ibaPDA-Request-TwinCAT

Request data interface to TwinCAT systems

Manual

Issue 1.4

Measurement Systems for Industry and Energy

www.iba-ag.com

Manufacturer

iba AG

Koenigswarterstrasse 44

90762 Fuerth

Germany

Contacts

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

Version	Date	Revision	Author	Version SW
1.4	08-2025	Note on data type ULINT	nm	8.7.0

Contents

1 About this documentation

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

This documentation in particular addresses persons who are concerned with the configuration, test, commissioning or maintenance of Programmable Logic Controllers of the supported products. For the handling *ibaPDA-Request-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 *Logic diagram*
Calling the menu command	*Step 1 – Step 2 – Step 3 – Step x* Example: Select the menu *Logic diagram – Add – New function block*.
Keys	<Key name> Example: <Alt>; <F1>
Press the keys simultaneously	<Key name> + <Key name> Example: <Alt> + <Ctrl>
Buttons	<Key name> Example: <OK>; <Cancel>
Filenames, paths	`Filename`, `Path` Example: `Test.docx`

1.3 Used symbols

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

Danger!

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

- Observe the specified measures.

Warning!

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

- Observe the specified measures.

Caution!

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

- Observe the specified measures

Note

A note specifies special requirements or actions to be observed.

Tip

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

Other documentation

Reference to additional documentation or further reading.

2 System requirements

The following system requirements are necessary for using the *ibaPDA-Request-TwinCAT* data interface.

- *ibaPDA* v8.7.0 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
- Additional licenses if UDP 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 of 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, 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 websitehttps://www.beckhoff.com. Start the installation program and select full installation. The TwinCAT 3 ADS runtime also supports TwinCAT 2.
- Ethernet cponnection 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 with an Generic UDP interface Number of connections: 64

Hardware

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

3 About ibaPDA-Request-TwinCAT

The interface *ibaPDA-Request-TwinCAT* is suitable for the measurement data acquisition with a free symbol selection from Beckhoff-TwinCAT controllers via EtherCAT or Ethernet (UDP/IP). The measurement data is actively sent from the controller to *ibaPDA*. This requires the integration of request blocks into the TwinCAT controller. These request blocks are used 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 using a browser. This enables access to all 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. The TwinCAT TCP/IP server library must be integrated for communication via UDP. 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. The TwinCAT TCP/IP server library must be integrated for communication via UDP realtime. 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_ECAT, 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 "iba Software & Manuals" data medium at `\04_Libraries_and_Examples\10_Libraries\04_TwinCAT\ibaTwinCAT-Lib_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)

3.1 Request blocks

The request blocks are used to initialize and control 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 part 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 can 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	Validation of the variables received from 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 correspond-ing IBA_TCREQ_MAN function block.

Name	Type	In/Out	Description
DataBuffer	POINTER TO BYTE	IN	Pointer to the data buffer into which the values of the requested variables are to be written This data buffer must be linked with EtherCAT output variables.
MaxDataSize	UINT	IN	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.
DataBuffer	POINTER TO BYTE	IN	Pointer to the data buffer in which the values of the requested variables are written
MaxDataSize	UINT	IN	Size of the data buffer
Adapter	STRING(15)	IN	The IP address of the network adapter via which data is to be sent to *ibaPDA* If this is empty, the standard 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 via which the data is sent to ibaPDA
SendCounter	UINT	OUT	Counter that is incremented each time a message is sent to ibaPDA

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

Status	Description
TCREQ_DATA_INIT	Initial state: Search for the module of the same name IBA_TCREQ_MAN.
TCREQ_DATA_NO_PATH	Connected to the management module, but no data path available.
TCREQ_DATA_READY	Connected to the management module and data path found. The variable list is empty.
TCREQ_DATA_COPYING	Copy data for the variable list.
TCREQ_DATA_ONLINECHANGE	An online change has been made; wait for the management module to respond.

4 Request-TwinCAT via EtherCAT

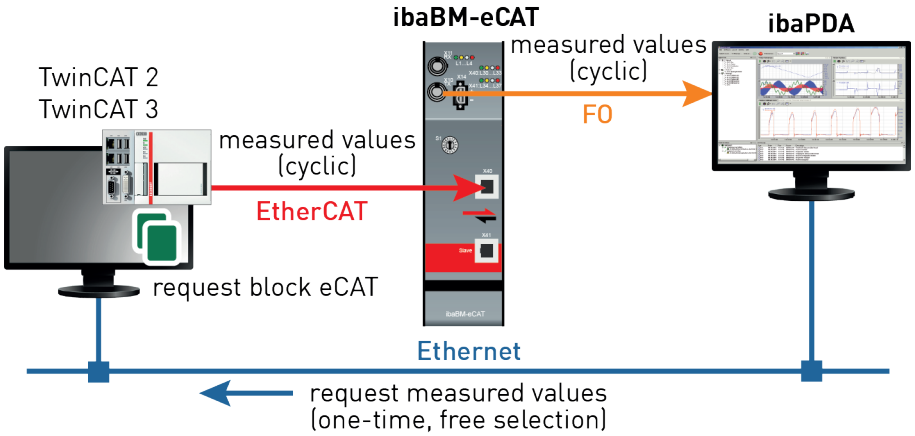
4.1 System integration via data path EtherCAT

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

The following figure shows the topology for the request via EtherCAT:



An additional prerequisite is the ibaTwinCATRequest library in the TwinCAT controller.

4.2 Configuration and engineering of the the TwinCAT controller

On the TwinCAT side, the following configuration and engineering steps must be carried out in TwinCAT:

Hardware engineering
Integration of the *ibaBM-eCAT* device in the device configuration. A description of this can be found in the ibaBM-eCAT documentation in the chapters *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 into the TwinCAT program is described in the following chapters.

4.2.1 TwinCAT 2

Proceed as follows to integrate the request blocks into the TwinCAT program.

1. Open the library manager in your project and add the library *ibaTwinCATRequest.lib* from the directory `...\Vx.y.z\TwinCAT_V2`.
2. Now add an instance of the management module IBA_TCREQ_MAN and an instance of the signal data module IBA_TCREQ_DATA_ECAT.

   You can add the instances to the same program or to different programs.
3. Create a buffer that is linked to EtherCAT output variables.

   The syntax AT %Q* identifies 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 want to measure via this request function block pair.

   iba AG recommends creating 2 pairs of modules in order to utilize the maximum amount of data.

Example: Calling the modules
For the sake of simplicity, both blocks are called in one task in this example.

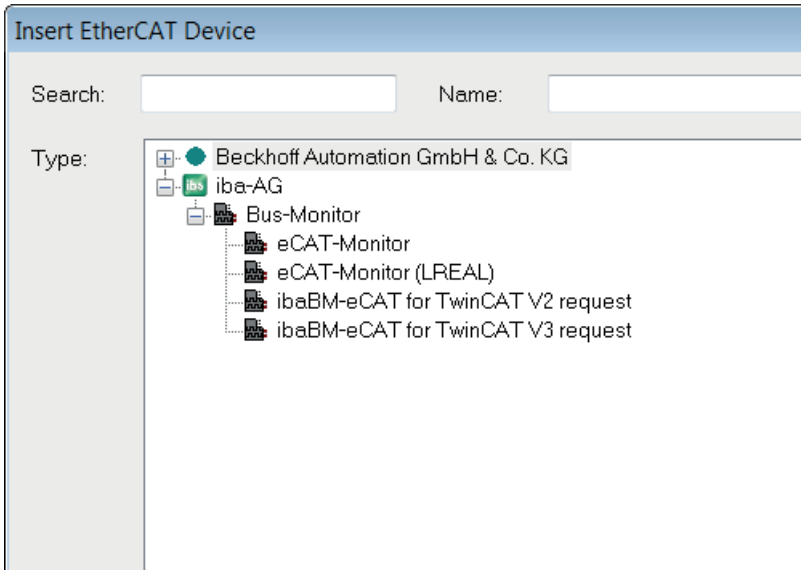
```
PROGRAM ibaRequest
VAR

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

    ibaReqMan_1 : IBA_TCREQ_MAN;
    ibaReqData_1 : IBA_TCREQ_DATA_ECAT;

    ibaReqMan_2 : IBA_TCREQ_MAN;
    ibaReqData_2 : IBA_TCREQ_DATA_ECAT;

END_VAR
```

You can call them as follows:

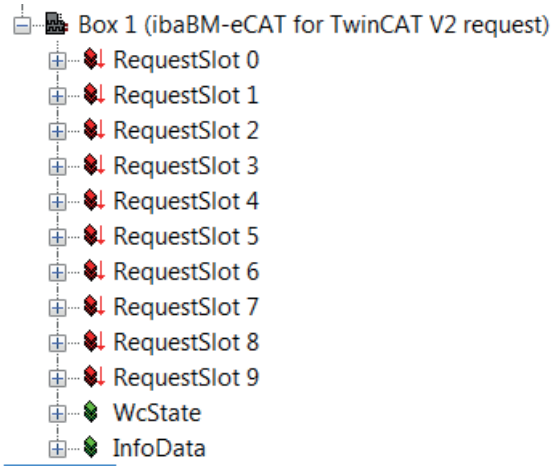
```
ibaReqMan_1(
    Name:= 'RequestData_1',
    State=> );

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=> );
```

