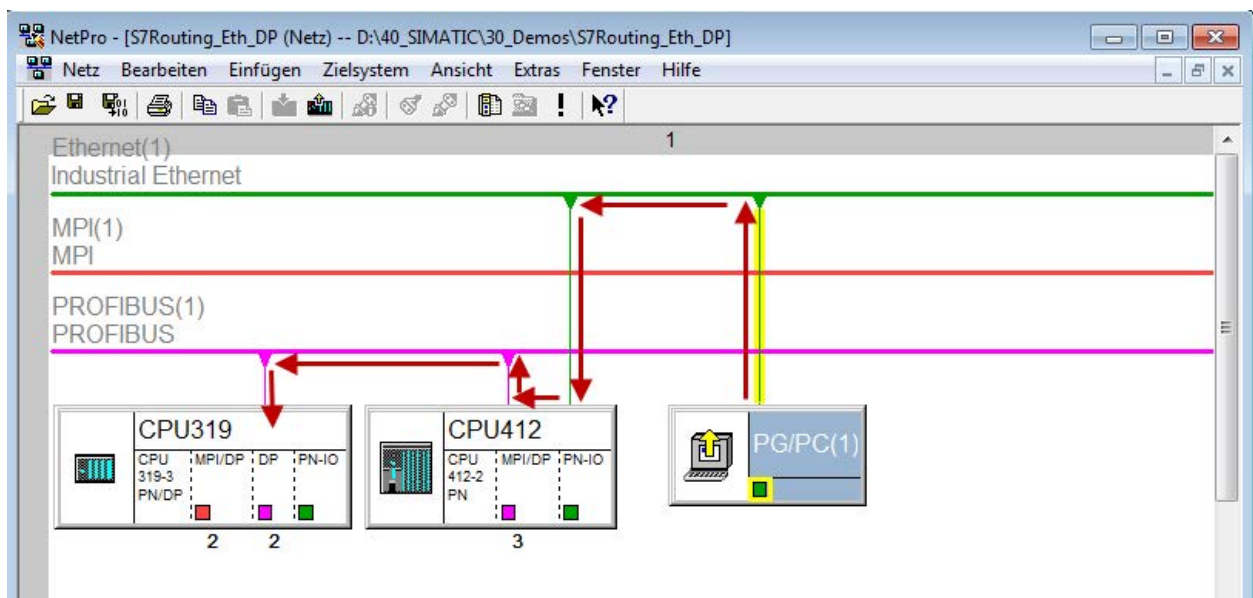


Fig. 49: Communication path

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

### 5.6.2.2 ibaPDA configuration

The following entries have to be made:

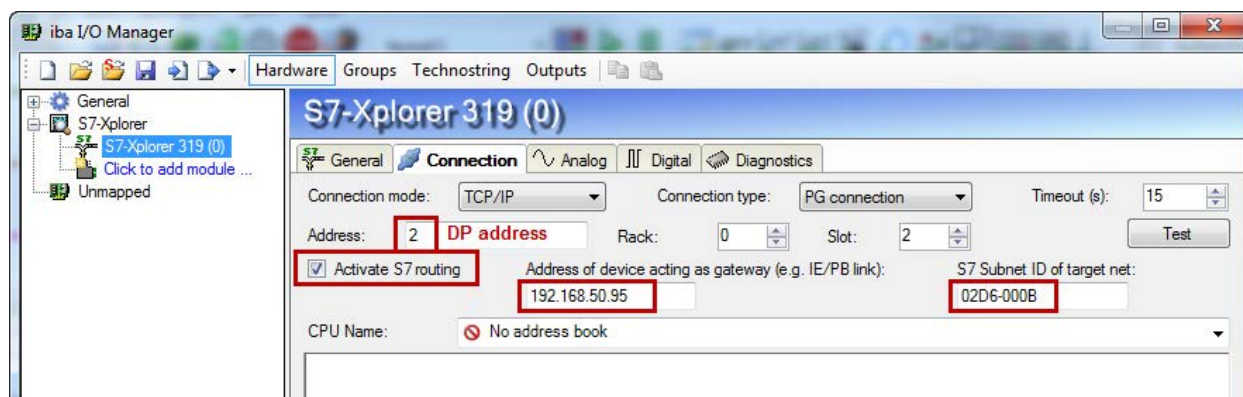


Fig. 50: Enable S7 routing

#### Activate S7 routing

Enable to use S7 routing

#### Address

Address of the target control (here CPU319)

