ibaPDA-Interface-EGD

Data interface to EGD (Ethernet Global Data)

Manual

Issue 2.0

Measurement Systems for Industry and Energy

www.iba-ag.com

Manufacturer

iba AG
Koenigswarterstrasse 44
90762 Fuerth
Germany

Contacts

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

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

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

Version	Date	Revision	Author	Version SW
2.0	07-2024	New version ibaPDA v8	nm	8.6.0

Contents

1 About this documentation

This documentation describes the function and application of the software interface

ibaPDA-Interface-EGD.

This documentation is a supplement to the *ibaPDA* manual. Information about all the other characteristics and functions of *ibaPDA* can be found in the *ibaPDA* manual or in the online help.

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.

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 the use of the EGD data interface:

- *ibaPDA* v7.0.0 or higher
- License for *ibaPDA-Interface-EGD*
- The following controllers are supported:
  - GE Energy Power Conversion (formerly Converteam) HPCi
  - Converteam Alspa 8035
  - GE Fanuc 9030, 9070
  - GE Fanuc RX3i, RX7i

ibaPDA specific limitations

- ibaPDA acts as a consumer only
- ibaPDA supports up to 64 exchanges (from multiple producers)

Note

It is highly recommended to operate the EGD communication on a separate network.

An additional network interface card may be needed in order to avoid interferences of the EGD messages with the Ethernet traffic from the *ibaPDA* system to other network nodes (fileservers, users consulting data files …).

Licenses

Order no.	Product name	Description
31.001070	ibaPDA-Interface-EGD	Extension license for an *ibaPDA* system with an EGD interface Number of connections 64
31.101070	one-step-up-Interface-EGD	Extension license for an existing interface *ibaPDA-Interface-EGD* for another 64 connections, maximum of 3 extension licenses permissible

3 EGD interface

Using a UDP driver, it is possible to handle the EGD protocol via UDP/IP.

EGD protocol

GE Fanuc Automation and GE Drive Systems developed an Ethernet Global Data (EGD) exchange for PLC and computer data in 1998. EGD uses UDP or datagram messages for fast transfer of up to 1400 bytes of data from a producer to one or more consumers. EGD protocol messages are classified as either command or data messages.

EGD has been designed to operate on a message-based, connectionless network transport layer, such as the internet UDP/IP protocol. Each protocol message is sent to a specific network access point (UDP port) on the destination node(s).

EGD supports the ability to share information between controllers (nodes) in a networked environment. EGD allows one controller, referred to as the producer of the data, to simultaneously send information to any number of peer controllers (consumers) at a fixed periodic rate.

In addition, EGD supports a set of commands for accessing data and protocol information on EGD nodes. EGD also provides a mechanism for sharing configuration information among nodes. EGD protocol messages are categorized as command, data, or configuration messages.

Command messages can be used to monitor and control the operation of EGD on the destination node.

Data messages are individually configured to send a sample of data at a fixed periodic rate. Each data message that a node sends or receives is associated with a specific identifier, which uniquely defines the configuration of the data sample. This configuration is referred to as an exchange. EGD allows the configuration of exchanges that are sent to a single destination address (IP Unicast addressing), a group of addresses (IP multicast addressing), or to all EGD nodes (IP Broadcast addressing).

The following table shows the defined network parameters for EGD:

Parameter	UDP Port
Command Port	7937 (1F01H)
Data Port	18246 (4746H)

4 Configuration of an EGD exchange

Below you will find explanations of the configuration using an example with the HPCi system from GE Energy Power Conversion.