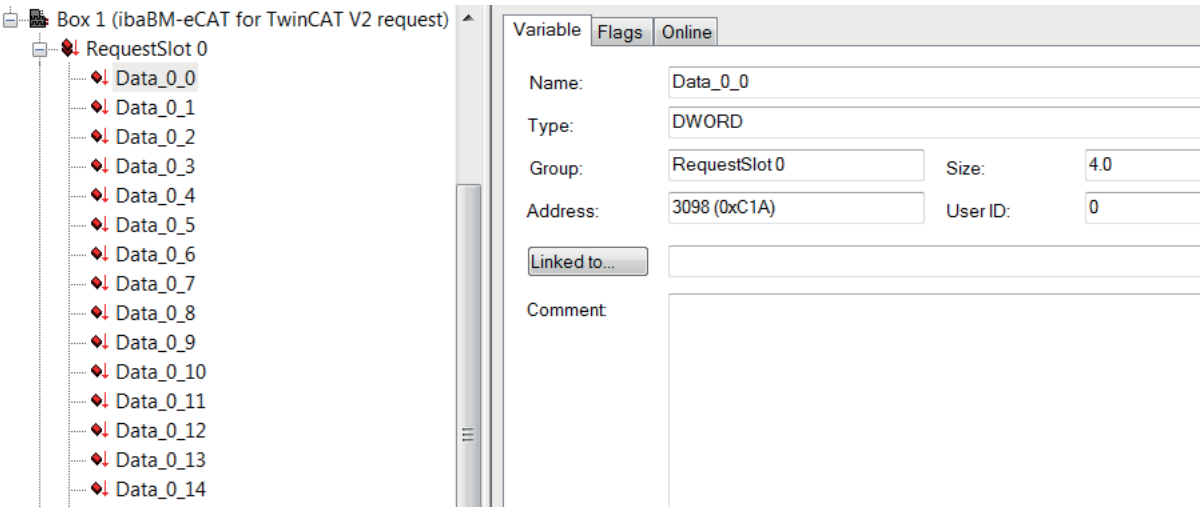
4. Now link the DataBuffer variable with the EtherCAT variables of *ibaBM-eCAT*.

   To do this, open the System Manager and add the i*baBM-eCAT for TwinCAT V2 request* device to your EtherCAT network.



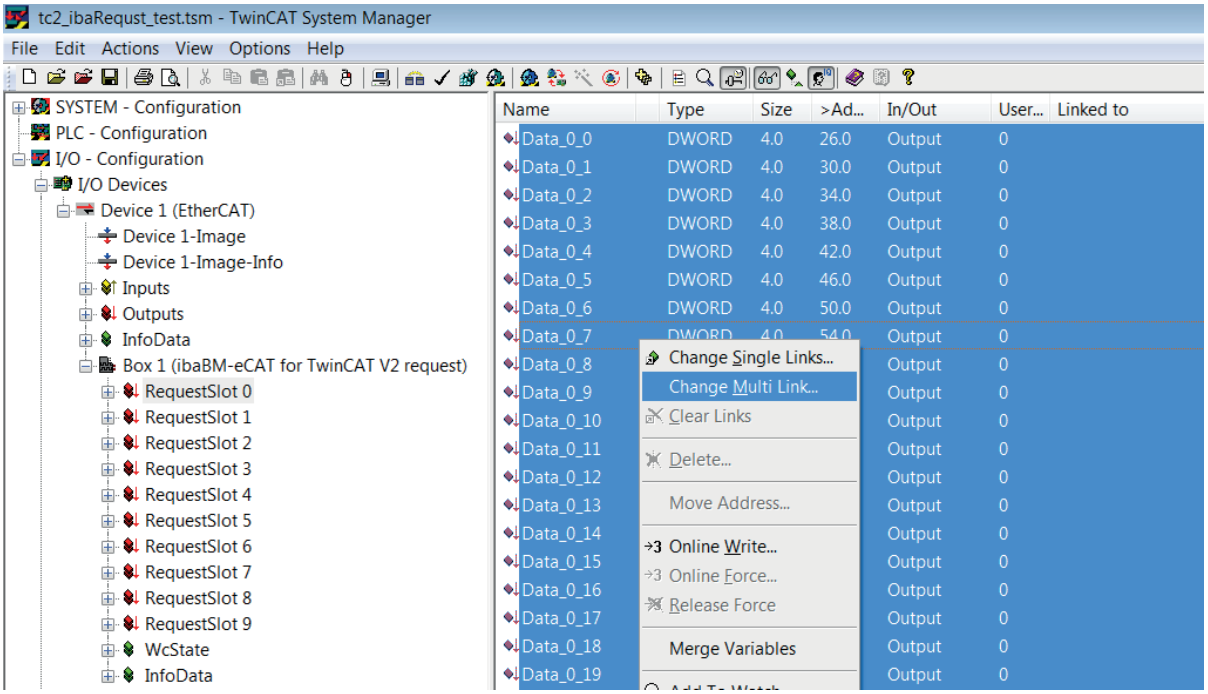
   ibaBM-eCAT has 10 request slots. Each request slot consists of 64 DWORD variables, which correspond to 256 bytes. The DWORD variables can be linked to arrays. To reduce the number of variables, DWORD variables are used instead of BYTE variables.





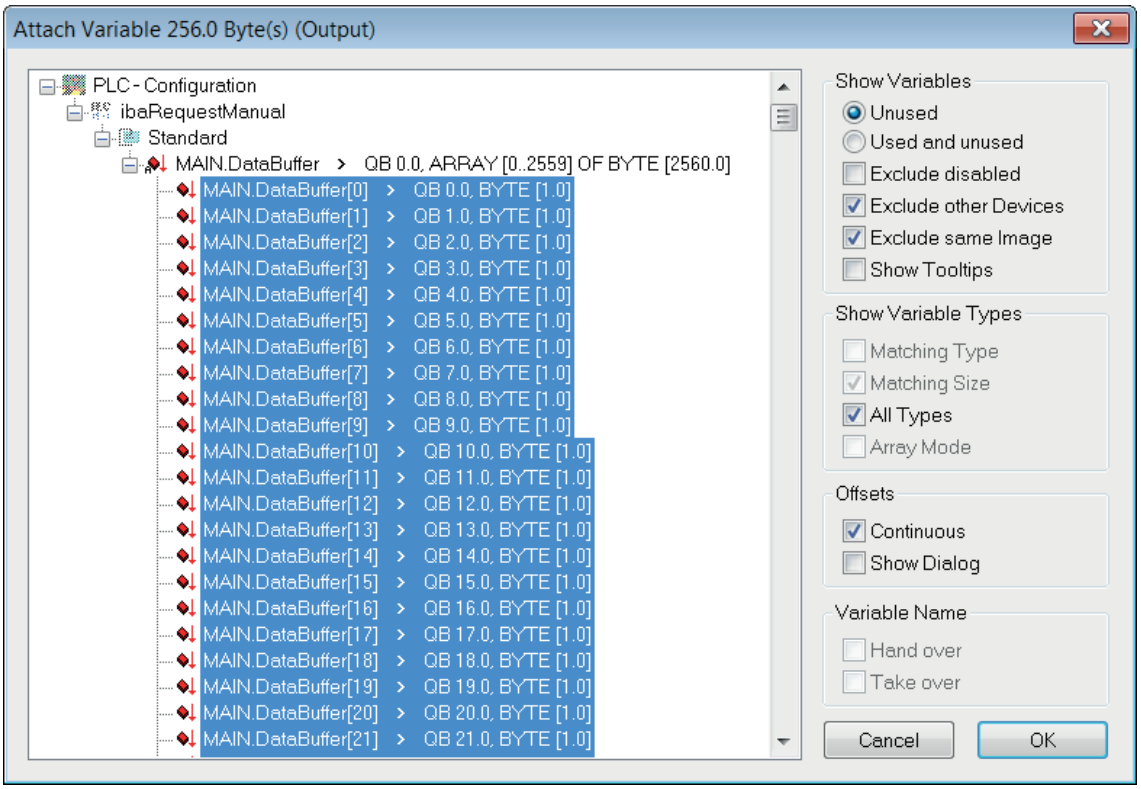
5. If you select the *ibaBM-eCAT* device in the tree, a table with all variables is displayed on the right-hand side. By holding down the Shift key, you can select several variables to be linked to the data buffer.
6. Select *Change Multi Link…* from the context menu.

   The multi-link command is only active if you select variables from the same EtherCAT telegram. By default, the first 5 slots are sent in the first EtherCAT telegram and the last 5 slots in a second EtherCAT telegram.