#### Address of device acting as gateway

Enter address of the gateway (here CPU412)

#### S7 subnet ID of target net

Enter subnet ID from STEP 7 NetPro

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

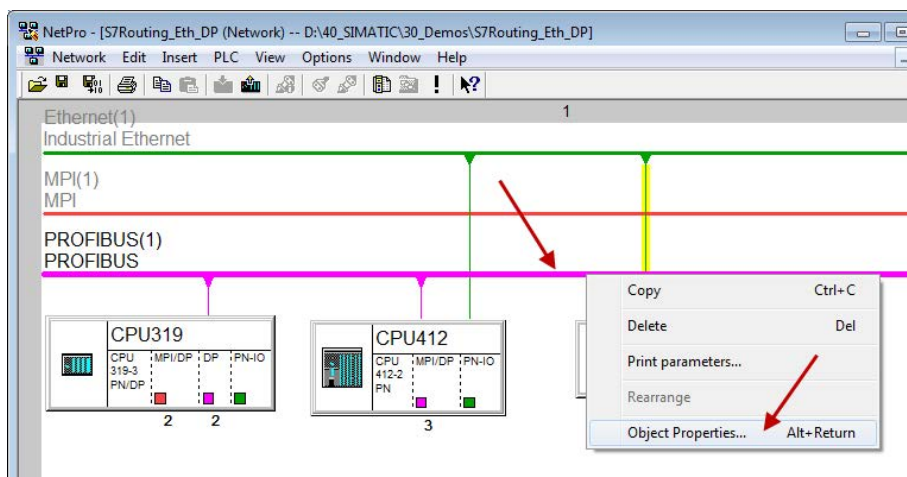


Fig. 51: Determine S7 subnet ID

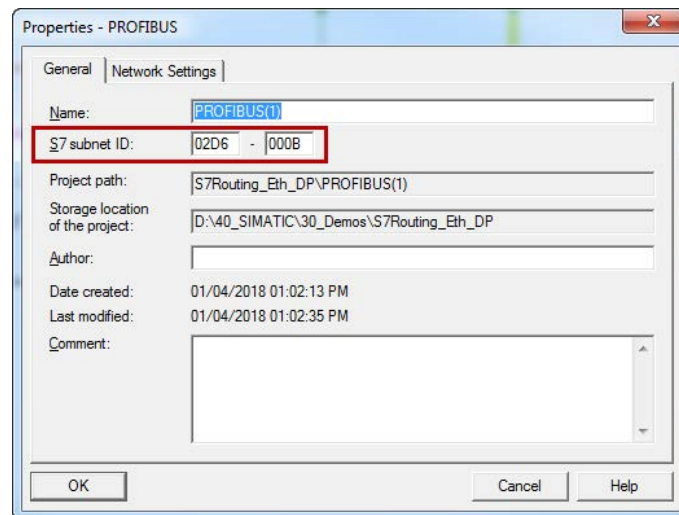


Fig. 52: S7 subnet ID

## Reference



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

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

<http://support.automation.siemens.com/ww/view/en/584459>

Which requirements must be fulfilled and what do I have to observe if I want to execute routing? <https://support.industry.siemens.com/cs/ww/de/view/2383206>

How do you enable cross-project S7 Routing in the TIA Portal and in STEP 7 V5.x?