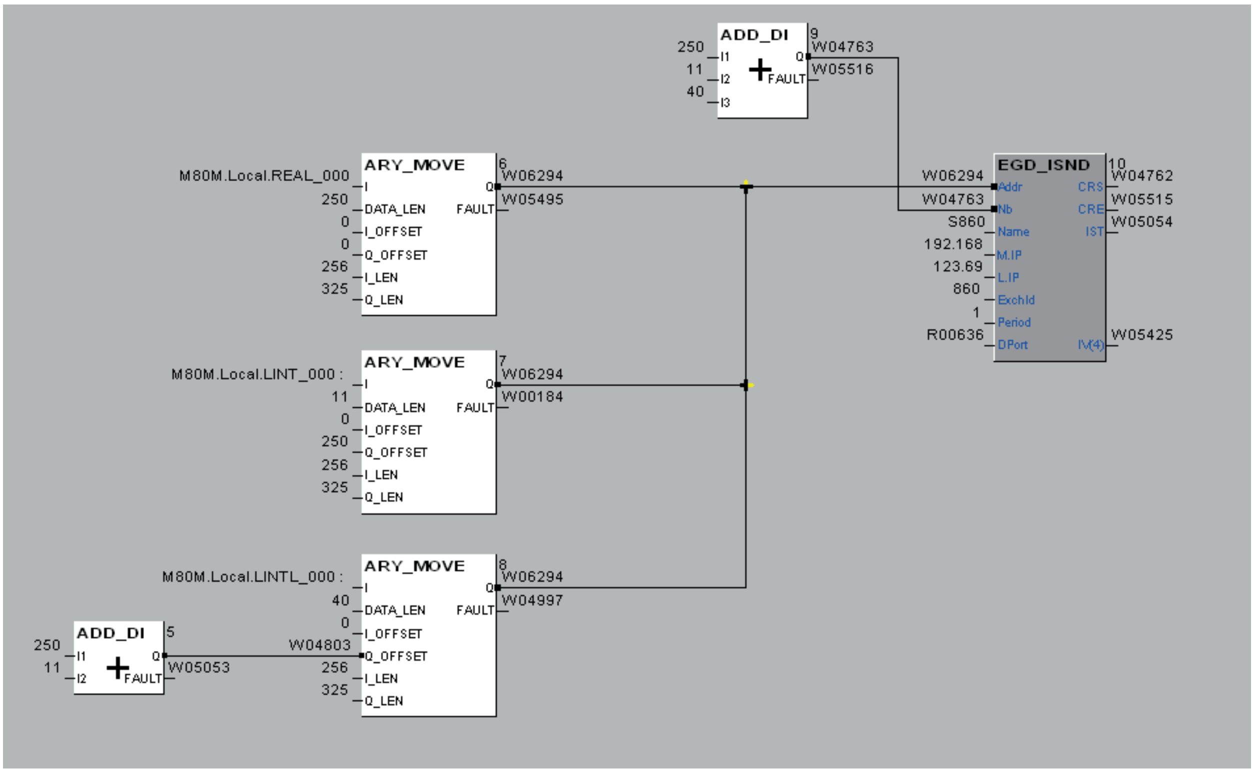
This documentation will be extended as more systems are applied.

4.1 Configuration of the GE Energy HPCi controller

1. Setting up the EGD send block

By using the EGD_ISND function an EGD producer exchange can be set up. (Please refer to the online help for more information).

- The Name input should be a unique name identifying this exchange.
- M.IP is the high part of the IP address of the consumer remote node. In our case this is the IP address of the ibaPDA system.
- L.IP Remaining part of the IP address of consumer remote node.
- ExchId : Ident of the exchange. This ID together with Producer ID uniquely identifies the exchange for the ibaPDA system.



2. Gather the producer data

In the example above the ARY_MOVE block is used to build the producer data. The producer data can contain up to 1400 bytes of data.

The following data types are possible for use with *ibaPDA*:

- BYTE (8 Bit unsigned integer)
- INT (16 Bit signed integer)
- WORD (16 Bit unsigned integer)
- DINT (32 Bit signed integer)
- DWORD (32 Bit unsigned integer)
- FLOAT (32 Bit IEEE real)
- Any combination of above mentioned type (as structure or array)

Other documentation

Please see the P80i online help for further information.

4.2 Configuration ibaPDA

Once the controller has been successfully configured, proceed with the configuration in *ibaPDA*. For information on configuration using the GE Energy HPCi controller as an example, see ↗ *Configuration of the GE Energy HPCi controller*, page 8.

If the EGD interface is enabled in the *ibaPDA* dongle, you can see it in the tree view of the I/O Manager. If some connections from the controller to this ibaPDA system are running, they will be listed here.

Two module types are available:

- EGD

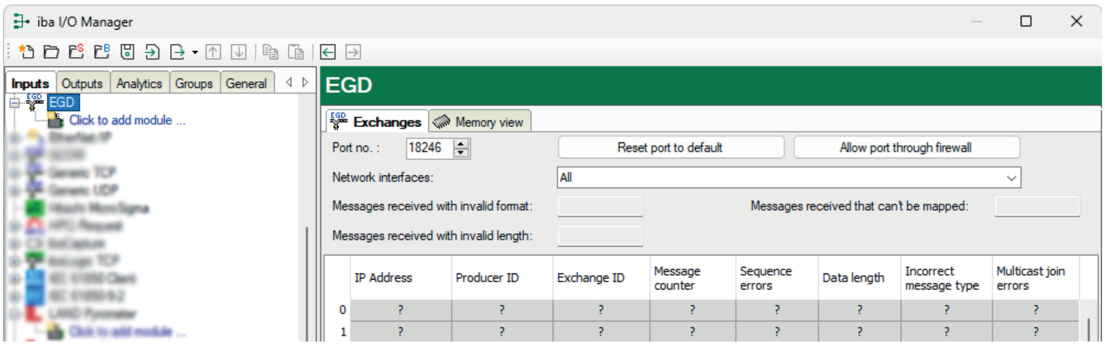
  See ↗ *EGD module (unicast)*, page 12

- EGD multicast

  See ↗ *EGD multicast module*, page 14

4.2.1 Interface settings

The interface has the following functions and configuration options.



Port no. : 18246

Network interfaces: All

Messages received with invalid format:

Messages received that can't be mapped:

Messages received with invalid length:

	IP Address	Producer ID	Exchange ID	Message counter	Sequence errors	Data length	Incorrect message type	Multicast join errors
0	?	?	?	?	?	?	?	?
1	?	?	?	?	?	?	?	?

Port no.
Port number that *ibaPDA* listens to for messages from an EGD producer. Usually, the default port no. 18246 can remain unchanged.