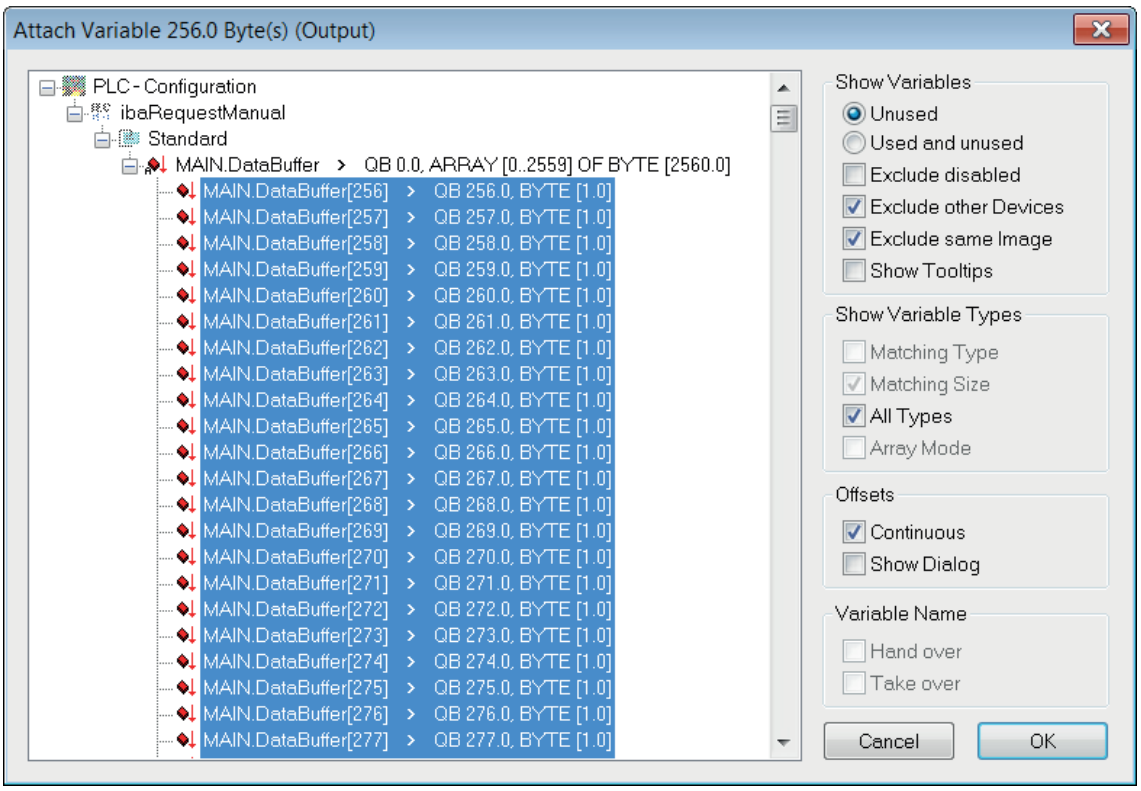


→ The dialog "Attach variable" is opened.

7. 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.



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.



If you link the variables in another way, please do not attach more than 1024 bytes at once.

8. Then enable the configuration and download your program.

4.2.2 TwinCAT 3

Proceed as follows to integrate the request blocks into the TwinCAT program.

1. Add the *TwinCATRequestLibCommon.compiled-library* from the directory `...\Vx.y.z\TwinCAT_V3` to your project.
2. Now add an instance of the management module IBA_TCREQ_MAN and an instance of the signal data module IBA_TCREQ_DATA_ECAT.
3. Also create a buffer that is linked to EtherCAT output variables.

   The syntax AT %Q* identifies 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 want to measure via this request function block pair.

   iba AG recommends creating 2 pairs of modules in order to utilize the maximum amount of data.

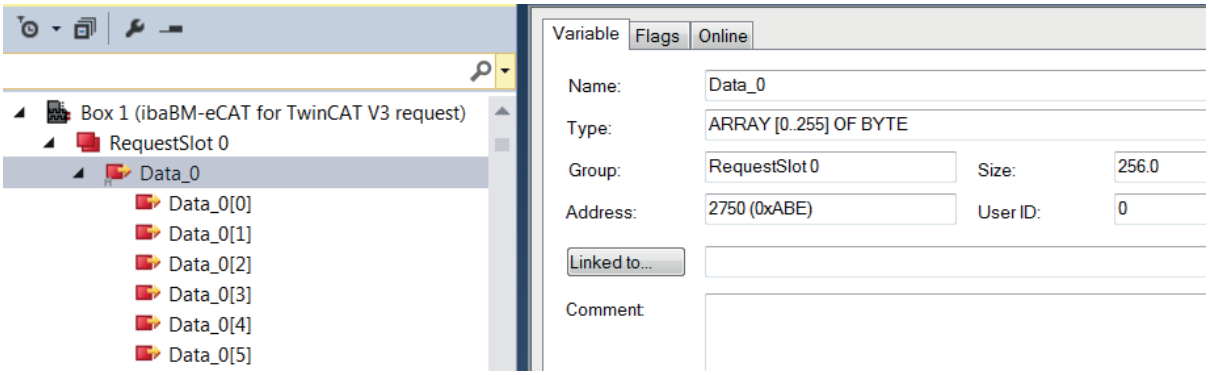
```
reqManEcat : IBA_TCREQ_MAN;
reqDataEcat : IBA_TCREQ_DATA_ECAT;
DataBufferEcat AT %Q* : ARRAY[0..2559] OF BYTE;
```

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

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

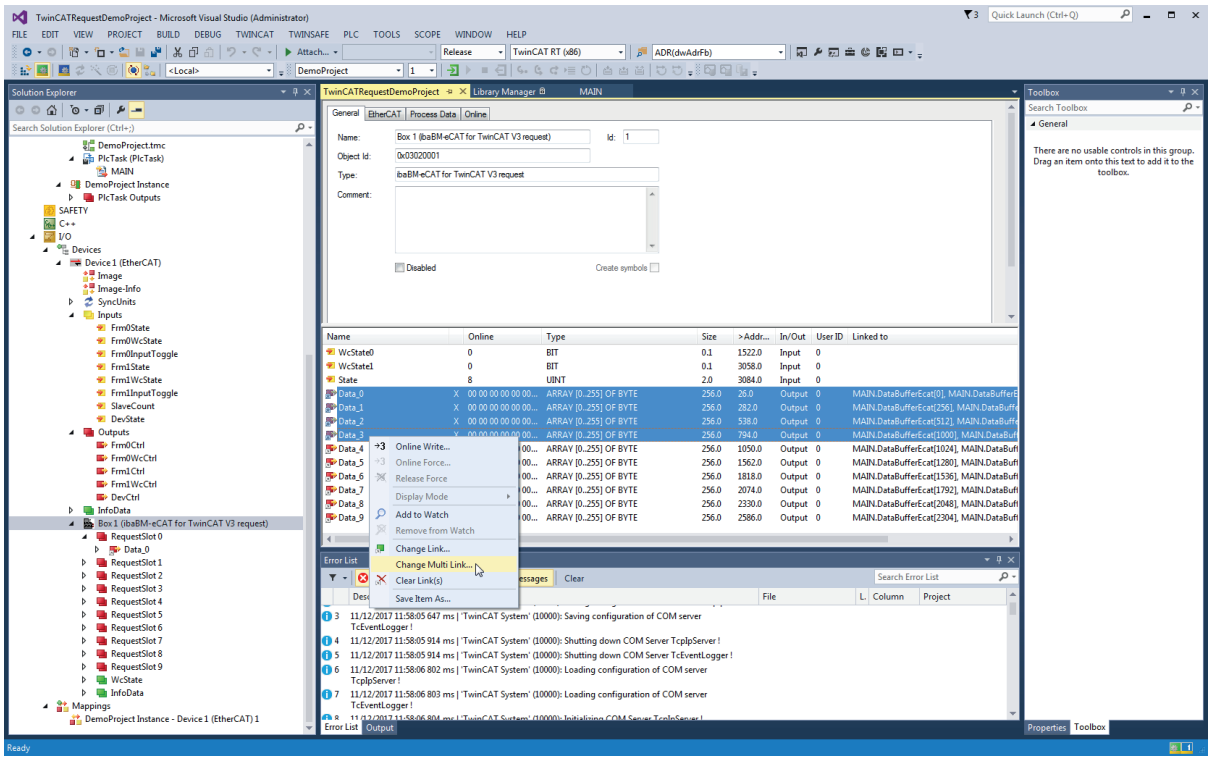
4. Add an *ibaBM-eCAT* device to your EtherCAT network in the I/O section. First copy the ESI file of the *ibaBM-eCAT* device to the subdirectory `Config\Io` of your TwinCAT installation directory.
5. Then select *ibaBM-eCAT for TwinCAT V3 request* as TwinCAT 3 request device and add it.

   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_ECAT.

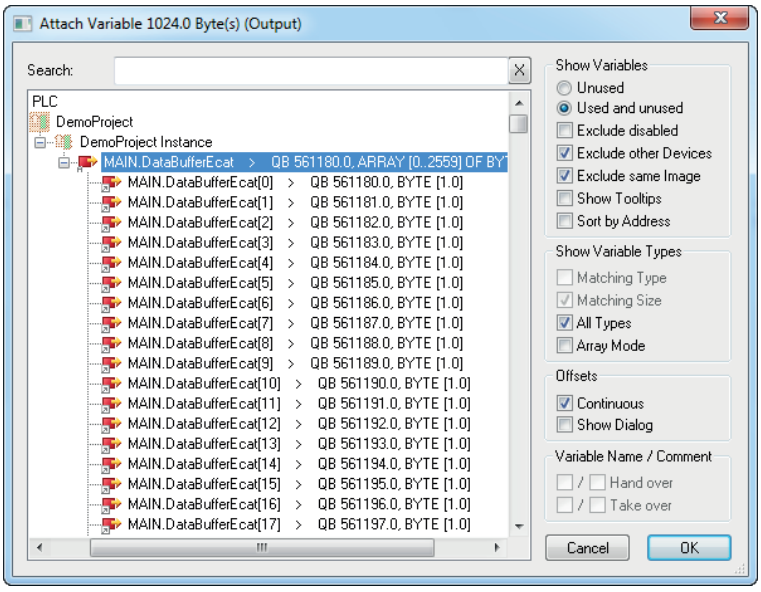


6. To link the data buffer to the request slots, select the device in the tree and then select the output variables in the table on the right.
7. Select *Change multi link* from the context menu.