<https://support.industry.siemens.com/cs/ww/en/view/109474569>

## 5.7 Error codes of Request blocks

The Request blocks deliver the following possible error codes.

### FB140/141/...

Value ERROR_STATUS	Description
1	datablock ibaREQ_DB is write protected
2	datablock ibaREQ_DB invalid (DB =0 or > limit of cpu)
3	datablock ibaREQ_DB does not exist
4	datablock ibaREQ_DB undefined error
5	datablock ibaREQ_DB too short
6	datablock ibaREQ_DB too short for ibaREQ_UDP
9	internal error (RD_SINFO)
10	no access to datablock ibaREQ_DB (read)
11	no access to datablock ibaREQ_DB (write)
20	initialization not finished

Value ERROR_STATUS	Description
21	insufficient memory for SZL
22	wrong SZL_ID
23	wrong or invalid index of SZL
24	error while reading I&M data from cpu
25	error while reading plc data
31	initialization canceled with error
32	initialization not completed
41	too many pointers (ibaREQ_DB too small)
42	too many pointers in one command (>128)
44	invalid command id
45	operand invalid (not defined)
46	operand invalid (datatype)
47	operand invalid (memory area)
200	no connection to PN device / DP slave
300	version of ibaREQ_UDPack does not match with ibaREQ_M (ID)
301	version of ibaREQ_UDPack does not match with ibaREQ_M (FB)
302	version of ibaREQ_UDPack does not match with ibaREQ_M (DB)
303	type of transmit agent does not match with configured request type in ibaPDA
305	PROFIBUS DP slave hardware configuration is invalid
306	configured peripheral address is invalid
310	no access to datablock ibaREQ_DB (read)
311	no access to datablock ibaREQ_DB (write)
315	error while masking of synchronous faults
316	error while demasking of synchronous faults
320	operand invalid (datatype)
321	operand invalid (pointer)
401	ADR_SLOT / ADR_SLOT_0 invalid hw-id
402	ADR_SLOT / ADR_SLOT_0 invalid hw-id, no IO-Device or DP-Slave
403	ADR_SLOT / ADR_SLOT_0 invalid hw-id, is no PROFIBUS or PROFINET
406	ADR_SLOT / ADR_SLOT_0 invalid configuration slot (0)
407	ADR_SLOT / ADR_SLOT_0 invalid configuration slot (0)
409	ADR_SLOT_1 invalid configuration slot 1
410	no connection to PN device / DP slave or error
411	ADR_SLOT_1 invalid hw-id
412	ADR_SLOT_1 invalid hw-id, no IO-Device or DP-Slave
413	ADR_SLOT_1 invalid hw-id, is no PROFIBUS
416	ADR_SLOT_1 invalid configuration slot 1
0x8yyy	errorcode of inner TUSEND / AG_SEND / AG_LSEND

Table 11: Error codes of Request blocks FB140/141/...

**FC122 (PROFIBUS)**

Value ERROR_STATUS	Description
1	DB_PDA is write protected
2	DB_PDA = 0 or > limit of cpu
3	DP_PDA does not exist
5	DB_PDA too short
11	DB_INTERN is write protected
12	DB_INTERN = 0 or > limit of CPU
13	DB_INTERN does not exist
15	DB_INTERN too short
16	error while reading identification data of CPU
19	initialization not completed
21	insufficient memory for system status list
22	wrong or unknown system status list
23	wrong or invalid index of system status list
30	invalid OUTPUT_ADR_SLAVE
31	OUTPUT_ADR_SLAVE no PROFIBUS DP slave
100	bit number not 0
101	bit number not 0-7
103	operand invalid (memory area)
104	operand invalid (datatype)
105	operand invalid (datablock 0)
106	datablock number > limit of cpu
107	datablock does not exist
109	datablock too short
110	address does not exist
111	initialization canceled with error
112	initialization not completed
150	request fragmentation not supported
151	wrong number of requested values
152	only <64 digital signals are supported
153	only <64 analog signals are supported
200	no connection to DP slave

