



ibaPDA-Request-TwinCAT

Request Data Interface to TwinCAT Systems

Manual
Issue 1.3

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

Manufacturer

iba AG
Koenigswarterstr. 44
90762 Fuerth
Germany

Contacts

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

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

© iba AG 2021, All rights reserved.

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

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

Version	Date	Revision - Chapter / Page	Author	Version SW
1.3	11-21	Data path UDP Realtime	St	7.3.8

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

Content

1	About this Manual	5
1.1	Target group and previous knowledge	5
1.2	Notations	6
1.3	Used symbols.....	7
2	System requirements	8
3	ibaPDA-Request-TwinCAT	10
3.1	General Information	10
3.2	Request blocks.....	11
4	Request TwinCAT via EtherCAT	14
4.1	System integration with data path EtherCAT	14
4.2	Configuration and engineering of the TwinCAT controller	14
4.2.1	TwinCAT 2	15
4.2.2	TwinCAT 3	18
4.3	Configuration in ibaPDA.....	21
4.3.1	Setting up the connection.....	21
4.3.2	TwinCAT request module.....	23
4.3.2.1	General Settings.....	23
4.3.2.2	Configuration of the control path.....	24
4.3.2.3	Configuration of the data path	30
4.3.3	Selecting symbols	31
4.3.4	Diagnostics.....	32
5	Request TwinCAT via UDP	33
5.1	System integration with data path UDP and UDP realtime (RT)	33
5.2	Configuration and engineering of the TwinCAT controller	33
5.2.1	TwinCAT 2	33
5.2.2	TwinCAT 3	34
5.3	Configuration in ibaPDA.....	35
5.3.1	Setting up the connection.....	35
5.3.2	TwinCAT request module.....	36
5.3.2.1	General Settings.....	36
5.3.2.2	Configuration of the control path.....	36

- 5.3.2.3 UDP and UDP RT data path..... 36
- 5.3.3 Selecting symbols 37
- 6 Diagnostics..... 38**
 - 6.1 Checking the license 38
 - 6.2 Log files..... 38
 - 6.3 Connection diagnostics with PING..... 39
- 7 Technical data 40**
- 8 Support and contact..... 41**



1 About this Manual

This document describes the function and application of the software interface *ibaPDA-Request-TwinCAT*.

The product *ibaPDA-Request-TwinCAT* is an extension of *ibaPDA* for the optional access to variables when recording data from TwinCAT 2 and TwinCAT 3 controllers. This manual only shows the extensions and differences. Refer to the manual from *ibaPDA* for all other functions and operating options.

1.1 Target group and previous knowledge

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

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

- Basic knowledge of *ibaPDA*
- Basic knowledge of network technology
- Knowledge of configuration and operation of the relevant control system

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

- *ibaPDA* v7.3.8 or higher
- Basic license for *ibaPDA*
- Additional license for *ibaPDA-Request-TwinCAT*
- Additional licenses if UDP is used as data path:
 - *ibaPDA-Interface-Generic-UDP*
 - Beckhoff TwinCAT TCP/IP server license on the run time system:
TS6310 for TwinCAT 2, TF6310 or TF6311 for TwinCAT 3
- Additionally required if EtherCAT is used as data path:
 - *ibaBM-eCAT*
 - FO card of the type *ibaFOB-D* or USB adapter *ibaFOB-io-USB* in the *ibaPDA* computer
- Beckhoff controller TwinCAT 2 or TwinCAT 3
- The TwinCAT ADS Communication Library from Beckhoff must be installed on the *ibaPDA* computer. If TwinCAT 2 or 3 is installed on the computer on which the *ibaPDA* service is running, then the library is normally available.

If TwinCAT is not installed on the *ibaPDA* computer, you have to download TwinCAT 3 ADS Runtime from the Beckhoff website <http://www.beckhoff.com>. Start the installation program and select full installation. TwinCAT 3 ADS Runtime also supports TwinCAT 2.
- Ethernet connection to the controller
- Libraries with iba Request blocks
 - TwinCATRequestLibCommon.lib for shared blocks and the connection via EtherCAT
 - TwinCATRequestLibUDP.lib for the connection via UDP (Beckhoff TF6310)
 - TwinCATRequestLibUDPRT.lib for the connection via UDP Realtime (Beckhoff TF6311)

System specification

- A maximum of 64 request blocks are supported per controller
- A maximum of 500 signals (analog or digital) or a maximum of 2000 bytes are possible per request block
- A maximum of 512 analog signals and 512 digital signals are supported per *ibaBM-eCAT* device

Licenses

Order no.	Product name	Description
31.001303	ibaPDA-Request-TwinCAT	Extension license for an ibaPDA system to be able to use the request functionality with Beckhoff TwinCAT controllers
31.001075	ibaPDA-Interface-Generic-UDP	Extension license for an ibaPDA system for a generic UDP interface Number of connections: 64

Table 1: Available licenses

Hardware

Order no.	Product name	Description
13.127000	ibaBM-eCAT	Bus monitor for EtherCAT

Table 2: Hardware

3 ibaPDA-Request-TwinCAT

3.1 General Information

The interface *ibaPDA-Request-TwinCAT* is suitable for the measuring data acquisition with a free symbol selection from Beckhoff-TwinCAT controllers via EtherCAT or Ethernet (UDP/IP). The measuring data is actively sent here from the controller to *ibaPDA*. For this purpose, it is necessary to integrate request blocks into the TwinCAT controller. These request blocks serve to cyclically send the current values of the variables selected by the user within *ibaPDA* to *ibaPDA* for recording.

In *ibaPDA*, the variables to be measured are selected with a browser. This makes it possible to access all of the variables available in the controller. The values of the variables can be sent to *ibaPDA* via three different data paths:

- EtherCAT connection via *ibaBM-eCAT*
- UDP connection via *ibaPDA-Interface-Generic-UDP*
- UDP realtime connection via *ibaPDA-Interface-Generic-UDP*

ibaPDA-Request-TwinCAT supports TwinCAT 2 and 3 on industrial PCs and embedded PCs. Bus terminal controllers of the BX/BC series are not supported.

An ibaTwinCAT library must be added to the project in the TwinCAT controller:

- TwinCATRequestLibCommon library for shared blocks and support of EtherCAT as data path
- TwinCATRequestLibUDP library supports UDP as data path. For communication via UDP, the TwinCAT TCP/IP server library must be integrated. This library requires a license and must be installed separately:
 - TS6310 for TwinCAT 2 controllers
 - TF6310 for TwinCAT 3 controllers
- TwinCATRequestLibUDPRT library supports UDP realtime as data path. For communication via UDP realtime, the TwinCAT TCP/IP server library must be integrated. This library requires a license and must be installed separately:
 - TF6311 for TwinCAT 3 controllers

The ibaTwinCATRequest libraries contain the following function blocks:

- Management block IBA_TCREQ_MAN
- Signal data blocks IBA_TCREQ_DATA_ECANT, IBA_TCREQ_DATA_UDP and IBA_TCREQ_DATA_UDPRT

The management block can also be inserted in a (slow) task with low priority. It communicates with *ibaPDA* (via ADS communication) via the control path and checks the list of variables.

The signal data block is assigned to a faster task with a higher priority. It collects the data and sends it to *ibaPDA* with each access on the data path. The EtherCAT signal data block is recommended if you want to measure very short cycle times.

You can find the libraries as archive files on the DVD or USB stick “iba Software & Manuals” at
 \04_Libraries_and_Examples\10_Libraries\04_TwinCAT\ibaTwinCATLib_Vx.y.z.zip

The libraries are available in different versions:

- TwinCAT 2
 - ibaTwinCATRequest.lib (for EtherCAT)
 - ibaTwinCATRequestWithUDP.LIB (for UDP)
- TwinCAT 3
 - TwinCATRequestLibCommon.compiled-library (shared blocks and EtherCAT)
 - TwinCATRequestLibUDP.COMPILED-LIBRARY (for UDP)
 - TwinCATRequestLibUDPRT.COMPILED-LIBRARY (for UDP realtime)

3.2 Request blocks

The request blocks are used to initialize and control the communication between the TwinCAT controller and *ibaPDA*.

A request block set always consists of a management block and a signal data block. There are separate signal data blocks for the connection via EtherCAT and UDP or UDP realtime. The blocks are components of the ibaTwinCAT libraries.

Management block IBA_TCREQ_MAN

Name	Type	In/Out	Description
Name	STRING(20)	IN	Name of the function block. The same name must be used for the corresponding IBA_TCREQ_DATA function block.
State	IBA_TCREQ_STATE_MAN	OUT	Status of the function block

The block IBA_TCREQ_MAN may assume the following states (IBA_TCREQ_STATE_MAN):

Status	Description
TCREQ_MAN_INIT	Initial state before the block has registered with its name
TCREQ_MAN_IDLE	Waiting for messages from ibaPDA
TCREQ_MAN_VALIDATING	Validating of the tags received by ibaPDA
TCREQ_MAN_RELEASING_HANDLES	Releasing handles to variables

Signal data block IBA_TCREQ_DATA_ECAT

Name	Type	In/Out	Description
Name	STRING(20)	IN	Name of the function block. The same name must be used for the corresponding IBA_TCREQ_MAN function block.
Data buffer	POINTER TO BYTE	IN	Pointer to the data buffer where the values of the requested variables should be written to. This data buffer must be linked with EtherCAT output variables.
MaxDataSize	UINT	IN	The size of the data buffer
State	IBA_TCREQ_STATE_DATA	OUT	Status of the function block
ADSError	UDINT	OUT	The last error code that was received when reading the data of the requested variables.
Size	UINT	OUT	The actual size of the data written in the data buffer.