The multi-link command is only active if you select variables from the same EtherCAT telegram. By default, the first 5 slots are sent in the first EtherCAT telegram and the last 5 slots in a second EtherCAT telegram.



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



9. Activate the configuration and restart the TwinCAT system to apply the I/O changes.

4.3 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 *ibaBM-eCAT*.

Once the connection has been set up, add a TwinCAT request module accordingly, see chapter ↗ *TwinCAT request module*, page 35.

The configuration of the signals and selection in the symbol browser is described in chapter ↗ *Selecting symbols*, page 29.

4.3.1 Setting up the connection

The bus monitor *ibaBM-eCAT* and the installation of an *ibaFOB-D* card in the *ibaPDA* computer are prerequisites for using EtherCAT as a data path. Once the *ibaFOB-D* card has been successfully installed, it is displayed in the interface tree.

Other documentation

Further information on the *ibaFOB-D* card can be found in the associated device documentation.

To use the TwinCAT request function, *ibaBM-eCAT* must be an active part of the EtherCAT network. The "EtherCAT SubDevice" option must therefore be activated in *ibaPDA* and the device integrated in the EtherCAT configurator.

To do this, the corresponding ESI file, an XML file, is imported into the EtherCAT configurator from the "iba Software & Manuals" data medium supplied. The signals are then configured and linked and exported as an ENI configuration file. The configuration file can then be loaded into *ibaPDA*. For further information, see chapter ↗ *Configuration and engineering of the the TwinCAT controller*, page 14.

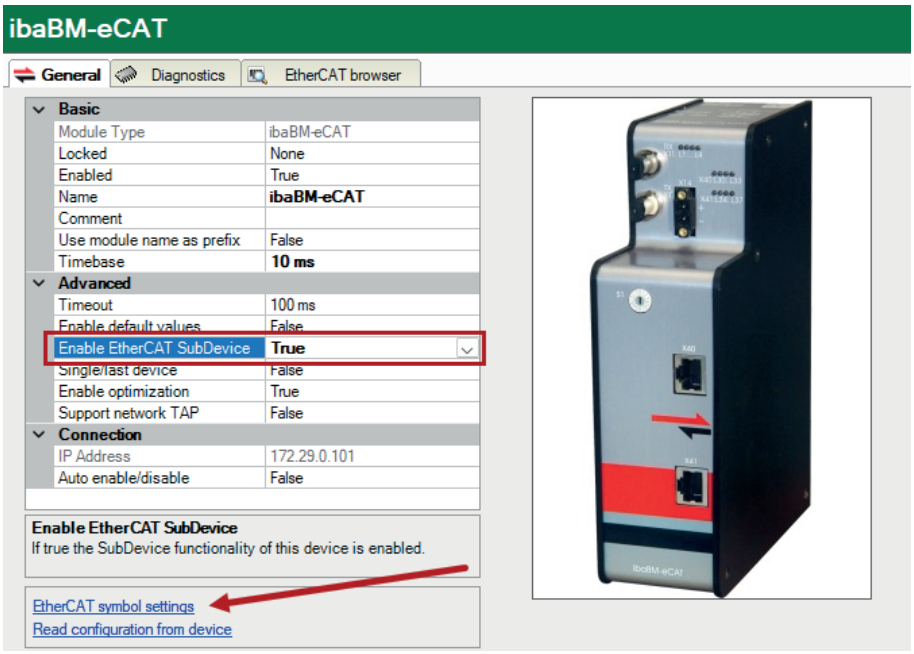
Configuring ibaBM-eCAT

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. Click on the link with the right mouse button and select *Autom. detection*.

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

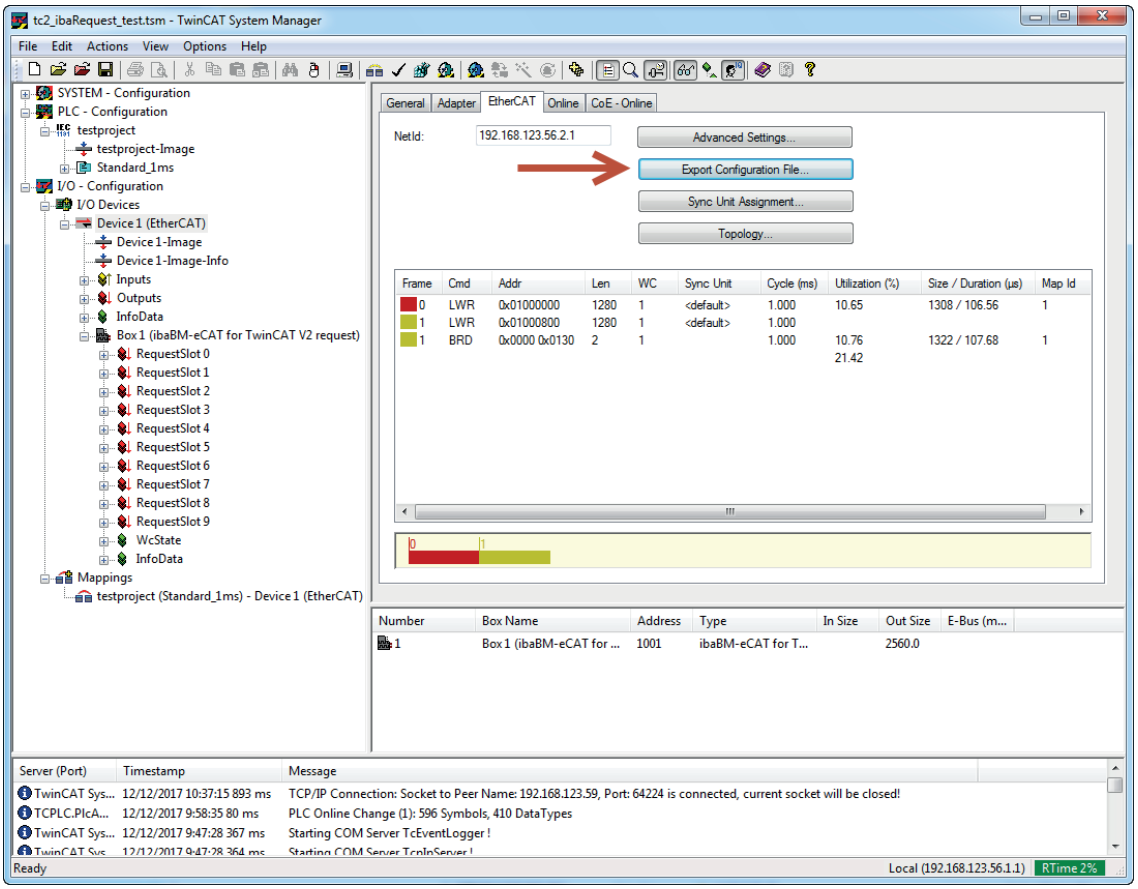
3. Open the *General* tab of the *ibaBM-eCAT* device module in the I/O Manager and select the "True" option in the *Enable EtherCAT SubDevice* field.
4. Add a TwinCAT request module to the *ibaBM-eCAT* device module by clicking below the device module (or several if you require further connections to other PLCs).

The following figure shows the *General* tab of the *ibaBM-eCAT* device module with the required setting as "EtherCAT SubDevice". In the *General* tab, you can also load the ENI configuration file for the EtherCAT network by clicking on the *EtherCAT symbol settings* link.



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

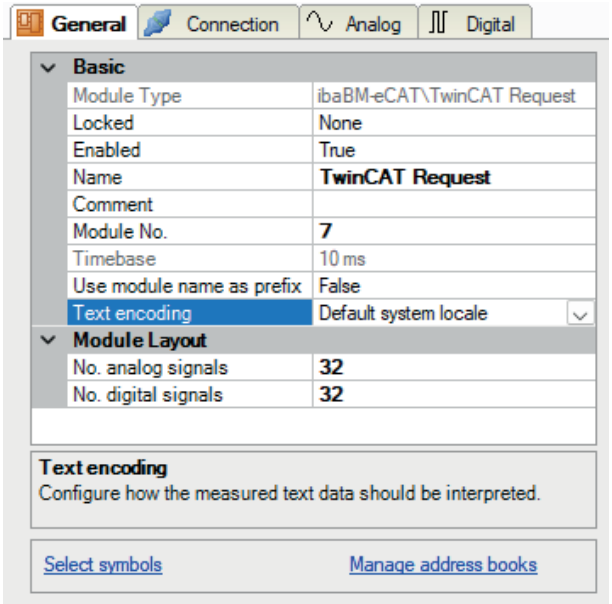
Example System Manager in TwiCAT 2



4.3.2 TwinCAT request module

4.3.2.1 General module settings

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



Basic settings

Module Type (information only)
Indicates the type of the current module.