<Reset port to default>
Use this button to reset the port to the default port number.

Allow ports through firewall
When installing *ibaPDA*, the default port numbers of the used protocols are automatically entered in the firewall. If you change the port number or enable the interface subsequently, you have to enable this port in the firewall with this button.

Network interfaces
Using this drop-down list, you can select which network adapters on your computer are used for this interface. The sockets will be opened for communication only on the selected network adapters. In case a network adapter has multiple IP addresses configured, a socket will be opened for all of these IP addresses. At least one network adapter should be selected to get the interface configuration validated. If you select *None*, an error message will be displayed when validating the I/O configuration. By default, the option *All* is selected.

Error counter: Messages
Three global error counters are displayed above the connection table:

- *Messages received with invalid format*:

  Counts up when a message with one of the following properties is received:

  - PDU-Typ is not 13
  - PVN is not 1
  - Request ID is not in the permitted range

- *Messages received with invalid length*:

  Counts up when a message with one of the following properties is received:

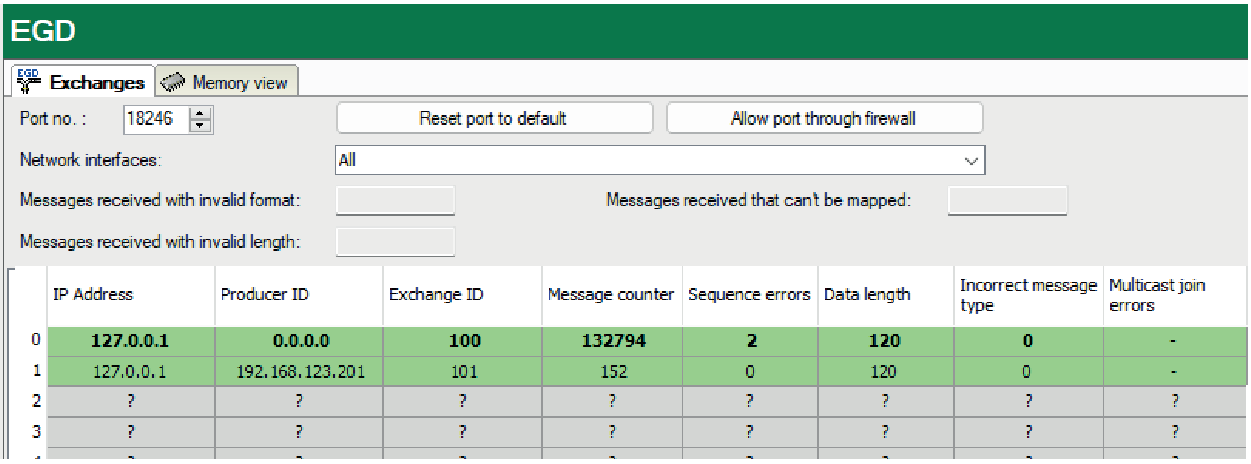
  - The length of the message is shorter than the message header (32 bytes).
  - The length of the message is longer than the maximum message length (1432 bytes).

- *Messages received that can’t be mapped*:

  Counts up if a data packet from a new Exchange is received but all 64 available Exchange entries have already been used.

Connection table
The table shows the individual EGD connections and 64 possible exchanges. The active (connected) exchanges are highlighted in green.

For explanations of the various fields in the message, see ➚ *Message layout - data generation*, page 26.



Port no. : 18246

Network interfaces: All

Messages received with invalid format:

Messages received that can't be mapped:

Messages received with invalid length:

	IP Address	Producer ID	Exchange ID	Message counter	Sequence errors	Data length	Incorrect message type	Multicast join errors
0	**127.0.0.1**	**0.0.0.0**	**100**	**132794**	**2**	**120**	**0**	**-**
1	127.0.0.1	192.168.123.201	101	152	0	120	0	-
2	?	?	?	?	?	?	?	?
3	?	?	?	?	?	?	?	?

The table columns and their meaning:

- *IP Address:*

  IP address of the connected controller, i. e., of the producing node. This can be the IP address of the producer or the multicast address.

- *Producer ID*:

  Should correspond to the IP address.

- *Exchange ID*:

  Should correspond to the specified exchange ID on the producing node.

- *Message counter*:

  Continuous counter that is incremented by 1 with each data packet received. This message counter is also used to control if the producer of this exchange still exists.

- *Sequence errors*:

  Increments if the request ID of the previous data packet incremented by 1 does not correspond to the request ID of the newly received data packet. This can indicate that a data sample is lost.

- *Data length*:

  Indicates the length of the received data packet.

- *Incorrect message type*:

  Increments when unicast messages are sent to multicast connections and vice versa.

- *Multicast join errors*:

  Increments if the joining of *ibaPDA* to the multicast group fails. If the exchange is a unicast exchange, there is a minus sign in front of the value.

For more information on the connection table and how to proceed in the event of connection problems, see ➚ *Checking individual connections*, page 19.