Table 12: Error codes of Request block FC122

**FC123**

Value ERROR_STATUS	Description
1	DB_PDA is write protected
2	DB_PDA = 0 or > limit of cpu
3	DP_PDA does not exist
5	DB_PDA too short
11	DB_INTERN is write protected
12	DB_INTERN = 0 or > limit of CPU
13	DB_INTERN does not exist
15	DB_INTERN too short
16	error while reading identification data of CPU
19	initialization not completed
21	insufficient memory for system status list
22	wrong or unknown system status list
23	wrong or invalid index of system status list
30	invalid OUTPUT_ADR_SLAVE
31	OUTPUT_ADR_SLAVE no PROFIBUS DP slave
32	RM: the parameterized "OUTPUT_ADR_SLAVE_BUS_0" ist wrong.
33	RM: the parameterized "OUTPUT_ADR_SLAVE_BUS_0" is not assigned to a PROFIBUS DP slave
34	RM: the parameterized "OUTPUT_ADR_SLAVE_BUS_1" ist wrong
35	RM: the parameterized "OUTPUT_ADR_SLAVE_BUS_1" is not assigned to a PROFIBUS DP slave
36	RM: SLAVE BUS0 and SLAVE BUS1 do not have the same DP address
100	bit number not 0
101	bit number not 0-7
103	operand invalid (memory area)
104	operand invalid (datatype)
105	operand invalid (datablock 0)
106	datablock number > limit of cpu
107	datablock does not exist
109	datablock too short
110	address does not exist
111	initialization canceled with error
112	initialization not completed
150	request fragmentation not supported
151	wrong number of requested values
152	only <64 digital signals are supported
153	only <64 analog signals are supported



Value ERROR_STATUS	Description
200	no connection to DP slave
201	RM: slave bus 0 has failed
202	RM: slave bus 1 has failed
203	RM: slaves bus 0 + 1 have failed
210	output modules of the slaves bus 0 and 1 are configured differently

Table 13: Error codes of Request block FC123

### 5.7.1 S7-1500

#### S7-1500

For the ibaREQ\_UDP Request block, the following error codes may appear:

ERROR_TSEND (W#16#...)	
80A1	Connection or port already being used by user. Communication error: - The specified connection has not yet been established. - The specified connection is being terminated. - Transfer via this connection is not possible. - The interface is being re-initialized.
80A3	The nested "T_DIAG" instruction has reported that the connection has closed.
80A4	IP address of the remote endpoint of the connection is invalid or it matches the IP address of the local partner.
80A7	Communication error: You called the instruction with COM_RST = 1 before the send job was complete.
80AA	A connection is currently being established with the same connection ID by another block. Repeat the job with a new rising edge at the REQ parameter.
80B6	Parameter assignment error in the connection_type parameter of the data block for connection description.
80B7	Error in one of the following parameters of the data block for connection description: block_length, local_tsap_id_len, rem_subnet_id_len, rem_staddr_len, rem_tsap_id_len, next_staddr_len.
8085	The LEN parameter is larger than the highest permitted value.
8086	The ID parameter within the CONNECT parameter is outside the permitted range.
8087	Maximum number of connections reached; no additional connection possible.



ERROR_TSEND (W#16#...)	
8088	The value at the LEN parameter does not correspond to the receive area set at the DATA parameter.
8091	Maximum nesting depth exceeded.
809A	The CONNECT parameter points to a field that does not correspond to the length of the connection description.
809B	InterfaceID is invalid. It is either zero or it does not point to a local CPU interface or a CP.
80C3	All connection resources are in use. A block with this ID is already being processed in a different priority group.
80C4	Temporary communication error: - The connection cannot be established at this time. - The interface is receiving new parameters or the connection is being established. - The configured connection is currently being removed by a "TDIS-CON" instruction. - The connection used is being terminated by a call with COM_RST = 1.
80C6	Remote network error. Remote partner cannot be reached.