Signal data blocks IBA_TCREQ_DATA_UDP and IBA_TCREQ_DATA_UDPRT

Name	Type	In/Out	Description
Name	STRING(20)	IN	Name of the function block. The same name must be used for the corresponding IBA_TCREQ_MAN function block.
Data buffer	POINTER TO BYTE	IN	Pointer to the data buffer where the values of the requested variables should be written to.
MaxDataSize	UINT	IN	The size of the data buffer.
Adapter	STRING(15)	IN	The IP address of the network adapter that needs to be used to send data to ibaPDA. If this is empty, the default adapter is used.
State	IBA_TCREQ_STATE_DATA	OUT	Status of the function block
ADSError	UDINT	OUT	The last error code that was received when reading the data of the requested variables.
Size	UINT	OUT	The actual size of the data written in the data buffer.
UsedAdapter	STRING(15)	OUT	The IP address of the network adapter that is used to send data to ibaPDA.
SendCounter	UINT	OUT	Counter that is incremented every time a message is sent to ibaPDA.

The signal data blocks may assume the following states (IBA_TCREQ_STATE_DATA):

Status	Description
TCREQ_DATA_INIT	Initial state. Search for the block with the same name IBA_TCREQ_MAN.
TCREQ_DATA_NO_PATH	Connected to the management block, but no data path available.
TCREQ_DATA_READY	Connected to the management block and data path found. The variables list is empty.
TCREQ_DATA_COPYING	Copying data for the variables list.
TCREQ_DATA_ONLINECHANGE	An online change has occurred. Waiting for the management block to react to this.

4 Request TwinCAT via EtherCAT

4.1 System integration with data path EtherCAT

The measurement data is transmitted via EtherCAT to the *ibaBM-eCAT* device.

You need the following connections:

- Ethernet connection between *ibaPDA* and the TwinCAT PLC
- Fiber optic connection between *ibaPDA/ibaFOB-io-D* and *ibaBM-eCAT*
- EtherCAT connection between *ibaBM-eCAT* and TwinCAT PLC

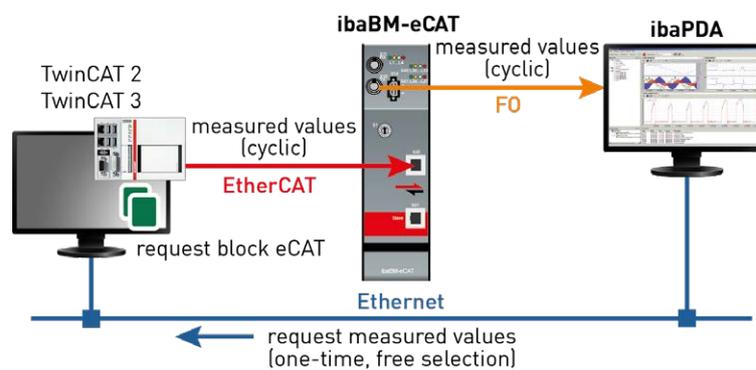


Fig. 1: Topology request via EtherCAT

An additional prerequisite is the *ibaTwinCAT* request library in the TwinCAT controller.

4.2 Configuration and engineering of the TwinCAT controller

The following configuration and engineering steps are generally to be made in TwinCAT on the TwinCAT side:

Hardware engineering

Integration of the *ibaBM-eCAT* device in the device configuration. You can find a description of this in the corresponding chapters of the *ibaBM-eCAT* manual *System integration* and *Configuration in the EtherCAT-Master*.

The only difference when using Request-TwinCAT is the selection of the request device (*ibaBM-eCAT for TwinCAT Vx Request*) and the resulting signal grouping in the EtherCAT configuration.

Software engineering

The integration of the request blocks in the TwinCAT program is described in the following chapters.

4.2.1 TwinCAT 2

Open the library manager in your project and add the library *ibaTwinCATRequest.lib* from the directory ...\\Vx.y.z\\TwinCAT_V2. Now add an instance of the management block IBA_TCREQ_MAN and an instance of the signal data block IBA_TCREQ_DATA_ECATCH. You can add the instances to the same program or to different programs.

Create a buffer that is linked with EtherCAT output variables. The syntax AT %Q* labels the data buffer as an output variable. The size of the buffer should be between 256 and 2560 bytes. The size depends on how much data you would like to measure via this request function block pair.

It is recommended to create 2 pairs of blocks to utilize the maximum data volume.

For simplicity's sake, both blocks are accessed in one task in this example.

```
PROGRAM ibaRequest
VAR

    DataBuffer AT %Q*: ARRAY[0..2559] OF BYTE;

    ibaReqMan_1 : IBA_TCREQ_MAN;
    ibaReqData_1 : IBA_TCREQ_DATA_ECATCH;

    ibaReqMan_2 : IBA_TCREQ_MAN;
    ibaReqData_2 : IBA_TCREQ_DATA_ECATCH;

END_VAR
```

You call them as follows:

```
ibaReqMan_1(
    Name:= 'RequestData_1',
    State=>);

ibaReqData_1(
    Name:= 'RequestData_1',
    DataBuffer:= ADR(DataBuffer[0]),
    MaxDataSize:= 1280,
    State=>,
    ADSError=>,
    Size=>);

ibaReqMan_2(
    Name:= 'RequestData_2',
    State=>);

ibaReqData_2(
    Name:= 'RequestData_2',
    DataBuffer:= ADR(DataBuffer[1280]),
    MaxDataSize:= 1280,
    State=>,
    ADSError=>,
    Size=>);
```

Now link the data buffer variable with the EtherCAT variables from *ibaBM-eCAT*. To do this, open the system manager and add the device *ibaBM-eCAT for TwinCAT V2 request* to your EtherCAT network.

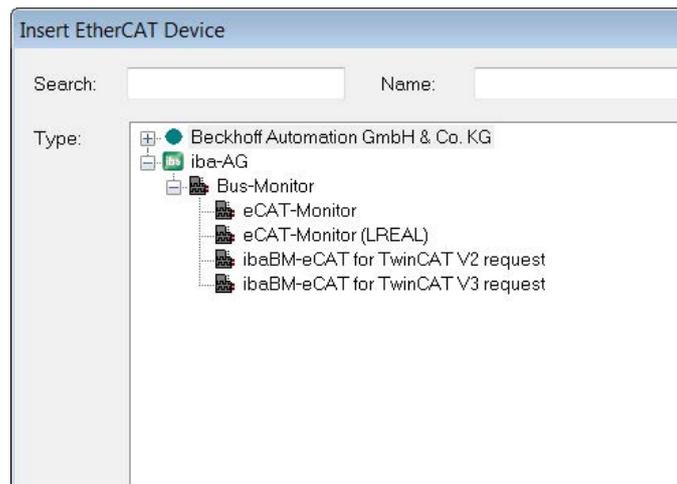


Fig. 2: Add ibaBM-eCAT

ibaBM-eCAT has 10 request slots. Each request slot consists of 64 DWORD variables, which correspond to 256 bytes. The DWORD variables can be linked with arrays. DWORD variables are used instead of BYTE variables in order to reduce the number of variables.

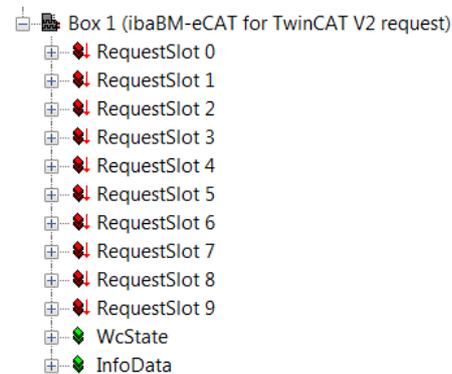


Fig. 3: 10 request slots per device

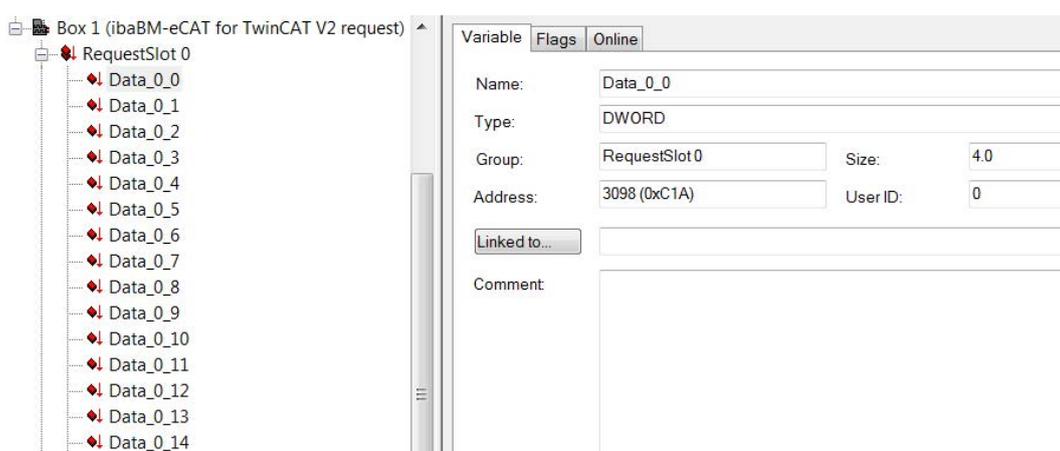


Fig. 4: Data type of the variables

If you select the *ibaBM-eCAT* device in the tree, a table with all variables will be shown on the right side. You can hold down the Shift key to select several variables, which are to be linked with the data buffer.