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

Enabled
Enable the module to record signals.

Name
You can enter a name for the module here.

Comment
You can enter a comment or description of the module here. This will be displayed as a tooltip in the signal tree.

Module No.
This internal reference number of the module determines the order of the modules in the signal tree of *ibaPDA* client and *ibaAnalyzer*.

Timebase
All signals of the module are sampled on this timebase.

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

Module Layout

No. analog signals
Definition of the number of analog signals for this module (max. 500)

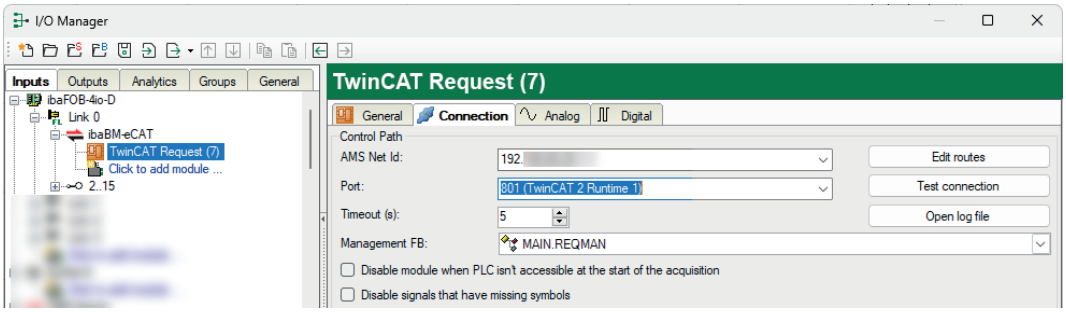
No. digital signals
Definition of the number of digital signals for this module (max. 500)

4.3.2.2 Configuration of the control path

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

Prerequisite: *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, see chapter ↗ *System requirements*, page 8.

Configuration



You have the following setting and configuration options in the *Control Path* area:

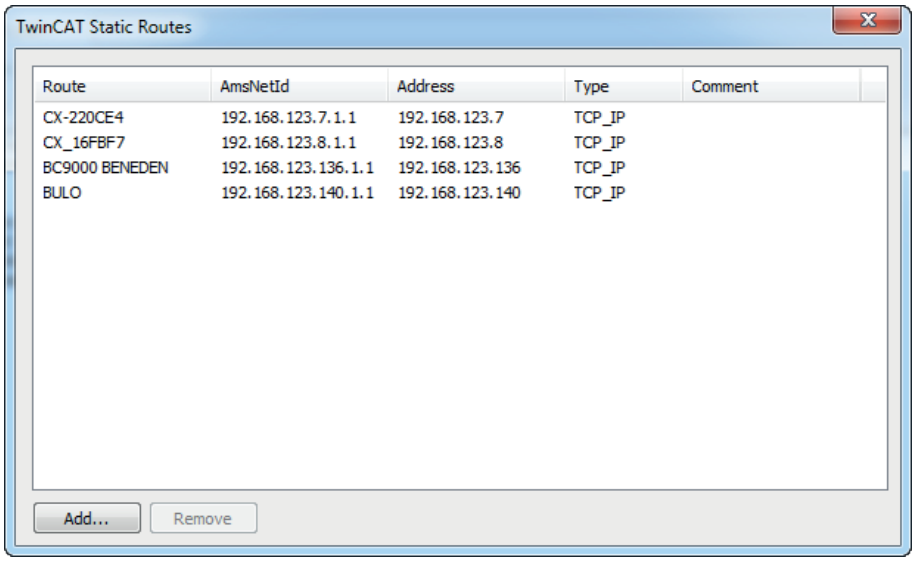
AMS Net-ID
The address of the control consists of the AMS Net-ID and a port number. The AMS Net-ID is the address of an ADS router and consists of 6 bytes. Normally the first 4 bytes are the IP address and the last 2 bytes are each 1.

Port
The port number determines the ADS device that is connected to the router. There are predefined port numbers for the TwinCAT runtimes:

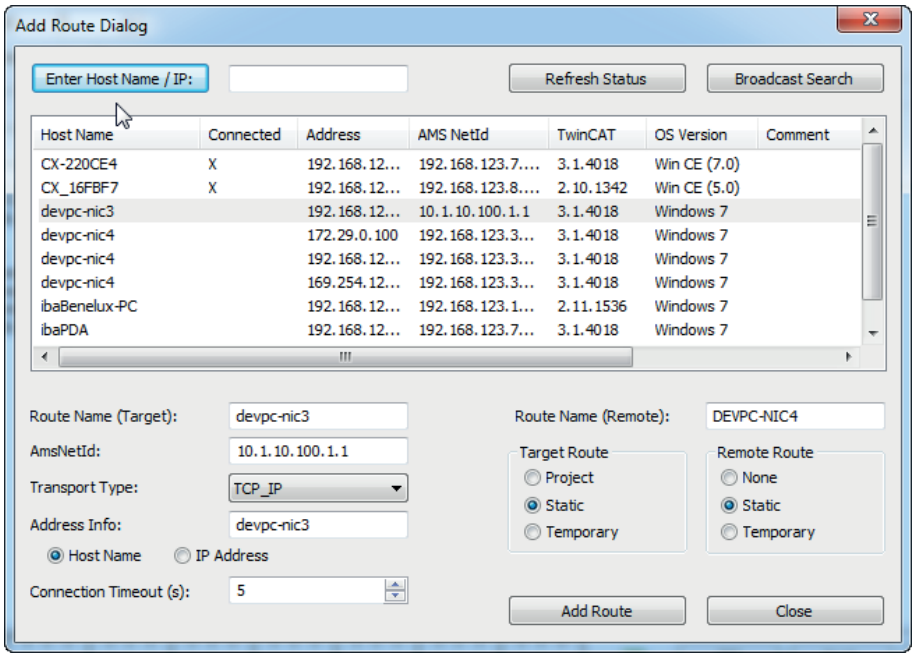
- 801, 811, 821, 831: for the 4 possible TwinCAT 2 runtimes
- 851, 852, 853, 854: for the 4 possible TwinCAT 3 runtimes

<Edit routes>
An ADS router is also installed on the *ibaPDA* server computer when the TwinCAT ADS library is installed. An entry for the remote control must be made in the routing table of this ADS router. In addition, an entry for the AMS Net-ID of the *ibaPDA* server computer must be made in the routing table of the ADS router. This can be done using the ADS router configuration tool.

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

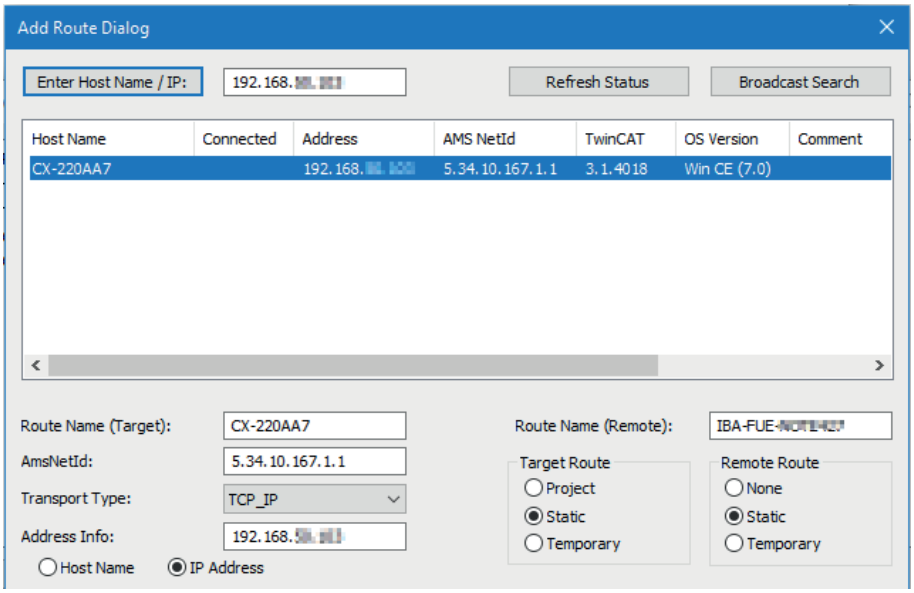


You can add a new entry with <Add>:



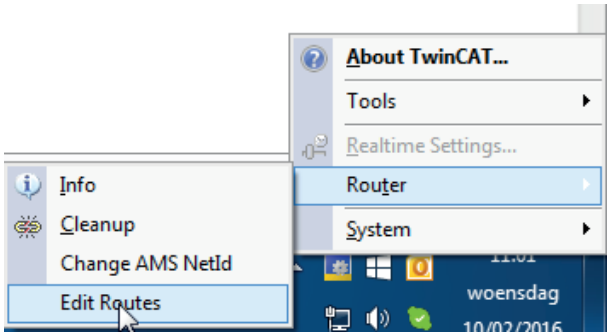
You can use the <Broadcast Search> button to search for compatible TwinCAT controllers in the local network. Either select a controller found or enter the route name, the AmsNetId and the address information manually.