#### Other documentation



For more information, please see the SIEMENS documentation about the TSEND\_C block.

## 5.7.2 S7-300/400

### S7-300/400

For the ibaREQ\_UDP Request block, the following error codes may appear:

ERROR_TCON (W#16#...)	
8086	The ID parameter is outside the permitted range.
8087	Maximal number of connections reached; no additional connection possible
8089	The parameter CONNECT parameter does not point to a data block.
809A	The parameter CONNECT parameter points to a field that does not match the length of the connection description (UDT65).
809B	The local_device_id in the connection description does not match the CPU.

ERROR_TCON (W#16#...)	
80A0	Group error for error codes W#16#80A1 and W#16#80A2
80A1	Connection or port is already occupied by user
80A2	Local or remote port is occupied by the system
80A3	Attempt being made to re-establish an existing connection
80A4	IP address of the remote connection end point is invalid, it may match the local IP address
80A7	Communications error: you have called TDISCON before TCON was complete. TDISCON must first complexly terminate the connection referenced by the ID.
80B2	The parameter CONNECT parameter points to a data block that was generated with the keyword UNLINKED.
80B3	Inconsistent parameter assignment: Group error for the error codes W#16#80A0 to W#16#80A2, W#16#80A4, W#16#80B4 to W#16#80B9
80B5	Error in active_est parameter (UDT 65) in the UDP protocol variant
80B6	Parameter assignment error relating to the connection_type parameter (UDT 65)
80B7	Error in one of the following parameters of UDT 65: block_length, local_tsap_id_len, rem_subnet_id_len, rem_staddr_len, rem_tsap_id_len, next_staddr_len
80B8	Parameter ID in the local connection description (UDT 65) and parameter ID are different
80C3	All connection resources are in use.
80C4	Temporary communication error: <ul style="list-style-type: none"> <li>- The connection cannot be established at this time.</li> <li>- The interface is receiving new parameters.</li> <li>- The configured connection is currently being removed by a TDISCON.</li> <li>- The H system is connecting and updating.</li> </ul>

### Other documentation



For more information, please see the SIEMENS documentation about the TCON block.

ERROR_TSEND (W#16#...)	
8085	LEN parameter has the value 0 or is greater than the largest permitted value
8086	The ID parameter is not in the permitted address range
8088	LEN parameter is larger than the memory area specified in DATA

ERROR_TSEND (W#16#...)	
8089	ADDR parameter does not point to a data block
80A1	Communication error: - The specified connection between user program and communications layer of the operating system has not yet been established. - The specified connection between the user program and the communication level of the operating system is currently being terminated. Transmission over this connection is not possible. - The interface is being reinitialized.
80A4	IP address of the remote connection end point is invalid, it is possible that it matches the local IP address.
80B3	The set protocol variant (connection_type parameter in the connection description) is not UDP. Please use FB 63 "TSEND". ADDR parameter: Invalid settings for port no.
80C3	A block with this ID is already being processed in a different priority class. Internal lack of resources.
80C4	Temporary communication error: - The connection between the user program and the communication level of the operating system cannot be established at this time. - The interface is receiving new parameters.

### Other documentation



For more information, please see the SIEMENS documentation about the TSEND block.