Click the right mouse button and select *Change multi-link...* in the context menu. The multi-link command is only active if you select variables from the same EtherCAT telegram. By default, the first 5 slots in the first EtherCAT telegram and the last 5 slots in a second EtherCAT telegram are sent.

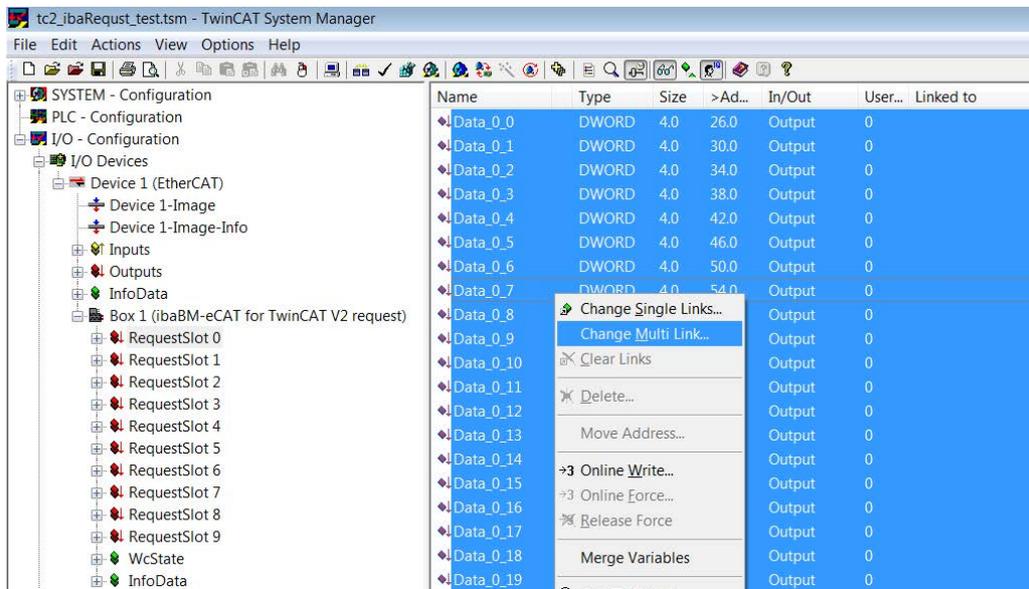


Fig. 5: Attach variables

The dialog “Attach variable” is opened.

Double click on the data buffer array to begin attaching from the first byte, or select some elements of the data buffer to attach them to other parts of the data buffer.

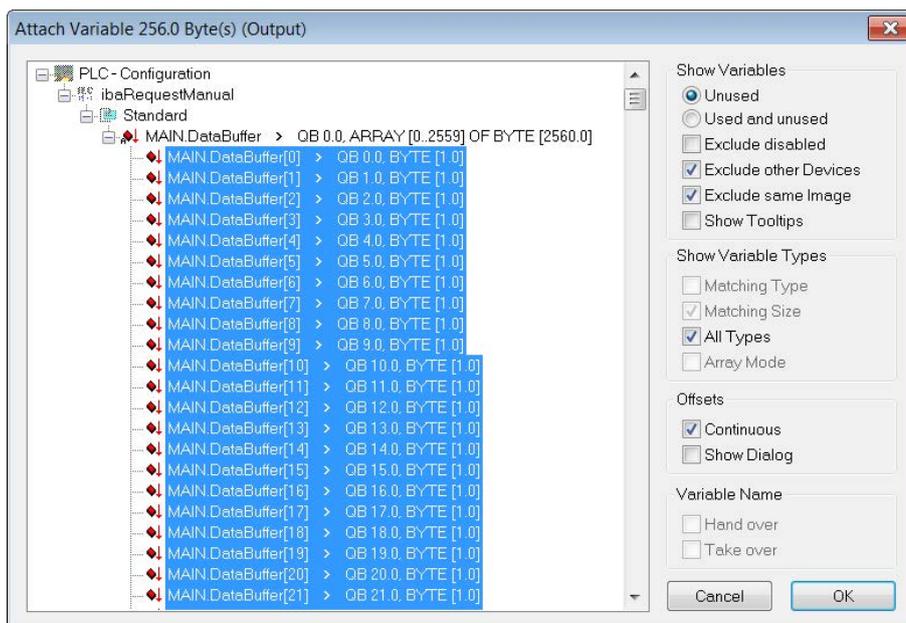


Fig. 6: Attach variables

If you have selected and linked all bytes, the next bytes appear in the selection list when selecting the following slot, here starting at byte 256.

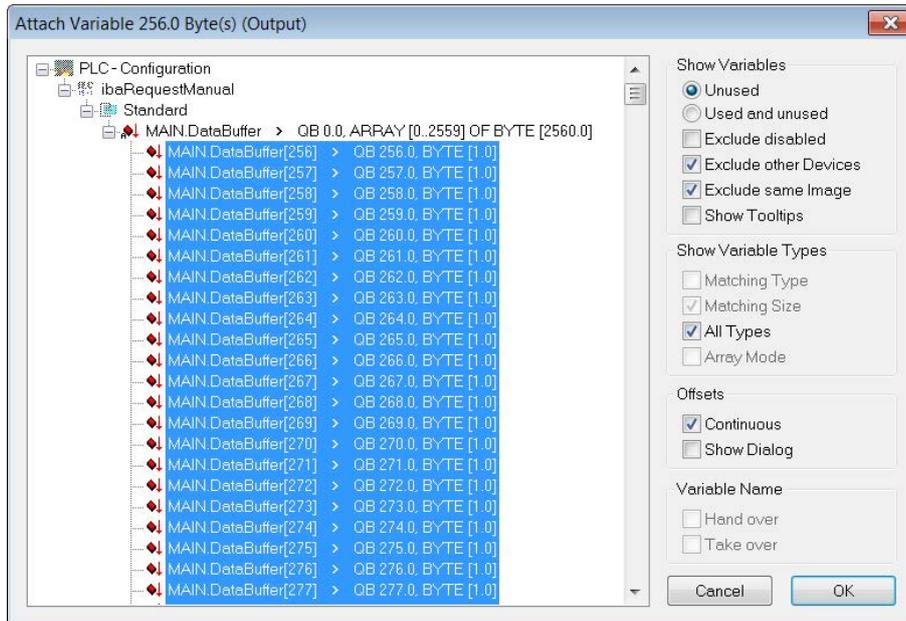


Fig. 7: Attaching the following variables

If you link the variables in another way, please do not attach more than 1024 bytes at once. Then enable the configuration and download your program.

4.2.2 TwinCAT 3

Add the library *TwinCATRequestLibCommon.compiled-library* from the directory ...\\Vx.y.z\\TwinCAT_V3 to your project. Now add an instance of the management block IBA_TCREQ_MAN and an instance of the signal data block IBA_TCREQ_DATA_ECATCH.

In addition, create a buffer that is linked with EtherCAT output variables. The syntax AT %Q* labels the data buffer as an output variable. The size of the buffer should be between 256 and 2560 bytes. The size depends on how much data you would like to measure via this request function block pair.

It is recommended to create 2 pairs of blocks to utilize the maximum data volume.

```
reqManEcat : IBA_TCREQ_MAN;
reqDataEcat : IBA_TCREQ_DATA_ECATCH;
DataBufferEcat AT %Q* : ARRAY[0..2559] OF BYTE;
```

The management and signal data blocks can be found in the same program or in separate programs. You call them as follows:

```
reqManEcat(Name := 'DemoRequestECATCH');
reqDataEcat(Name := 'DemoRequestECATCH', DataBuffer := ADR(DataBufferEcat[0]), MaxDataSize := SIZEOF(DataBufferEcat));
```

In the I/O section, add an *ibaBM-eCAT* device to your EtherCAT network. First copy the ESI file of the *ibaBM-eCAT* device into the subdirectory Config/Io of your TwinCAT installation directory.

Then select *ibaBM-eCAT for TwinCAT V3 request* as a TwinCAT 3 request device and add it. *ibaBM-eCAT* has 10 request slots. Each slot consists of an array with 256 bytes. You can use one or more slots per signal data block IBA_TCREQ_DATA_ECATCH.