In addition to the broadcast search, it is also possible to configure a connection directly by entering the IP address:

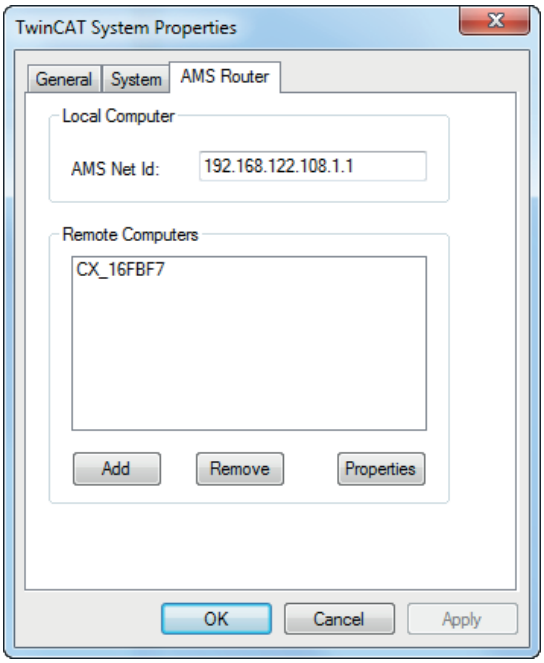


Select "Static" for the target route. For the remote route, select "Static" for the connection to a TwinCAT controller and "None" for the connection to a bus controller. Click on <Add Route> to enter the route in the routing table. The router attempts to connect to the remote ADS router and may require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

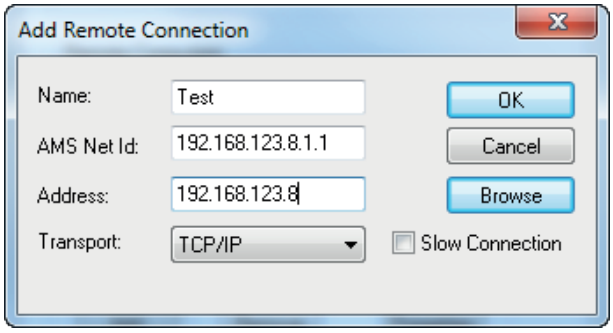
You can also open the configuration tool via the context menu on the TwinCAT taskbar icon. Select *Router - Edit Routes*.



If TwinCAT 2 is installed, the following dialog appears:

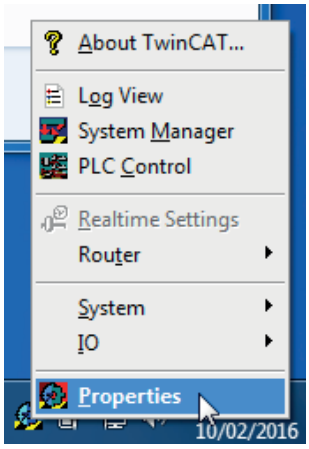


You can add a new route with <Add>:

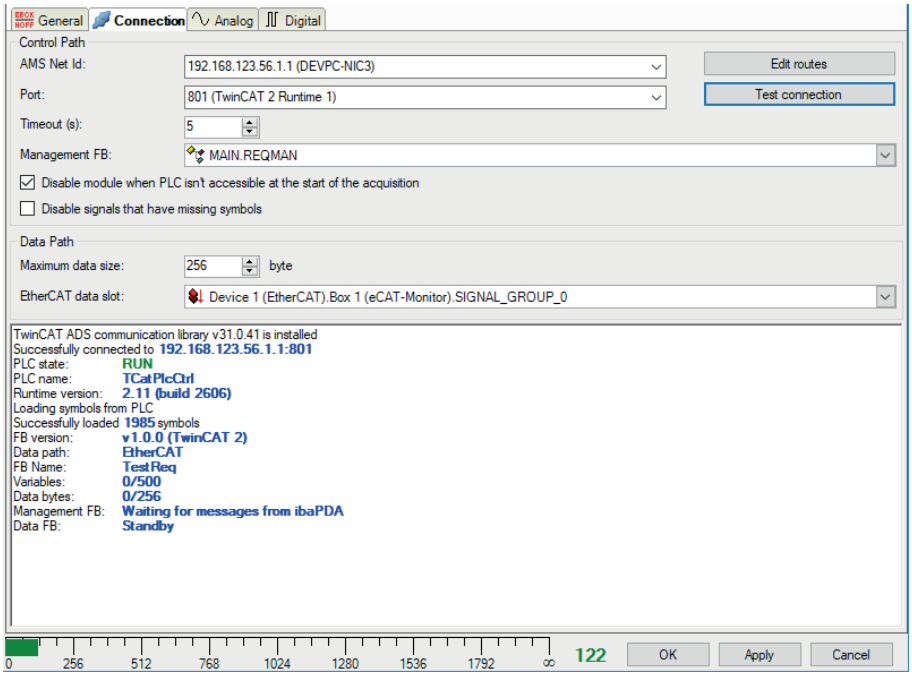


The fields *Name*, *AMS Net ID*, *Adress* must be filled out manually. Press <OK> to enter the route in the routing table.

You can also open the configuration tool via the context menu on the TwinCAT taskbar icon. Select *Properties*.



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



<Test connection>
Connection test and output of available diagnostic data. If the connection is successful, *ibaPDA* loads the symbols from the control unit.

Timeout
Timeout of the connection

Management function block
In a normal case, select the next unused management function block from the address book here. 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.

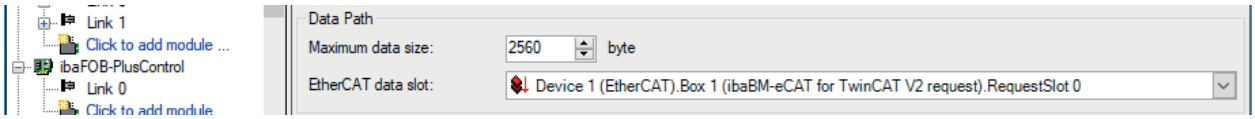
<Disable module when PLC isn't accessible at the start of the acquisition>
If this option is activated, the recording is started even if no connection to the PLC can be established. The module is disabled. During the measurement, *ibaPDA* attempts to reconnect to the PLC. If successful, the acquisition is restarted.

If this option is not activated, the recording is not started 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:



Maximum data size
You can set the maximum data size here.

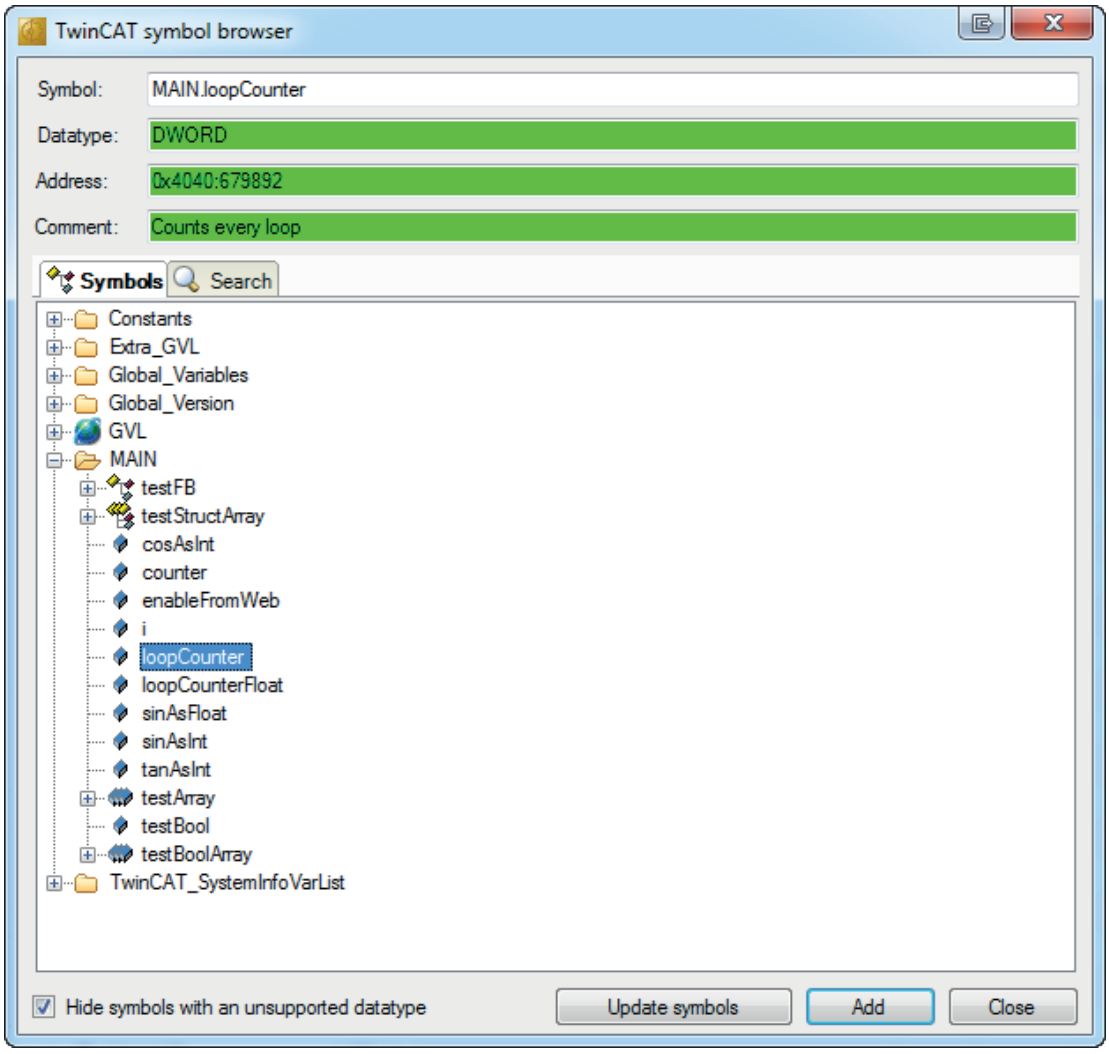
EtherCAT data slot
All of the *ibaBM-eCAT* devices in the network are listed in the drop-down menu. Select a request slot.