4.2.2 Adding a module

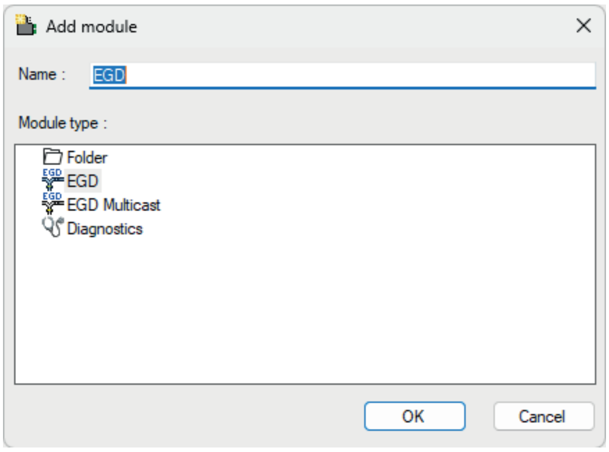
Below you will learn how to add a module to the EGD interface. Detailed information on the individual module types and their configuration can be found in the corresponding chapter:

➚ *EGD module (unicast)*, page 12

➚ *EGD multicast module*, page 14

Procedure:

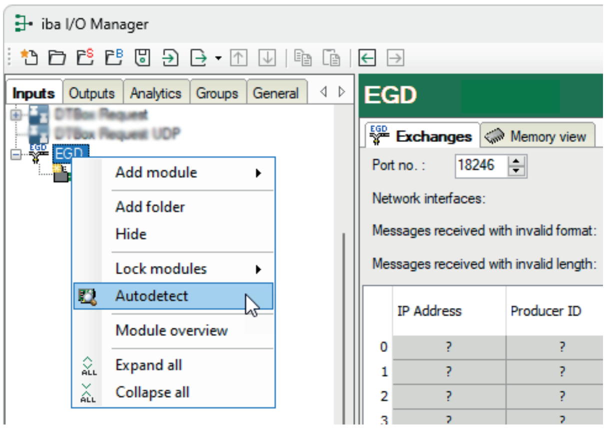
1. Click on the blue link *Click to add module* located under each data interface in the *Inputs* or *Outputs* tab.
2. Select the desired module type in the dialog box and assign a name via the input field if required.
3. Confirm the selection with <OK>.



Name : EGD

4.2.3 EGD module (unicast)

You can create modules for these exchanges easily with a right click on the “EGD” interface icon in the tree view and select “Autodetect”.



Port no. : 18246

	IP Address	Producer ID
0	?	?
1	?	?
2	?	?
3	?	?

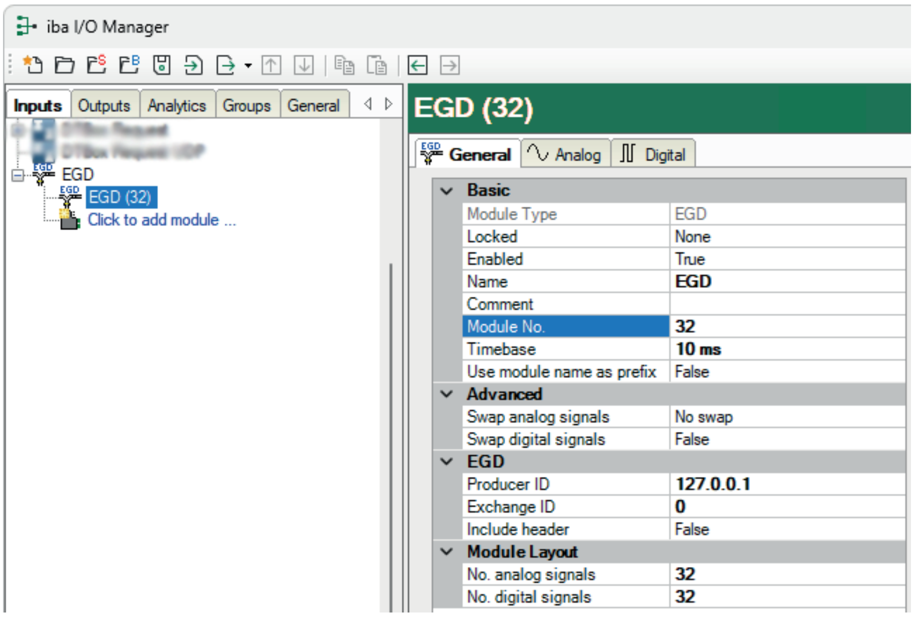
By applying the *Autodetect*, an EGD module with 32 analog and 32 digital signals is created for each active exchange.

Alternatively, you can also create these modules manually without an existing Exchange, see ➚ *Adding a module*, page 12.

Note

The module can also be used to receive EGD broadcasts.

You can configure the following module settings in the *General* tab:



Basic	
Module Type	EGD
Locked	None
Enabled	True
Name	**EGD**
Comment	
Module No.	**32**
Timebase	**10 ms**
Use module name as prefix	False
Advanced	
Swap analog signals	No swap
Swap digital signals	False
EGD	
Producer ID	**127.0.0.1**
Exchange ID	**0**
Include header	False
Module Layout	
No. analog signals	**32**
No. digital signals	**32**

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.