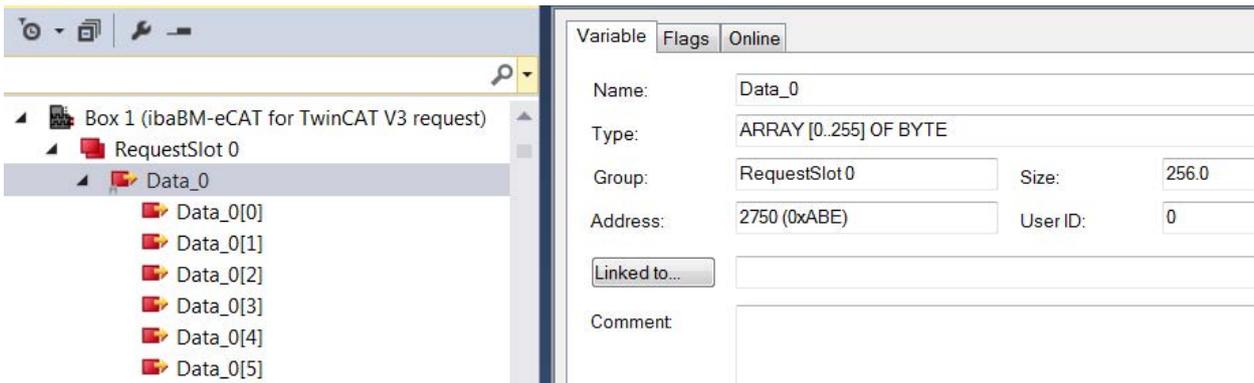


Fig. 8: Request slots in the TwinCAT 3 request device

To link the data buffer with the request slots, select the device in the tree and then select the output variables in the right table. Click the right mouse button and select *Change multi link...* in the context menu. The multi-link command is only active if you select variables from the same EtherCAT telegram. By default, the first 5 slots in the first EtherCAT telegram and the last 5 slots in a second EtherCAT telegram are sent.

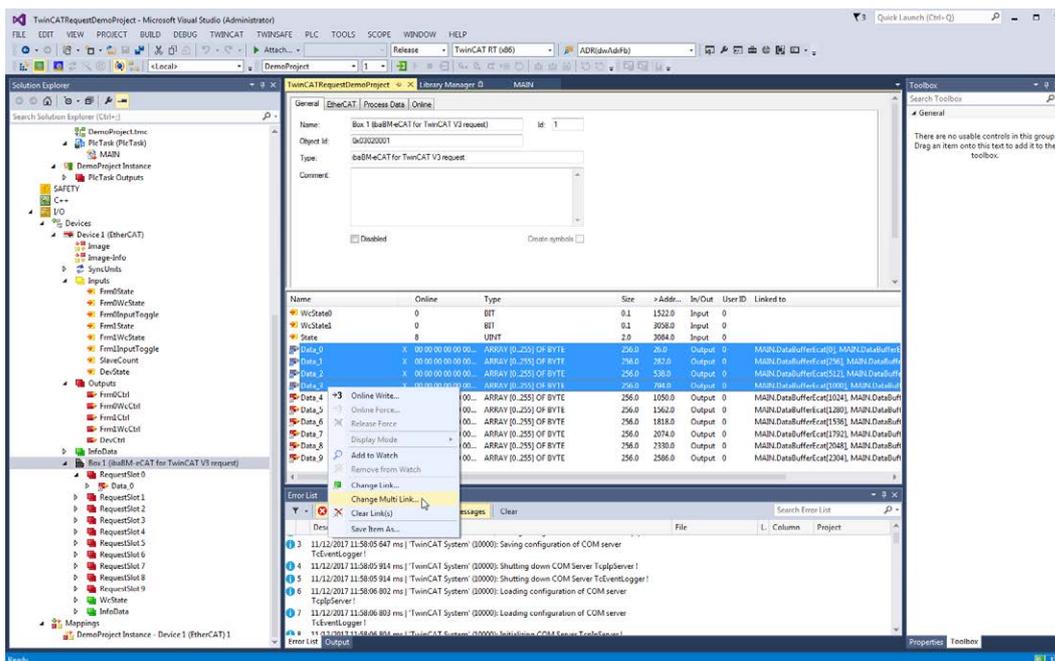


Fig. 9: Link variables

In the following dialog, select the data buffer or parts of the data buffer. This is then linked with the EtherCAT output variables.

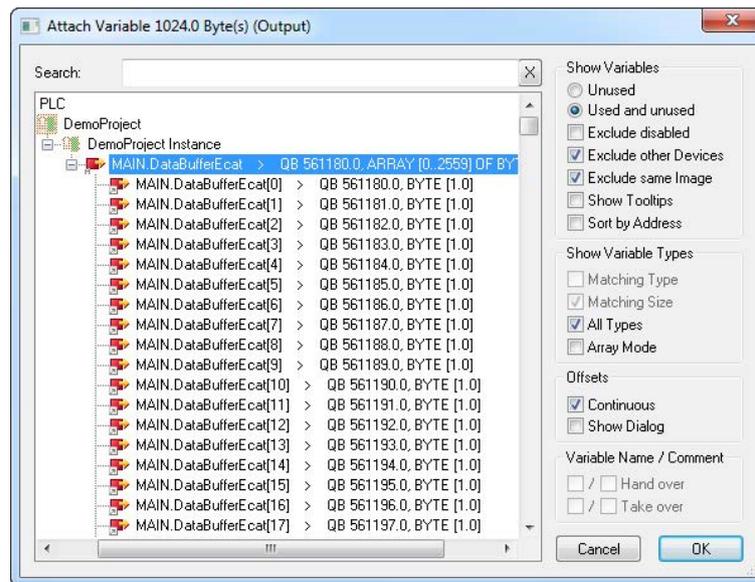


Fig. 10: Attach variables

Enable the configuration and restart the TwinCAT system in order to accept the I/O changes.

4.3 Configuration in ibaPDA

The configuration of is done in the I/O Manager of *ibaPDA*. First set up the connection from *ibaPDA* to the TwinCAT controller via *ibaBM-eCAT*.

Once the connection is set up, add a TwinCAT request module accordingly. See chapter ↗ *TwinCAT request module*, page 36.

The configuration of the signals and the selection in the symbol browser is described in the chapter ↗ *Selecting symbols*, page 31.

4.3.1 Setting up the connection

The prerequisite for using EtherCAT as data path is the bus monitor *ibaBM-eCAT* and the installation of an *ibaFOB-D* card in the *ibaPDA* computer. After successfully installing the *ibaFOB-D* card, this will be shown in the interface tree.

Other documentation



You can find additional information about the *ibaFOB-D* card in the associated device manual.

To use the TwinCAT request function, *ibaBM-eCAT* must be an active part of the EtherCAT network. That is why the option “active slave” must be enabled in *ibaPDA* and the device must be integrated into the EtherCAT configurator.

For this purpose, the associated ESI file (an XML file) from the supplied DVD or USB stick “iba Software & Manuals” must be imported into the EtherCAT configurator. Then the signals are configured and linked and exported as an ENI configuration file. The configuration file can then be loaded in *ibaPDA*. For additional information, see the chapter ↗ *Configuration and engineering of the TwinCAT controller*, page 14.

Configuring ibaBM-eCAT

1. Start *ibaPDA* and open the I/O Manager.
2. In the I/O Manager, search for the corresponding link of the *ibaFOB-D* card to which *ibaBM-eCAT* is connected. Use the right mouse button to click on the link and select “Autom. detection.”

If the device is connected and switched on correctly, the module “ibaBM-eCAT” is shown below the link of the *ibaFOB-D* card. The slot number corresponds to the device address, which is set on the rotary switch S1.

3. In the I/O Manager, open the *General* tab of the *ibaBM-eCAT* device module and select the option “True” in the “Enable active slave” field.
4. Add a TwinCAT request module to the *ibaBM-eCAT* device module by clicking under the device module (or several if you should need additional connections to other PLCs).

The following figure shows the *General* tab of the *ibaBM-eCAT* device module with the required setting as an “Active slave.” In the *General* tab, also load the ENI configuration file of the EtherCAT network by clicking on the link *Configure symbols*.



Fig. 11: Load the configuration file in the General tab

The ENI configuration file can be exported in TwinCAT 2 from the system manager and in TwinCAT 3 from the I/O section.

Example of system manager in TwinCAT 2:

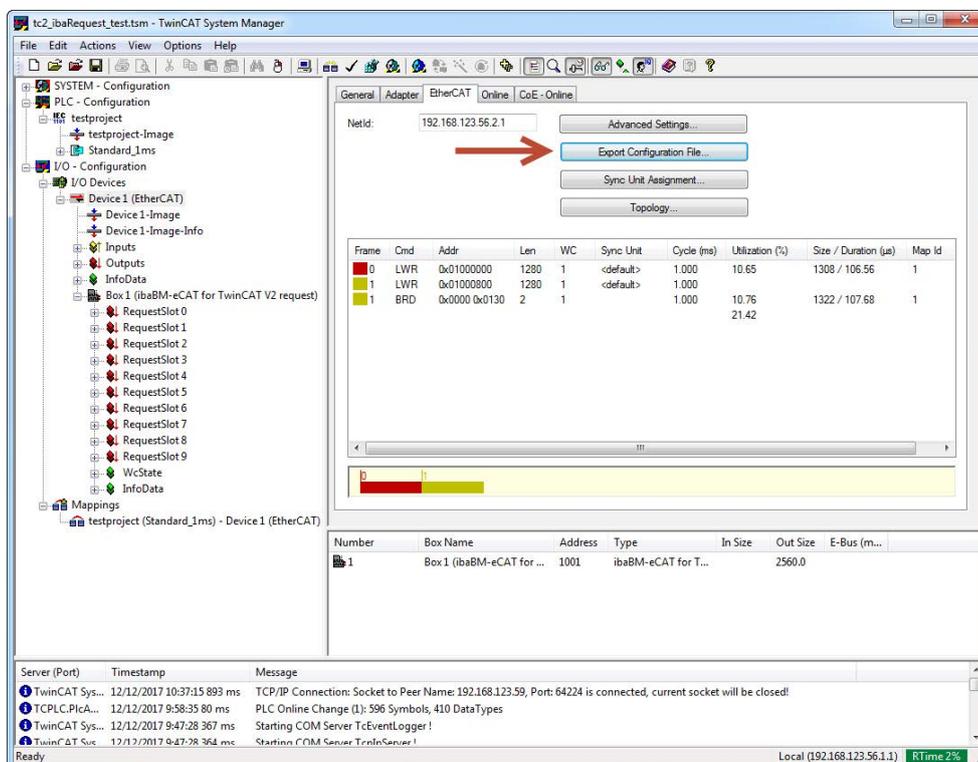


Fig. 12: Export of the ENI file in the TwinCAT system manager

4.3.2 TwinCAT request module

4.3.2.1 General Settings

The general settings are identical for the data path via EtherCAT, UDP and UDP Realtime.