Select the variables to be measured. See chapter ↗ *Selecting symbols*, page 29.

4.3.3 Selecting symbols

Once the connection to the PLC has been successfully established, the symbols are loaded and you can select them in the symbol browser.

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

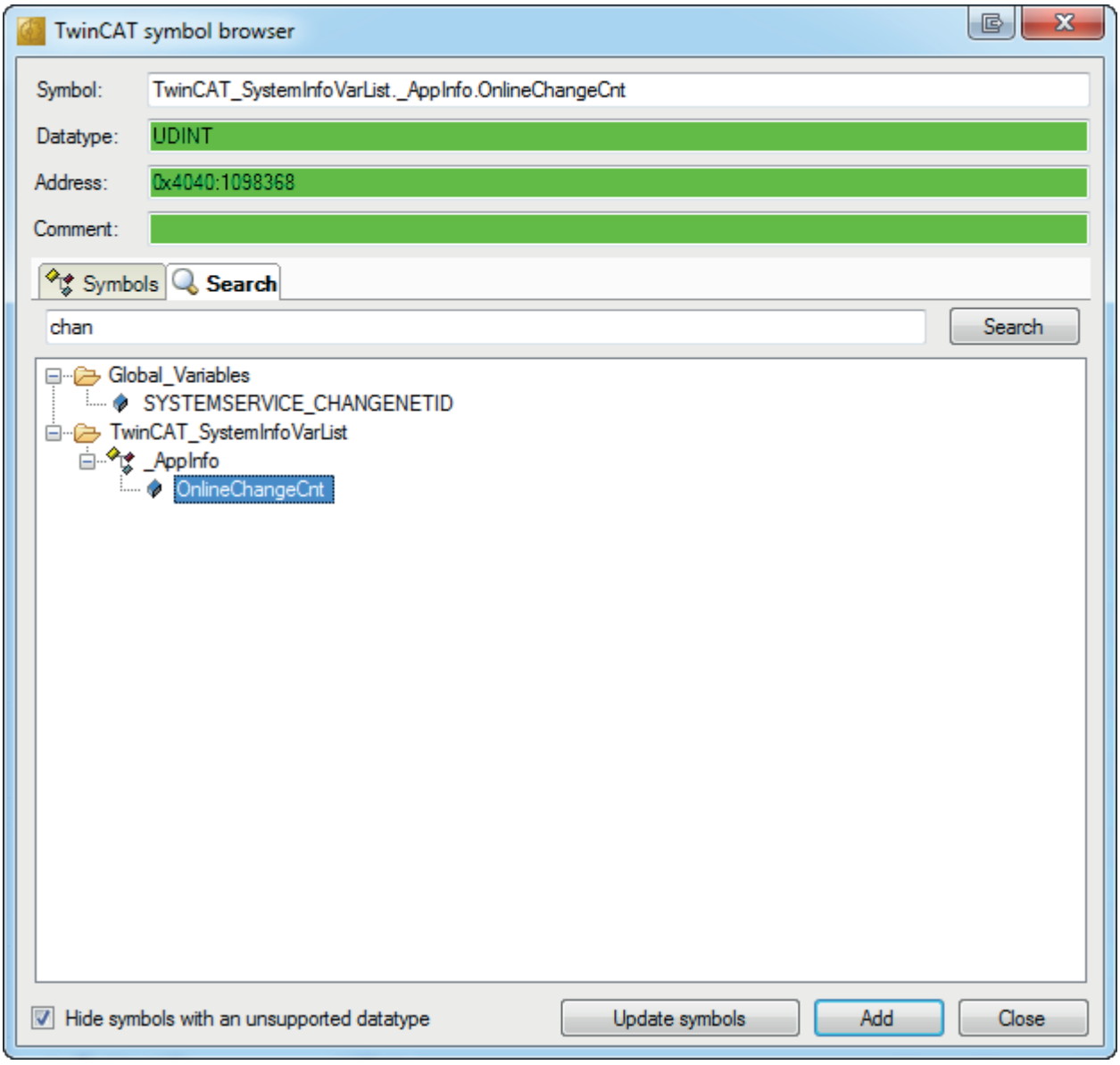


In the *Symbols* tab, you can select individual or multiple symbols in the tree. Clicking on <Add> inserts the symbols in the corresponding signal tree (analog or digital).

If you have selected a single symbol, the next symbol will be selected after you have clicked on <Add>. This allows you to add consecutive symbols by clicking <Add> several times.

Double-click on the symbol to transfer it to the signal table.

The symbols are reloaded from the PLC with <Update symbols>.



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

Note

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

Note

Values of the data type ULINT are converted by *ibaPDA* into the data type DOUBLE.

With large ULINT values, there may be a loss of resolution as these cannot be displayed exactly in the DOUBLE data type and are rounded.

4.3.4 Diagnostics

A list of all symbols recorded with the TwinCAT request module with current values can be found in the *Analog* and *Digital* registers of the TwinCAT request module.

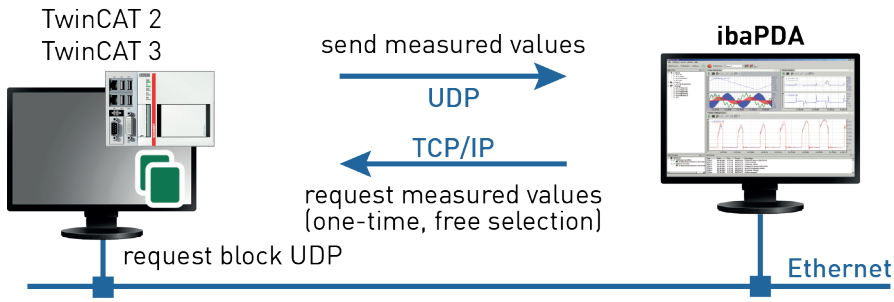
5 Request TwinCAT via UDP

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

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

You need an Ethernet connection via standard network cards.

The following image shows the topology.



An additional prerequisite is the TwinCATRequestLibUDP or TwinCATRequestLibUDPRT library in the TwinCAT controller. The integration of the TwinCAT TCP/IP server library is also required. 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 the TwinCAT controller

5.2.1 TwinCAT 2

Proceed as follows to carry out the configuration via UDP with a TwinCAT 2 device.

1. Add the TwinCATRequestLibUDP library from the directory `...\Vx.y.z\TwinCAT_V2` to your project.

   Also make sure that TwinCAT TCP Server (TS6310) is installed on the TwinCAT controller.

2. Create an instance of a management module IBA_TCREQ_MAN and a signal data module IBA_TCREQ_DATA_UDP.

3. Also create a buffer into which the data is written. The size of the buffer depends on how much data you want 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 can call them as follows:

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

4. To check whether the correct IP address is being used, you can leave the "Adapter" entry empty and select the "UsedAdapter" output.

   If the IP address is not correct, you can set the correct IP address under "Adapter".

5.2.2 TwinCAT 3

Proceed as follows to carry out the configuration via UDP with a TwinCAT 3 device.

1. Add either the TwinCATRequestLibUDP.COMPILED library or the TwinCATRequestLibUDPRT.COMPILED from the directory `...\Vx.y.z\ TwinCAT_V3` to your project.

   Also make sure that TF6310 or TF6311 is installed on the TwinCAT controller.

2. Create an instance of a management module IBA_TCREQ_MAN and a signal data block IBA_TCREQ_DATA_UDP or IBA_TCREQ_DATA_UDPRT.

3. Also create a buffer into which the data is written. The size of the buffer depends on how much data you want to measure via this request function block pair.

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

   Replace IBA_TCREQ_DATA_UDP with IBA_TCREQ_DATA_UDPRT in the case of UDP realtime use.

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

   You can call them as follows:

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

4. To check whether the correct IP address is being used, you can leave the "Adapter" entry empty and select the "UsedAdapter" output.

   If the IP address is not correct, you can set the correct IP address under "Adapter".