EGD

Producer ID
The Producer ID is a 4 byte unsigned integer, used to uniquely identify the producer of an exchange in a given network. The Producer ID value is generally assigned by a configuration tool and is set to IP address of the producing node (in network byte order) by default. So here the IP address of the producing node should be set.

Exchange ID
The Exchange ID is a 4 byte unsigned integer, used to uniquely identify a particular exchange definition on a specific producing node. The most significant 2 bytes (MSB) of the Exchange ID must be zero in this version of the protocol. This exchange ID should match the specified Exchange ID on the producing node.

Include headers
If enabled, not only the “pure“ data will be measured, also the header information of the EGD UDP-packet will be accessible by *ibaPDA*. This makes only sense for debugging purposes.

Advanced

Swap mode.
Choose the required swap mode from the drop-down list. Which mode is the right one depends on the system.

- No swap Default
- Depending on data type
- Swap 16 bit

Swap digital signals
Choose whether the digital signals should be swapped on a 4 byte base.

- False: No swap , default
- True: Byte order ABCD becomes DCBA

Module Layout

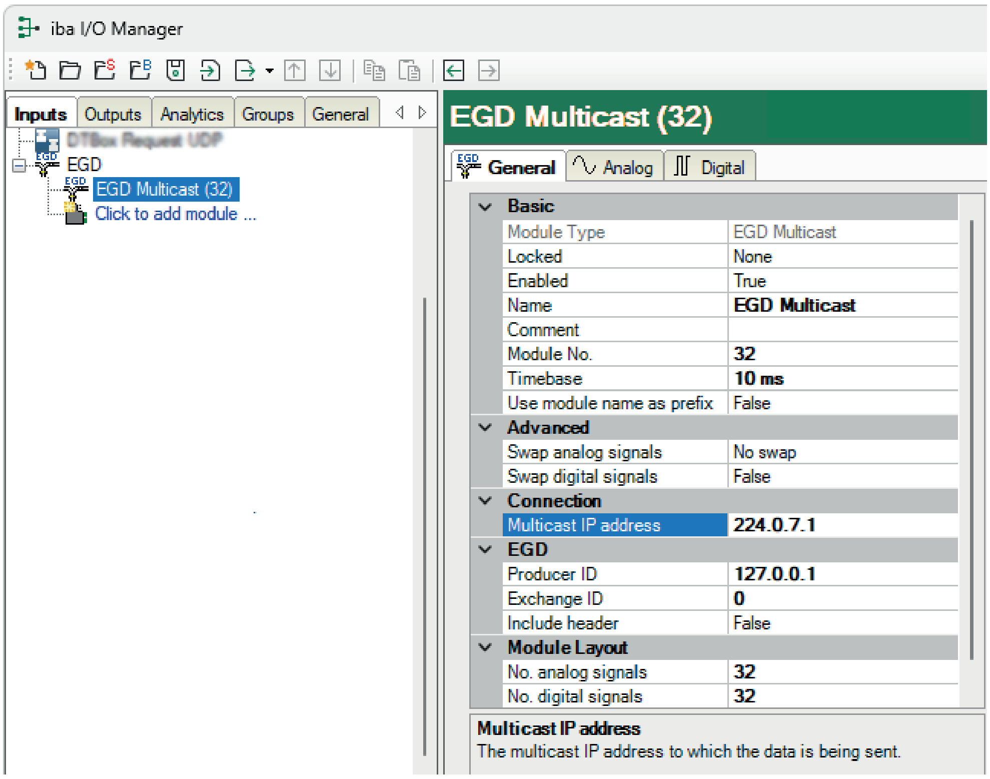
No. analog signals/No. digital signals
Define the number of configurable analog and digital signals in the signal tables. The default value is 32 for each. The maximum value is 1000. The signal tables are adjusted accordingly.

4.2.4 EGD multicast module

This type of module should be created manually.

For information on the basic settings, EGD settings, advanced settings and module structure settings, see ➚ *EGD module (unicast)*, page 12.

The only difference to the normal EGD module is a property called *Multicast IP address*. Here you have to enter the IP address of the multicast group that the EGD source is sending to. *ibaPDA* will join the multicast group and start receiving multicast EGD data when the acquisition is started.