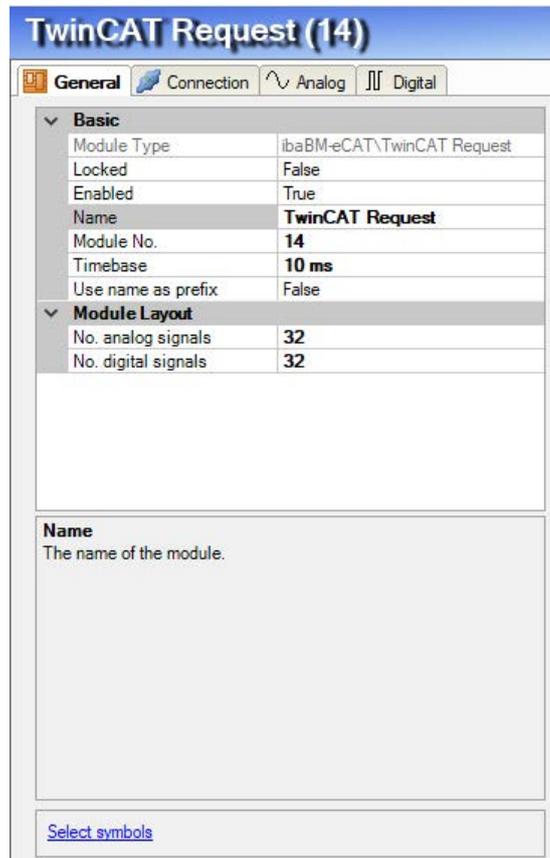


Fig. 13: TwinCAT request module, General tab

Basic settings

Module Type (information only)

Indicates the type of the current module.

Locked

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

Enabled

Disabled modules are excluded from signal acquisition.

Name

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

Module No.

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

Timebase

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

Use name as prefix

Puts the module name in front of the signal names.

Module layout

No. analog signals

Determination of the number of analog signals for this module (max. 500)

No. digital signals

Determination of the number of digital signals for this module (max. 500)

4.3.2.2 Configuration of the control path

Requirement

ibaPDA uses the TwinCAT ADS communication library from Beckhoff for the control path. This library is not part of the *ibaPDA* installation. If TwinCAT 2 or 3 is installed on the PC on which the *ibaPDA* service is running, the library is available. If TwinCAT is not installed, you have to download the TwinCAT 3 ADS runtime from the Beckhoff website (<http://www.beckhoff.com>) (see chapter ↗ *System requirements*, page 8).

Configuration

The control path is configured in the *Connection* tab and is identical to the data path via EtherCAT and UDP.

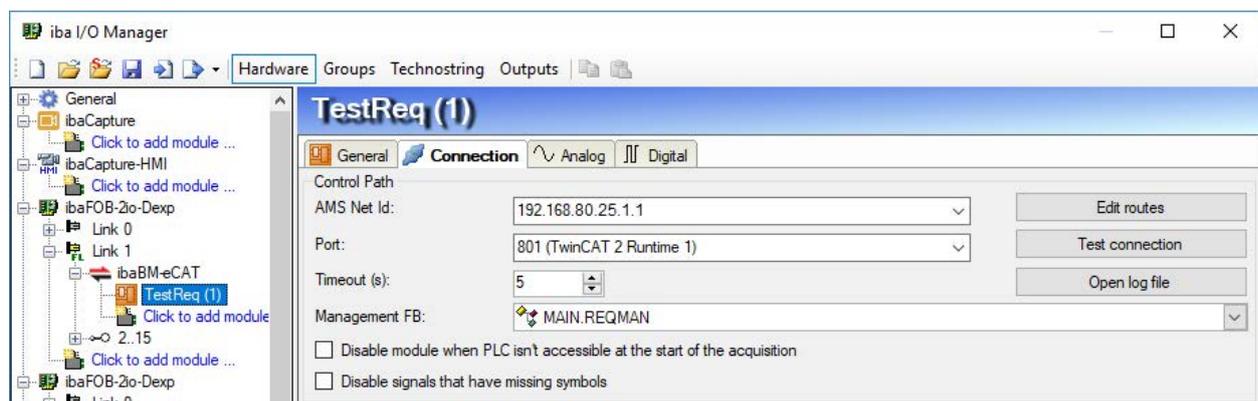


Fig. 14: Configuration of the control path

You have the following setting and configuration options in the control path section:

AMS Net-ID

The address of the control consists of the AMS net ID and a port number. The AMS net ID is the address of an ADS router and consists of 6 bytes. Normally, the first 4 bytes are the IP address and the last 2 bytes are 1.

Port

The port number determines the ADS device, which is connected to the router. There are pre-defined port numbers for the TwinCAT runtimes:

- 801, 811, 821, 831: for the 4 possible TwinCAT 2 runtimes
- 851, 852, 853, 854: for the 4 possible TwinCAT 3 runtimes

<Edit routes>

An ADS router is also installed on the *ibaPDA* server computer during the installation of the TwinCAT ADS library. An entry must be made for the remote controller in the routing table of this ADS router. In addition, an entry must be made for the AMS net ID of the *ibaPDA* server computer in the routing table of the ADS router. This can be done via the ADS router configuration tool.

Start the tool with the <Edit routes> button. The following dialog appears:

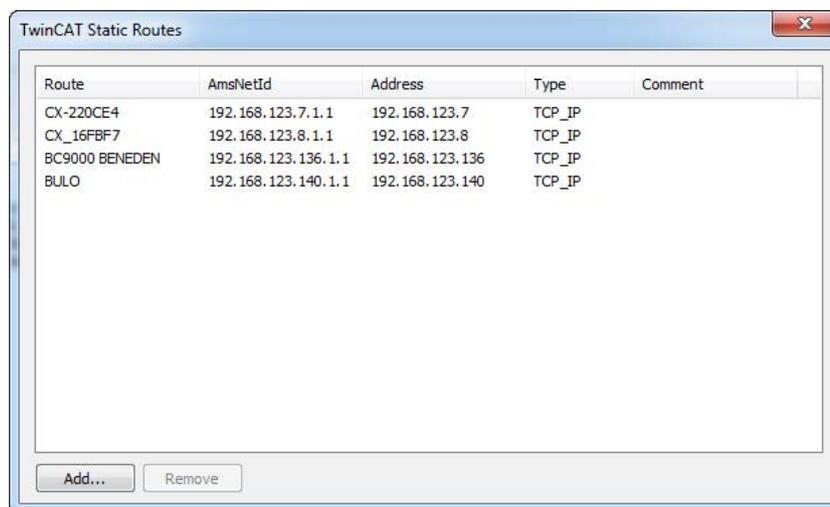


Fig. 15: Start the configuration tool

You can add a new entry using <Add>:

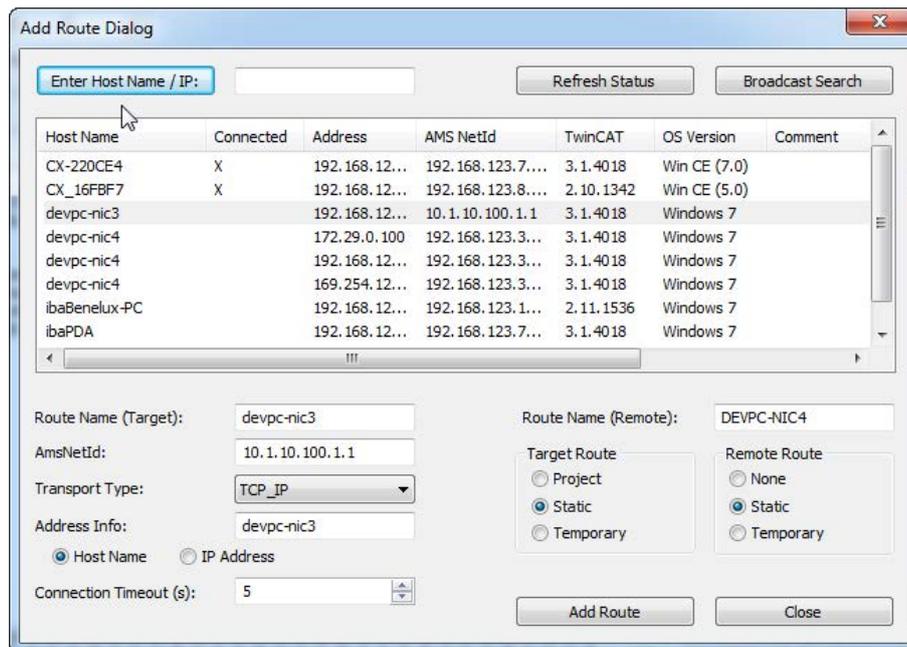


Fig. 16: Edit connections

You can search for compatible TwinCAT controllers in the local network by using the <Broadcast Search> button. Either select a found controller or enter the route name, the AmsNetId and the address information manually.