5.3 Configuration in ibaPDA

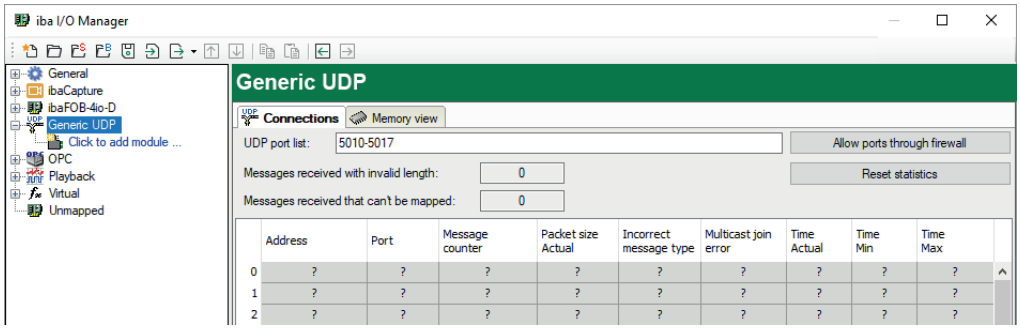
The configuration is done in the I/O Manager of *ibaPDA*. First set up the connection from *ibaPDA* to the TwinCAT controller via Ethernet.

Once the connection is set up, add a TwinCAT request module accordingly. See chapter ↗ *TwinCAT request module*, page 35.

The configuration of the signals and the selection in the symbol browser is described in the chapter ↗ *Selecting symbols*, page 29.

5.3.1 Setting up the connection

A prerequisite for the use of UDP or UDP realtime as a data path is the interface *ibaPDA-Interface-Generic-UDP*. If all system requirements are met (see, page), the interface *Generic UDP* is displayed in the signal tree. TwinCAT-Request is a module of this interface.



The interface provides the following functions and configuration options:

UDP Port-Liste
Ports on which *ibaPDA* waits for incoming UDP telegrams. You can enter the ports as port range, as list of individual ports or as a combination of both. If you enter a range, please separate it by a hyphen. Do not separate successive port numbers by commas. The default setting is the range 5010-5017. The port number must be used identically in the controller.

Other documentation

Information on configuring the controller can be found in the documentation for *ibaPDA-Interface-Generic-UDP*.

Allow port through firewall
When installing *ibaPDA*, the standard port numbers of the protocols used are automatically entered in the firewall. If the port number is changed or if the interface was subsequently enabled, this port has to be enabled in the firewall here by clicking on this button.

Sent message counter
The number of faulty telegrams and telegrams that cannot be mapped is displayed.

Connection table

Other documentation

Further information on the interface *ibaPDA-Interface-Generic-UDP* and the connection table can be found in the corresponding documentation.

Add a TwinCAT request module by clicking below the interface. Select the *TwinCAT Request* module type and click <OK>.

5.3.2 TwinCAT request module

You can make the general module settings as described in the chapter ↗ *TwinCAT request module*, page 23.

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

For the configuration of the UDP and UDP RT data path, see ↗ *UDP and UDP RT data path*, page 35.

5.3.2.1 UDP and UDP RT data path

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

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

TwinCAT IP address
IP address of the PLC.

ibaPDA must know the IP address in order to correctly identify the UDP data coming from the PLC. The IP address is entered automatically when the connection is tested.

ibaPDA IP address
IP address of the computer on which the *ibaPDA* service is running. The PLC needs the IP address to know where to send the data. Select the IP address that can be accessed from the PLC from the selection list.

ibaPDA port
UDP port number to which 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 selection of symbols is identical to chapter ↗ *Selecting symbols*, page 29.

Note

Text signals are only supported by Request TwinCAT via UDP.

Note

Values of the data type ULINT are converted by ibaPDA into the data type DOUBLE.

With large ULINT values, there may be a loss of resolution as these cannot be displayed exactly in the DOUBLE data type and are rounded.

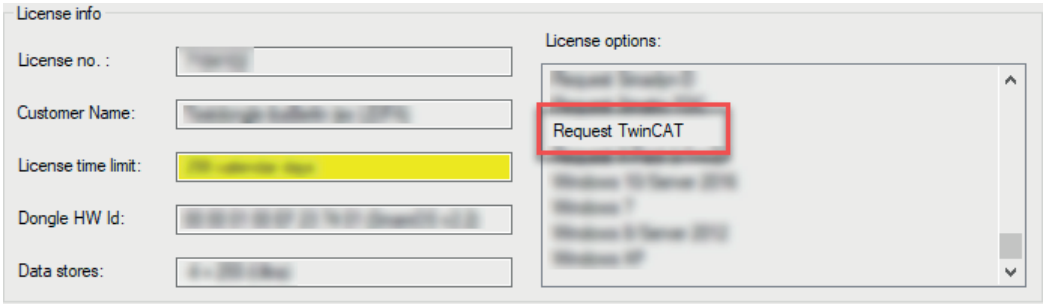
Note

The module *TDC TCP/UDP Generic* supports the acquisition and processing of strings as text signals. Therefore, you can select the data type STRING[32] in the *Analog* tab. In order to convert a text signal or to split it up into several text signals use the *text splitter* module under the *Virtual* interface.

6 Diagnostics

6.1 Checking the license

If the "TwinCAT request" modules are not displayed in the signal tree, you can check in I/O Manager under *General settings* - *License info* whether your *ibaPDA-Request-TwinCAT* license is recognized correctly.



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.

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

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

In the file system on the hard drive, you can find the log files of the *ibaPDA* server (`...\ProgramData\iba\ibaPDA\Log`). The file names of the log files include the name or abbreviation of the interface type.

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

Examples:

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

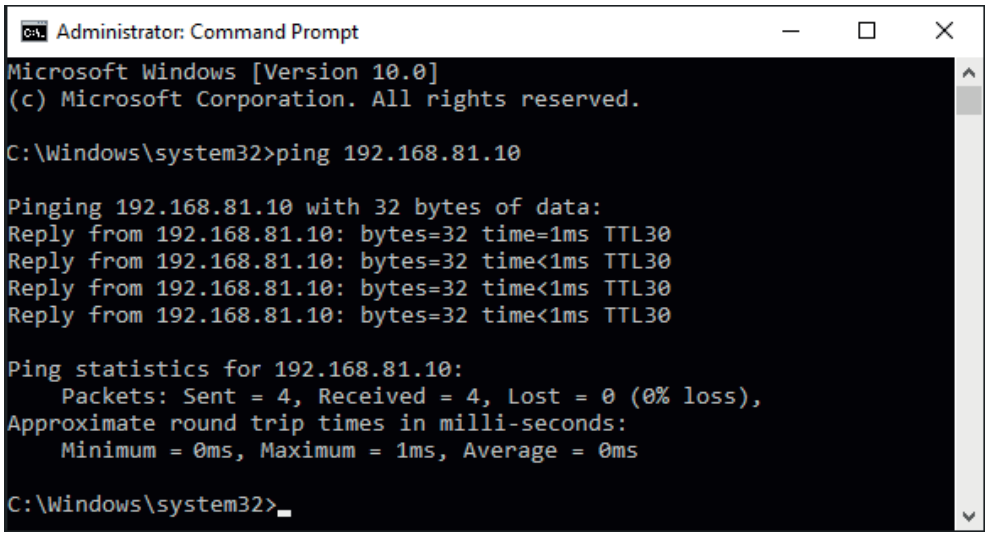
6.3 Connection diagnostics with PING

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

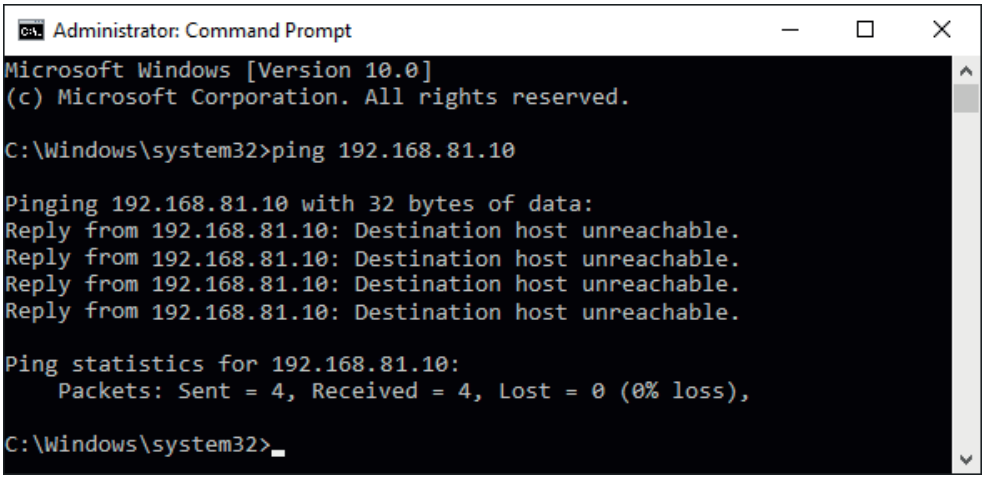
1. Open a Windows command prompt.

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

→ With an existing connection you receive several replies.



→ With no existing connection you receive error messages.



7 Technical data

Below you will find the technical data for the data interface *ibaPDA-Request-TwinCAT*.

Short 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

Email: support@iba-ag.com

Note

If you need support for software products, please state the number of the license container. For hardware products, please have the serial number of the device ready.

Contact

Headquarters

iba AG
Koenigswarterstrasse 44
90762 Fuerth
Germany

Phone: +49 911 97282-0

Email: iba@iba-ag.com

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