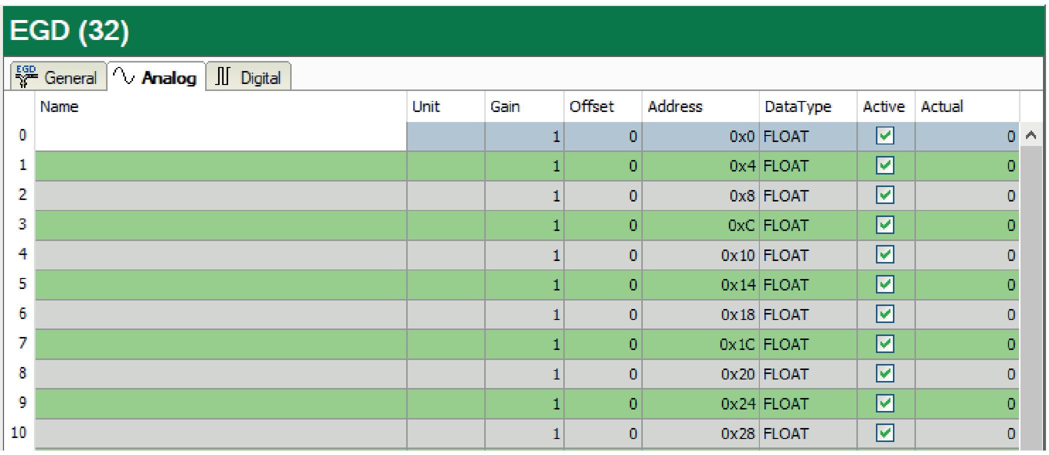


Basic	
Module Type	EGD Multicast
Locked	None
Enabled	True
Name	**EGD Multicast**
Comment	
Module No.	**32**
Timebase	**10 ms**
Use module name as prefix	False
Advanced	
Swap analog signals	No swap
Swap digital signals	False
Connection	
Multicast IP address	**224.0.7.1**
EGD	
Producer ID	**127.0.0.1**
Exchange ID	**0**
Include header	False
Module Layout	
No. analog signals	**32**
No. digital signals	**32**

Multicast IP address
The multicast IP address to which the data is being sent.

4.2.5 Signal configuration

Configure the signals to be measured in the *Analog* or *Digital* tabs. Set the length of the signal tables, i.e. the number of signals per table, in the *General* tab under *Module structure*.



	Name	Unit	Gain	Offset	Address	DataType	Active	Actual
0			1	0	0x0	FLOAT	☑	0
1			1	0	0x4	FLOAT	☑	0
2			1	0	0x8	FLOAT	☑	0
3			1	0	0xC	FLOAT	☑	0
4			1	0	0x10	FLOAT	☑	0
5			1	0	0x14	FLOAT	☑	0
6			1	0	0x18	FLOAT	☑	0
7			1	0	0x1C	FLOAT	☑	0
8			1	0	0x20	FLOAT	☑	0
9			1	0	0x24	FLOAT	☑	0
10			1	0	0x28	FLOAT	☑	0

Name
Enter a meaningful plain text name for the signal

Unit (analog signals only)
Assignment of an enineering unit (such as Ampere, Volt, m/s, tons) for the signal

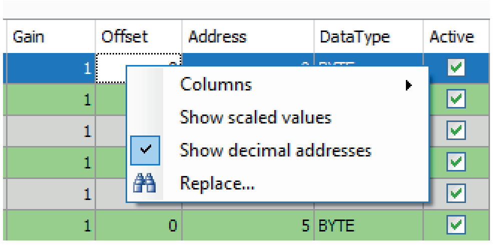
You can enter a maximum of 11 characters, the field is only considered a comment field. The unit is always displayed in conjunction with a numerical display of the values.

Gain, Offset (analog signals only)
Specification of gain and offset for scaling the incoming values

The values are used to describe a linear characteristic curve for scaling. If incoming values are specified in engineering units, this function can be ignored, i. e. Gain = 1 and Offset = 0.

Adresse
Position in the user data area of a telegram or dual-port memory (specified in bytes) where the desired signal is located

The offset can be entered as hexadecimal or decimal value.



In order to get some default values just click on the column header. The offset values are filled in automatically starting with the value in the first row, respectively in the field the cursor is currently in, downwards in address steps according to the selected data type. For digital signals the *Bit no.* is automatically increased.

- Analog signals (EGD module) as FLOAT-, DINT- or DWORD: 4-byte-steps
- Analog signals (EGD module) as INT or WORD: 2-byte-steps
- Analog signals (EGD module) as BYTE: 1-byte-steps
- For digital signals, the bit number is incremented by 1 from 0...31, then the address is incremented by 4.

If you enter all signal definitions with name and data type and click on *Address*, *ibaPDA* will automatically calculate the correct address offsets, based on the address of the first signal.

Data type (analog signals only)
Selection of the data type of the signal

The address of the next signal is determined by the data type.

Active
Activation or deactivation of the respective signal

Actual
Display of the current actual value of the signal

Other documentation

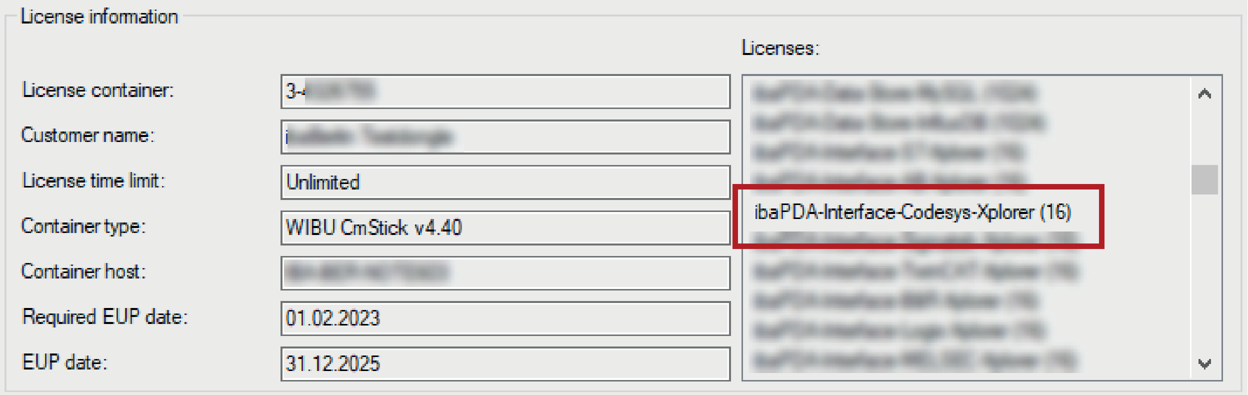
Detailed descriptions of the columns and how to fill in the signal tables can be found in the documentation for *ibaPDA*.