Besides the broadcast search there is also the possibility to configure a connection directly by entering the IP address:

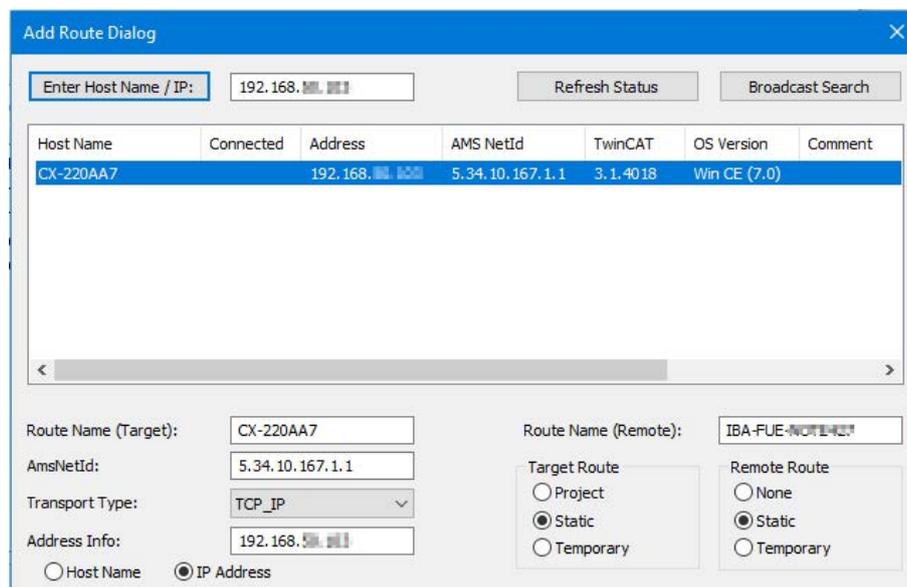


Fig. 17: Connection setting via IP address

Select "Static" for the target route. For the remote route, select "Static" when connecting to a TwinCAT connection and "None" when connecting to a bus controller. Clicking on <Add Route> will enter the route in the routing table. The router attempts to connect to the removed ADS router and may require a user name and a password. If the connection was successful, an "X" will appear in the "Connected" column.

You can also open the configuration tool via the context menu on the TwinCAT task bar symbol. Select *Router - Edit Routes*.

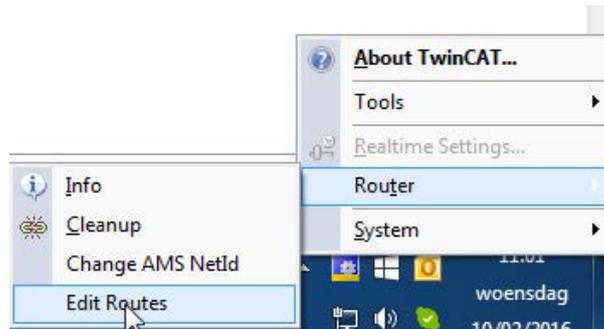


Fig. 18: Context menu TwinCAT task bar symbol

If TwinCAT 2 is installed, the following dialog appears:

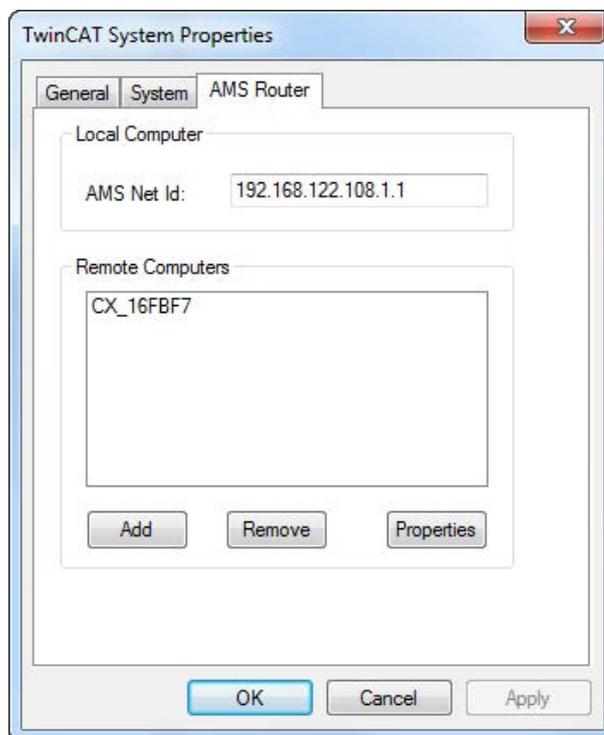


Fig. 19: System properties dialog

You can add a new route using <Add>:

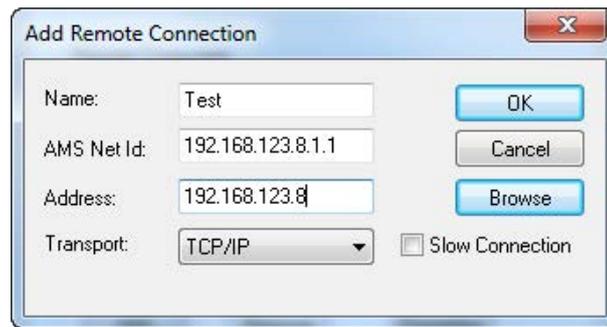


Fig. 20: Add connection

The fields Name, AMS Net ID and Address must be filled out manually. “OK” enters the route in the routing table.

You can also open the configuration tool via the context menu on the TwinCAT task bar symbol. Select *Properties*.

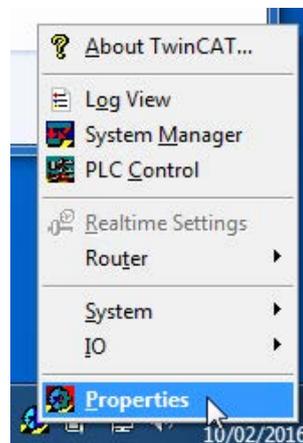


Fig. 21: Open the configuration tool

If the routing tables are correct, you can connect to the PLC with *ibaPDA*. To test the connection, select the *Connection* tab of the request module.

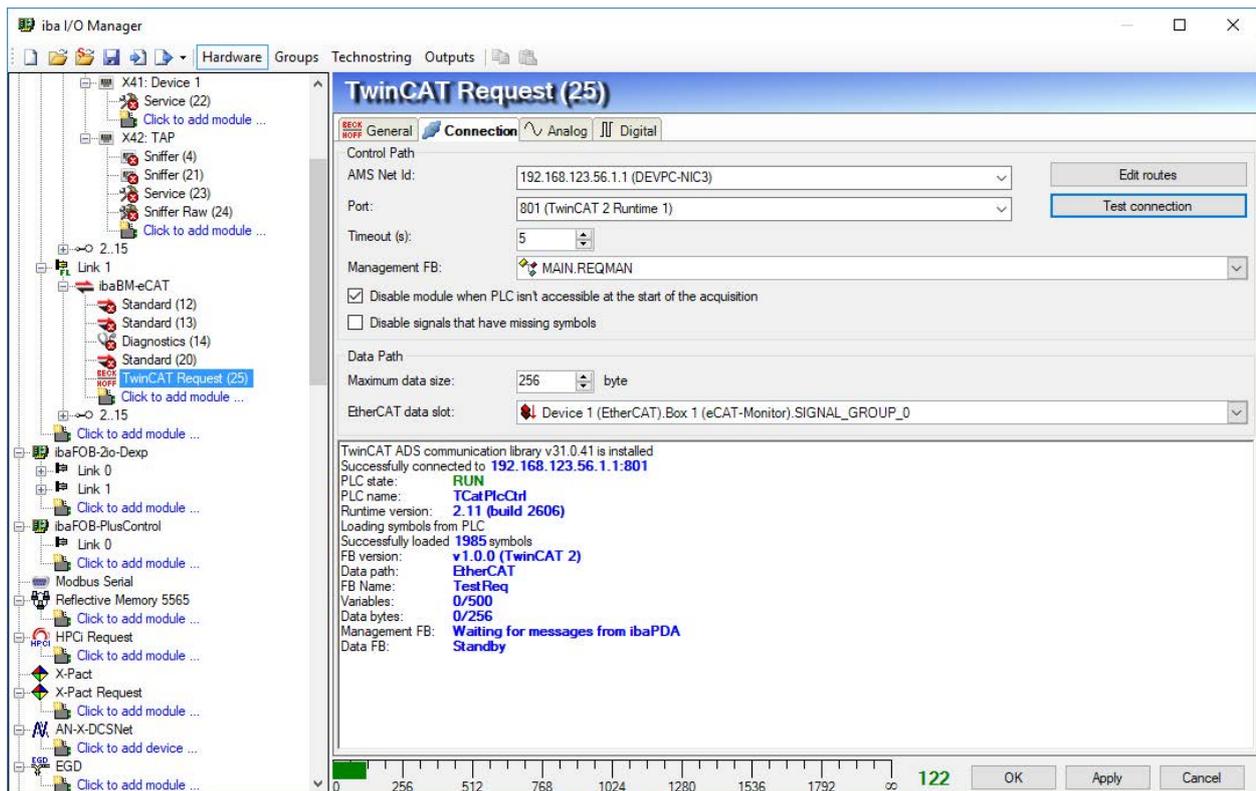


Fig. 22: TwinCAT request module, connection tab

<Test connection>

Connection test and output of available diagnostic data. If the connection is successful, *ibaPDA* loads the symbols from the controller.