### 5.7.3 More error messages

#### ERROR TCON

##### ERROR and STATUS parameters

ERROR	STATUS* (W#16#...)	Explanation
0	0000	Connection successfully established.
0	7000	No job processing active
0	7001	Start job execution, establish connection.
0	7002	Connection is being established (REQ irrelevant).
1	8085	Connection ID (ID parameter) is already being used by a configured connection.
1	8086	The ID parameter is outside the valid range.
1	8087	Maximum number of connections reached; no additional connection possible
1	8089	The CONNECT parameter does not point to a connection description or the connection description was created manually.
1	809A	The structure at the CONNECT parameter is not supported on an integrated interface or the length is invalid.
1	809B	The element Interfaceld within the TCON_xxx structure does not reference a hardware identifier of a CPU or CM/CP interface or has the value "0".
1	80A1	The specified connection or the port is already being used.
1	80A2	Local or remote port is being used by the system. The following ports are reserved locally: 20, 21, 80, 102, 135, 161, 162, 443, 34962, 34963, 34964 as well as the area 49152 to 65535.
1	80A3	ID is used by a connection created by the user program, which uses the same connection description at the CONNECT parameter.
1	80A4	IP address of the remote endpoint of the connection is invalid or it corresponds to the IP address of the local partner.
1	80A7	Communication error: You executed <a href="#">"TDISCON"</a> before "TCON" had completed.

1	80B4	Only with TCON_IP_RFC: The local T selector was not specified or the first byte does not contain the value 0x0E (only with a length of T selector = 2) or the local T selector starts with "SIMATIC-".
1	80B5	Only passive connection establishment is permitted for connection type 13 = UDP (parameter ActiveEstablished of the structure TCON_IP_v4 / TCON_PARAM has the value TRUE).
1	80B6	Parameter assignment error in the ConnectionType parameter of the data block for connection description. <ul style="list-style-type: none"> <li>Only valid with TCON_IP_v4: 0x11, 0x0B and 0x13.</li> <li>Only valid with TCON_IP_RFC: 0x0C and 0x12</li> </ul>
1	80B7	With TCON_IP_v4: <ul style="list-style-type: none"> <li>TCP (active connection establishment): Remote port is "0".</li> <li>TCP (passive connection establishment): Local port is "0".</li> <li>UDP: Local port is "0".</li> <li>IP address of the partner end point was set to 0.0.0.0.</li> </ul> With TCON_IP_RFC: <ul style="list-style-type: none"> <li>Local (LocalTSelector) or remote (RemoteTSelector) T selector was specified with a length of more than 32 bytes.</li> <li>For TSelLength of the T selector (local or remote), a length greater than 32 was entered.</li> <li>Error in the length of the IP address of the specific connection partner.</li> <li>IP address of the partner end point was set to 0.0.0.0.</li> </ul>
1	80B8	Parameter ID in the local connection description (structure at CONNECT parameter) and parameter ID of the instruction are different.
1	80C3	All connection resources are assigned, or ports may be dynamically used by other applications or connections.
1	80C4	Temporary communication error: <ul style="list-style-type: none"> <li>The connection cannot be established at this time.</li> <li>The connection cannot be established because the firewalls on the connection path are not open for the required ports.</li> <li>The interface is currently receiving new parameters.</li> <li>The configured connection is currently being removed by a "<a href="#">TDISCON</a>" instruction.</li> </ul>
1	80C5	The connection partner refuses to establish the connection, has terminated the connection or actively ended it.
1	80C6	The connection partner cannot be reached (network error).
1	80C7	Execution timeout.