5 Diagnostics

5.1 License

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

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



License information

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

Licenses:

[redacted]

ibaPDA-Interface-Codesys-Xplorer (16)

[redacted]

5.2 Visibility of the interface

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

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

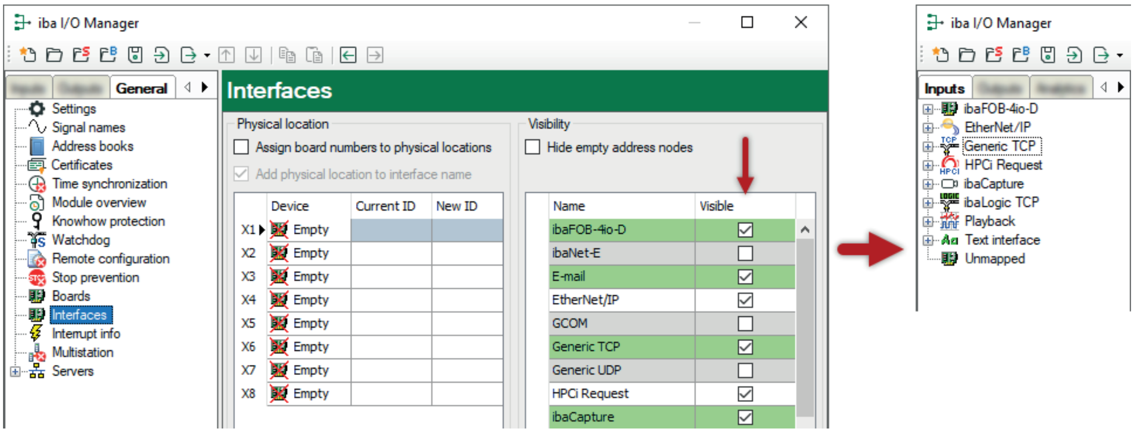
Visibility

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

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

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

Selected interfaces are visible, the others are hidden:



Interfaces

Physical location

☐ Assign board numbers to physical locations

☑ Add physical location to interface name

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

Visibility

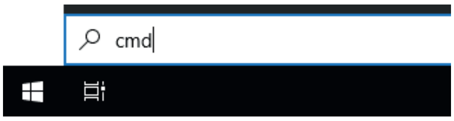
☐ Hide empty address nodes

Name	Visible
ibaFOB-4io-D	☑
ibaNet-E	☐
E-mail	☑
EtherNet/IP	☑
GCOM	☐
Generic TCP	☑
Generic UDP	☐
HPCi Request	☑
ibaCapture	☑

5.3 Connection diagnostics with PING

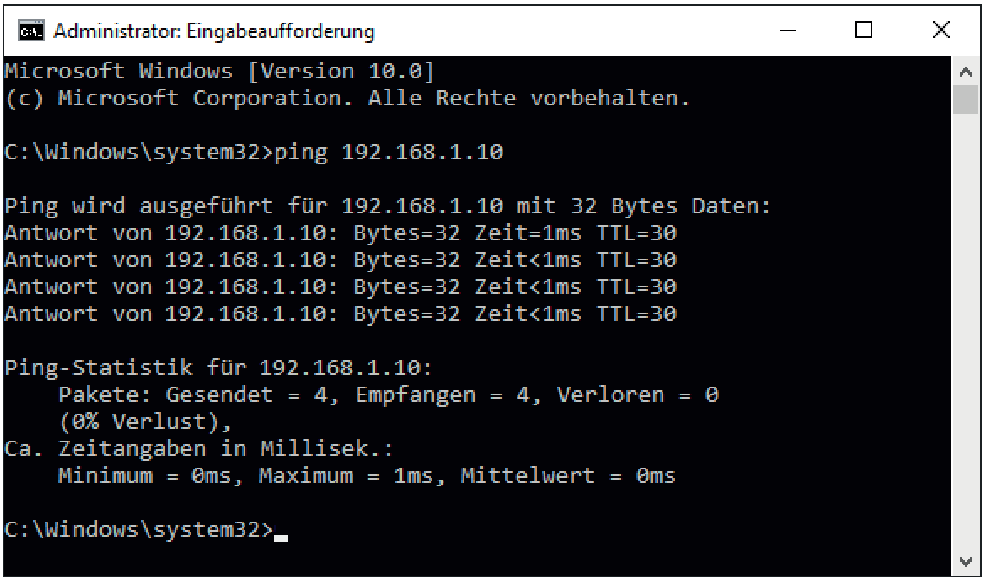
PING is a system command with which you can check if a certain communication partner can be reached in an IP network.