Timeout

Connection timeout

Management function block

In a normal case, select the next still unused management function block from the address book. If no management function block has been configured, *ibaPDA* searches for the first instance of the IBA_TCREQ_MAN function block in the symbols and uses this. If the management function block is properly configured, *ibaPDA* reads out its properties: Version, name, data path, the number of supported variables, the number of supported data bytes and the states of the management and data function blocks.

If the name of the TwinCAT request module is still the standard name, *ibaPDA* automatically changes the name in the instance name of the management function block. The maximum data size in the data path is automatically adjusted.

<Disable module when PLC is not accessible at the start of the acquisition>

If this option is enabled, the acquisition is started, even if no connection to the PLC can be established. The module is disabled. During the acquisition, *ibaPDA* tries again to connect with the PLC. If successful, the acquisition is restarted.

If this option is not enabled, the acquisition is not started, even if no connection to the PLC is possible.

<Disable signals that have missing symbols>

If the symbol configuration has changed, the module may contain a symbol that is no longer available. If *ibaPDA* then tries to read the data for this variable, the PLC will return an error. If the option “Disable signals that have missing symbols” is enabled, *ibaPDA* ignores this signal and starts the acquisition without this signal. If this option is not enabled, the acquisition is not started.

4.3.2.3 Configuration of the data path

Carry out the settings for the data path in the *Connection* tab:

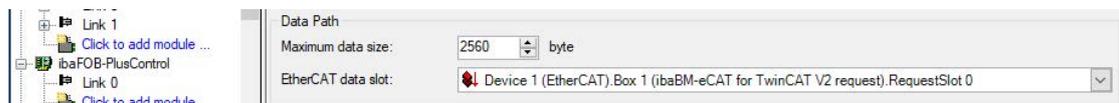


Fig. 23: Select the request slot

Maximum data size

You can set the maximum data size here.

EtherCAT data slot

All of the *ibaBM-eCAT* devices in the network are listed in the drop-down menu. Select a request slot.

Select the variables to be selected. See chapter [Selecting symbols](#), page 31.

4.3.3 Selecting symbols

Once the connection to the PLC has been successfully established, the symbols are loaded and can be selected in the symbol browser.

Open the symbol browser by clicking on the link “Select symbols” in the *General* tab of the TwinCAT request module.

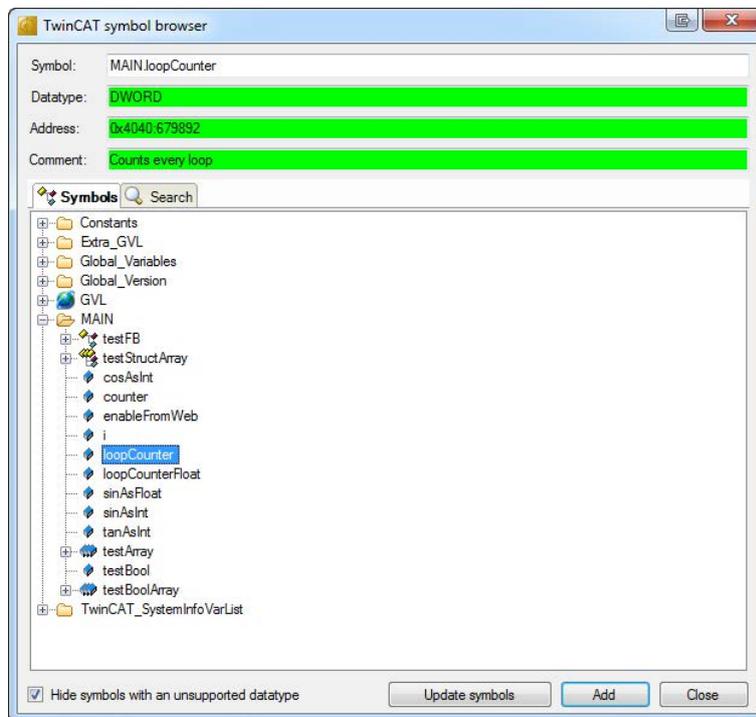


Fig. 24: TwinCAT symbol browser

In the *Symbols* tab, you can select individual or several symbols in the tree. Clicking on <Add> inserts the symbols in the corresponding signal table (analog or digital).

If you have selected an individual symbol, the next symbol is selected after you have clicked on <Add>. You can add consecutive symbols by clicking on <Add> several times.

By double clicking on the symbol, this is also adopted in the symbol table.

The symbols are re-loaded from the PLC with <Update symbols>.

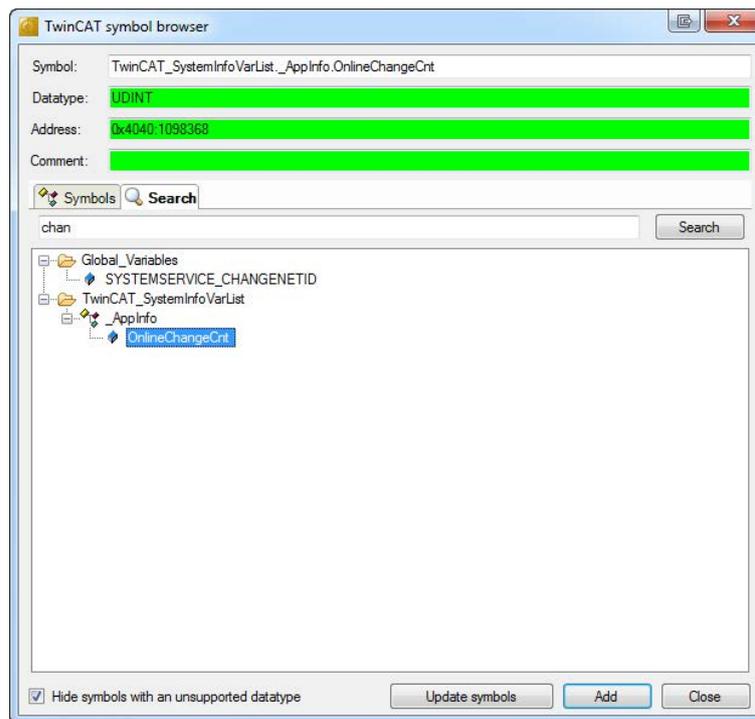


Fig. 25: Signal search in the TwinCAT symbol browser

You can search for symbols by name in the *Search* tab. The handling and selection in the search result tree is identical to the selection in the symbol tree.

Note



Text signals (data type String) are only supported by Request TwinCAT via UDP.

4.3.4 Diagnostics

You can find a list of all of the symbols recorded with the TwinCAT request module with the actual value in the *Analog* and *Digital* tabs of the TwinCAT request module.

5 Request TwinCAT via UDP

5.1 System integration with data path UDP and UDP realtime (RT)

The measurement data is transmitted via UDP to *ibaPDA*. The prerequisite in *ibaPDA* is the license for the communication interface *ibaPDA-Interface-Generic-UDP*.

You need an Ethernet connection via standard network cards.

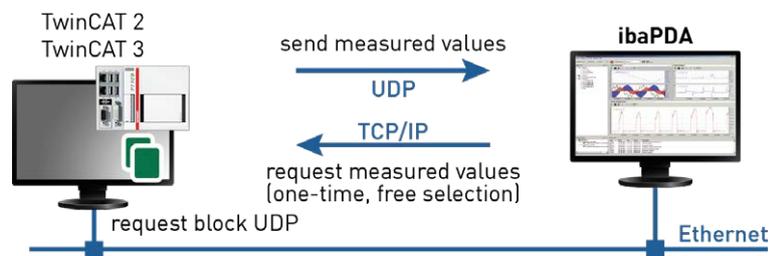


Fig. 26: Topology request via UDP

An additional prerequisite is the *TwinCATRequestLibUDP* or *TwinCATRequestLibUDPRT* library in the TwinCAT controller. In addition, it is necessary to incorporate the TCP/IP server library from TwinCAT. The TCP/IP server library requires a license and must be installed separately:

- TS6310 (UDP) for TwinCAT 2 controllers
- TF6310 (UDP) for TwinCAT 3 controllers
- TF6311 (UDP realtime) for TwinCAT 3 controllers

5.2 Configuration and engineering of the TwinCAT controller

5.2.1 TwinCAT 2

Add the *TwinCATRequestLibUDP* library from the directory `...\Vx.y.z\TwinCAT_V2` to your project. In addition, TwinCAT TCP server (TS6310) must be installed on the TwinCAT controller.

Create an instance of a management block `IBA_TCREQ_MAN` and a signal data block `IBA_TCREQ_DATA_UDP`. In addition, you must create a buffer in which the data is written. The size of the buffer depends on how much data you would like to measure via this request function block pair.

```
ibaReqMan : IBA_TCREQ_MAN;
ibaReqData : IBA_TCREQ_DATA_UDP;
DataBuffer : ARRAY[0..255] OF BYTE;
```

The management and signal data blocks can be in the same program or in separate programs.