1	80C8	Value at the ID parameter is already being used by a connection that was created using the user program. The connection uses the identical ID, but different connection settings at the parameter CONNECT.
1	80C9	Validation of the connection partner failed. The connection partner that wants to establish the connection does not match the defined partner of the structure at the CONNECT parameter.
1	80CE	The IP address of the local interface is 0.0.0.0.
1	80D0	In connection with TCP and the active connection end point: The remote_qdn parameter is an empty string. In this case, no connection can be established.
1	80D1	The remote_qdn parameter is not a fully qualified domain name. The period at the end may be missing.
1	80D2	No DNS server address is configured.
1	80D3	The fully qualified domain name could not be resolved. Possible causes: <ul style="list-style-type: none"> <li>• The DNS server is not reachable, for example, because it has been shut down or the remote port is not reachable.</li> <li>• An error occurred during communication with the DNS server.</li> <li>• The DNS server returned a valid DNS answer, but the answer contained no IPv4 address.</li> </ul>
1	80E0	Unsuitable or poor message was received.
1	80E1	Error during the handshake. Possible causes: <ul style="list-style-type: none"> <li>• Abort by the user</li> <li>• Security not high enough</li> <li>• Renewed negotiation is not supported</li> <li>• SSL/TLS version is not supported</li> <li>• Validation of the host name failed</li> </ul>
1	80E2	Not supported / invalid certificate Possible cause: The time-of-day of the module concerned is not set or the module is not synchronized. Example: The default setting for the date of the module is 1/1/2012 and it was not set during commissioning. The validity period of the certificate starts on 20 August 2016 and ends on 20 August 2024. In this case, the date of the module is outside the validity period of the certificate; the certificate is invalid for the module.
1	80E3	Certificate was discarded.
1	80E4	No valid certification authority found.
1	80E5	Certificate expired.
1	80E6	Integrity errors in the Transport Layer Security Protocol
1	80E7	Not supported extension in X.509-V3 certificate
1	80E9	TLS server without server certificate is not supported.
1	80EA	DTLS (UDP) protocol is not supported.
1	80EB	A client cannot request a client certificate.
1	80EC	The server cannot perform validation based on the subjectAlternateName (only clients can do this).
1	80ED	TLSServerCertRef_m-ID invalid

### Other documentation



For more information, please see the SIEMENS documentation about the TCON block.

## ERROR TUSEND

ERROR	STATUS* (W#16#...)	Explanation
0	0000	Send job completed without error
0	7000	No job processing active
0	7001	Start of job processing, data being sent Note: During this processing phase, the operating system accesses the data in the DATA send area.
0	7002	Intermediate call (REQ irrelevant), job is being processed Note: During this processing phase, the operating system accesses the data in the DATA send area.
1	8085	The LEN parameter has the value "0" or is greater than the highest permitted value.
1	8086	The ID parameter is not in the permitted value range.
0	8088	The LEN parameter is greater than the memory area specified in DATA.
1	8089	The parameter ADDR does not point to a data block with the structure TADDR_Param or TADDR_SEND_QDN.
1	80A1	Communication error: <ul style="list-style-type: none"> <li>• The specified connection between user program and communication layer of the operating system has not yet been established.</li> <li>• The specified connection between the user program and the communication layer of the operating system is currently being terminated. Transmission over this connection is not possible.</li> <li>• The interface is being reinitialized.</li> </ul>
1	80B1	You changed the DATA parameter before the current job finished.
1	80A4	IP address (at the ADDR parameter) of the remote connection end point is invalid; it may correspond to the local partner's own IP address.
1	80B3	<ul style="list-style-type: none"> <li>• The protocol variant (connection_type parameter in the connection description) is not set to UDP. Please use <a href="#">"TSEND"</a>.</li> <li>• Parameter ADDR: Invalid information for port no.</li> </ul>
1	80B7	The length of the structure referenced by the parameter ADDR is not 8 bytes.
1	80C3	<ul style="list-style-type: none"> <li>• A block with this ID is already being processed in a different priority class.</li> <li>• Internal lack of resources.</li> </ul>

## Other documentation



For more information, please see the SIEMENS documentation about the TUSEND block.

## 6 Support and contact

### Support

Phone: +49 911 97282-14  
Fax: +49 911 97282-33  
Email: support@iba-ag.com

---

#### Note



If you require support, indicate the serial number (iba-S/N) of the product or the license number.

---

### Contact

#### Headquarters

iba AG  
Koenigswarterstrasse 44  
90762 Fuerth  
Germany

Phone: +49 911 97282-0  
Fax: +49 911 97282-33  
Email: iba@iba-ag.com

#### Mailing address

iba AG  
Postbox 1828  
D-90708 Fuerth, Germany

#### Delivery address

iba AG  
Gebhardtstrasse 10  
90762 Fuerth, Germany

#### Regional and Worldwide

For contact data of your regional iba office or representative please refer to our web site

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