You call them as follows:

```

ibaReqMan(
    Name:= 'DemoRequest',
    State=>
);
ibaReqData(
    Name:= 'DemoRequest',
    DataBuffer:= ADR(DataBuffer[0]),
    MaxDataSize:= SIZEOF(DataBuffer),
    Adapter:= '',
    State=> ,
    ADSError=> ,
    Size=> ,
    UsedAdapter=>);

```

You can leave the entry for "Adapter" empty and mark the output "UsedAdapter" to check whether the correct IP address is used. If the IP address is not correct, then you can set the correct IP address for "Adapter."

5.2.2 TwinCAT 3

Add either the TwinCATRequestLibUDP.COMPILED library or the TwinCATRequestLibUDPRT.COMPILED library from the directory ...\\x.y.z\\TwinCAT_V3 to your project. In addition, TF6310 or TF6311 must be installed on the TwinCAT controller.

Create an instance of a management block IBA_TCREQ_MAN and a signal data block IBA_TCREQ_DATA_UDP or IBA_TCREQ_DATA_UDPRT. In addition, you must create a buffer in which the data is written. The size of the buffer depends on how much data you would like to measure via this request function block pair.

```

ibaReqMan : IBA_TCREQ_MAN;
ibaReqData : IBA_TCREQ_DATA_UDP;
DataBuffer : ARRAY[0..255] OF BYTE;

```

When using UDP realtime replace IBA_TCREQ_DATA_UDP by IBA_TCREQ_DATA_UDPRT.

The management and signal data blocks can be in the same program or in separate programs.

You call them as follows:

```

reqManUdp(Name := 'DemoRequestUDP');
reqDataUdp(Name := 'DemoRequestUDP', DataBuffer := ADR(DataBufferUdp[0]), MaxDataSize := SIZEOF(DataBufferUdp), Adapter := '');

```

You can leave the entry for "Adapter" empty and mark the output "UsedAdapter" to check whether the correct IP address is used. If the IP address is not correct, then you can set the correct IP address for "Adapter."

5.3 Configuration in ibaPDA

The configuration is done in the I/O Manager of *ibaPDA*. First set up the connection from *ibaPDA* to the TwinCAT controller via Ethernet.

Once the connection is set up, add a TwinCAT request module accordingly. See chapter [↗ TwinCAT request module](#), page 36.

The configuration of the signals and the selection in the symbol browser is described in the chapter [↗ Selecting symbols](#), page 31.

5.3.1 Setting up the connection

The prerequisite for using UDP or UDP realtime as data path is the interface *ibaPDA-Interface-Generic-UDP*. If all system requirements are met, the "Generic UDP" interface will be displayed in the interface tree. TwinCAT request is a module of this interface.

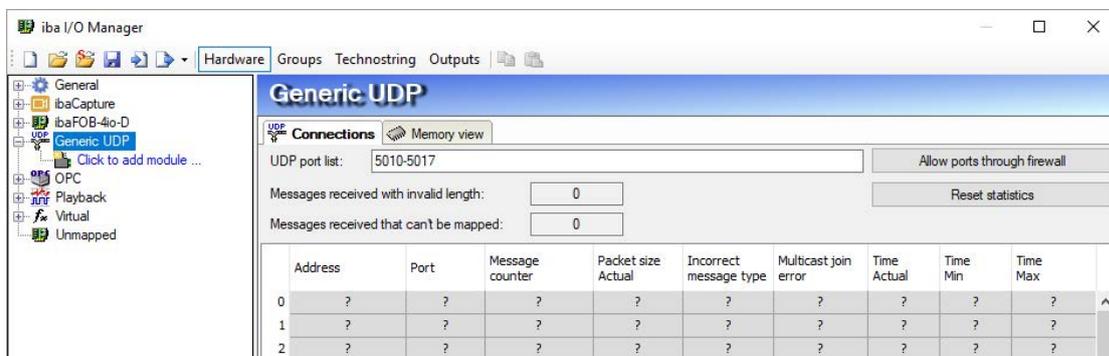


Fig. 27: "Generic UDP" interface

The interface itself has the following functions and configuration options.

UDP port list

Ports where *ibaPDA* waits for incoming UDP telegrams. You can enter the port numbers as a range or as an enumeration or both combined. Enter a range with a hyphen and separate non-consecutive numbers with commas. The range 5010-5017 is default. The port number must be identical in the controller (see *Configuring the controller* in the manual *ibaPDA-Interface-Generic-UDP*).

Allow ports through firewall

When installing *ibaPDA*, the default port numbers of the protocols used are automatically entered in the firewall. When the port number is changed or when the interface was activated subsequently, it is necessary to allow the ports in the firewall here by clicking on this button.

Counter for faulty telegrams

Connection table

Note



You can find additional information about the interface *ibaPDA-Interface-Generic-UDP* in the associated manual.

Add a TwinCAT request module by clicking below the interface. Select the module type “TwinCAT request” and click on <OK>.

5.3.2 TwinCAT request module

5.3.2.1 General Settings

The general settings to be made are identical to chapter ↗ *General Settings*, page 23.

5.3.2.2 Configuration of the control path

The configuration of the control path is identical to the configuration in chapter ↗ *Configuration of the control path*, page 24 .

5.3.2.3 UDP and UDP RT data path

The data path is configured in the *Connection* tab. You have the following setting options:



Data Path	
Maximum data size:	1024 byte
TwinCAT IP address:	192.168.80.24
ibaPDA IP address:	192.168.80.10
ibaPDA port:	5010

Fig. 28: Configuration of the UDP data path in the Connection tab

Maximum data size

Maximum number of data bytes that can be received by the PLC. The value is automatically entered when testing the connection.

TwinCAT IP address

IP address of the PLC. *ibaPDA* must know the IP address in order to correctly identify the UPD data coming from the PLC. The IP address is automatically entered when testing the connection.

ibaPDA IP address

IP address of the computer on which the *ibaPDA* service is running. The PLC requires the IP address in order to know where it should send the data. From the drop-down menu, select the IP address that can be accessed from the PLC.

ibaPDA Port

UDP port number where the PLC should send the data. There must be a port within the port range that is configured in the generic UDP interface. The first port is selected by default. If several request function blocks are configured on the same PLC, you have to assign each one a different port number. Request function blocks on different PLCs can use the same port number.

5.3.3 Selecting symbols

The symbols are selected identically to chapter [↗ Selecting symbols](#), page 31.

Note



Text signals are only supported by Request TwinCAT via UDP.

Note



For further processing of the text signals (datatype STRING) or splitting these into other text signals use a text splitter module under the *Virtual* interface.

6 Diagnostics

6.1 Checking the license

If the “TwinCAT request” modules are not shown in the signal tree, you can check in the I/O Manager under "General - Settings - License info" whether your license *ibaPDA-Request-TwinCAT* is properly detected.

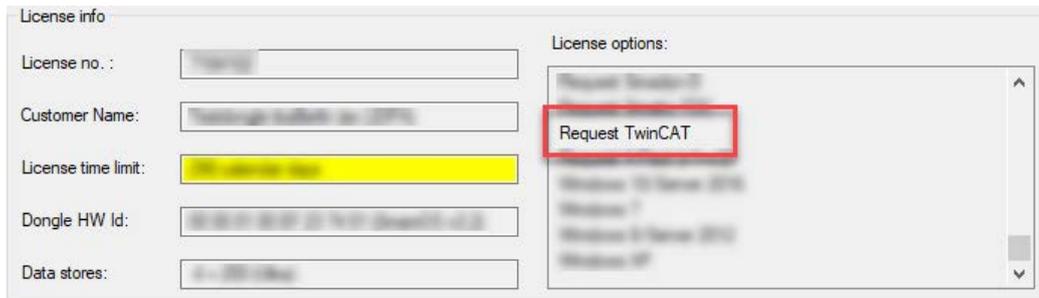


Fig. 29: Display of the license in the I/O Manager

6.2 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_yyyymmddhhmmss.txt` are archived log files.

Examples:

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

6.3 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. 30: 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.

Ping statistics for 192.168.21.121:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
C:\Users>_
```

Fig. 31: PING unsuccessful

7 Technical data

Brief description		
Name	ibaPDA-Request-TwinCAT	
Order number	31.001303	
Request interface		
	UDP and UDP RT	ibaBM-eCAT
Request blocks	64	
Data volume/block	a total of 500 signals (analog or digital), max. 2000 bytes	
Hardware restriction	-	Max. 512 analog and 512 digital signals
Transmission speed	fast	cycle precise
iba licenses		
ibaPDA base license* e.g. ibaPDA-V7-64 (30.770064)	✓	✓
ibaPDA-Request-TwinCAT* (31.001303)	✓	✓
ibaPDA-Interface-Generic-UDP* (31.001075)	✓	-
ibaBM-eCAT* (13.127000)	-	✓
TwinCAT licenses		
ibaRequest blocks	TwinCATRequestLibUDP TwinCATRequestLibUDPRT	TwinCATRequestLibCommon
TwinCAT TCP/IP Server*	TwinCAT 2: TS6310 TwinCAT 3: TF6310 or TF6311	-
TwinCAT-ADS communication	✓	✓

* with costs

8 Support and contact

Support

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

Note



If you need support for software products, please state the license number or the CodeMeter container number (WIBU dongle). For hardware products, please have the serial number of the device ready.

Contact

Headquarters

iba AG
Koenigswarterstrasse 44
90762 Fuerth
Germany

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

Mailing address

iba AG
Postbox 1828
D-90708 Fuerth, Germany

Delivery address

iba AG
Gebhardtstrasse 10
90762 Fuerth, Germany

Regional and Worldwide

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

www.iba-ag.com.