1. Open a Windows command prompt.



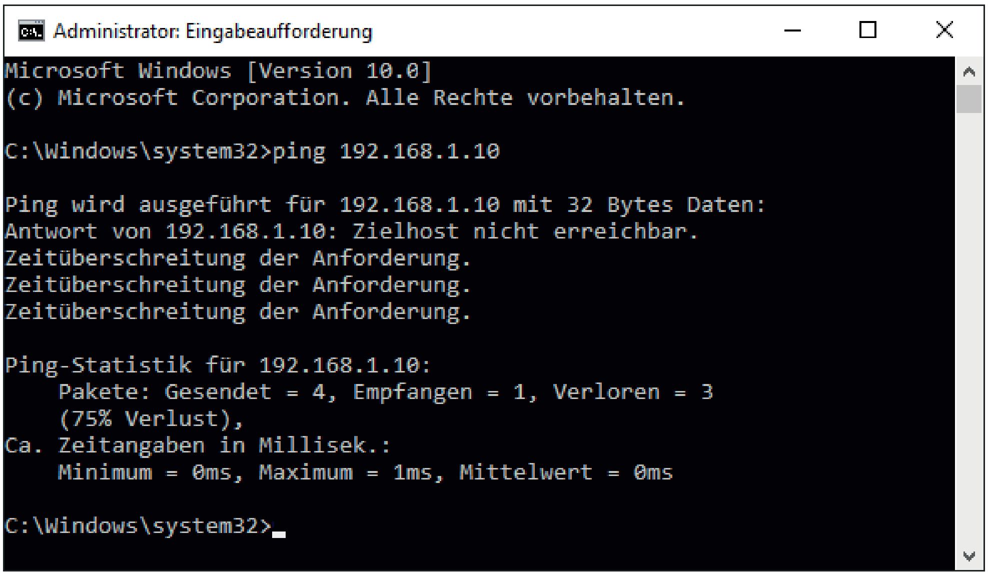
2. Enter the command "ping" followed by the IP address of the communication partner and press <ENTER>.

→ With an existing connection you receive several replies.



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

C:\Windows\system32>ping 192.168.1.10

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

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

C:\Windows\system32>_
```

→ With no existing connection you receive error messages.



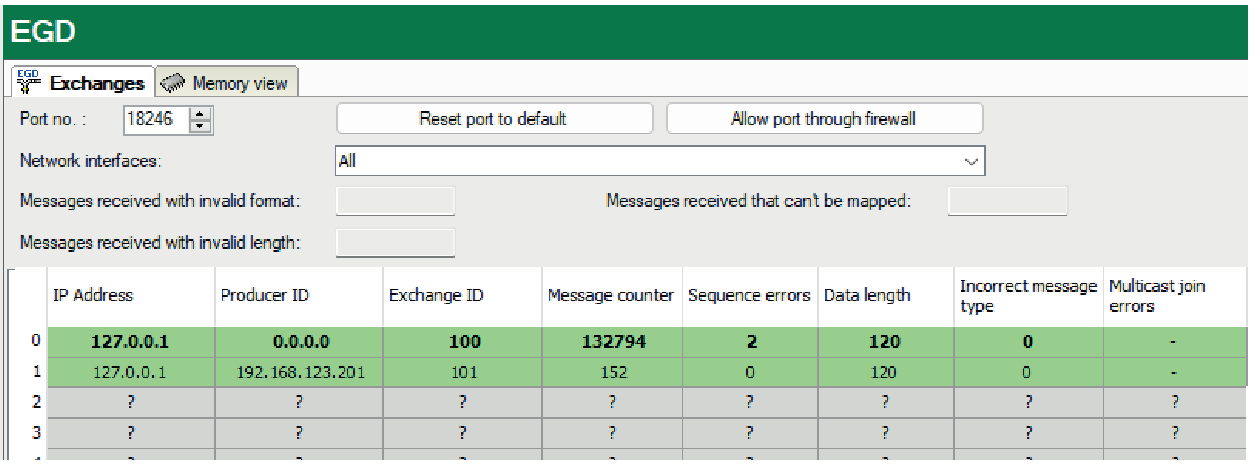
```
Microsoft Windows [Version 10.0]
(c) Microsoft Corporation. Alle Rechte vorbehalten.

C:\Windows\system32>ping 192.168.1.10

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

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

C:\Windows\system32>_
```

5.4 Checking individual connections

In case of trouble, first of all check the network settings and verify the IP addresses of both producer and consumer (ibaPDA system).

Use ping to check the network functionality, see ↗ *Connection diagnostics with PING*, page 18.

On the EGD interface examine the table of the different exchanges. For detailed information on the fields and error counters, see ↗ *Interface settings*, page 9.



EGD

Port no. : 18246

Network interfaces: All

Messages received with invalid format:

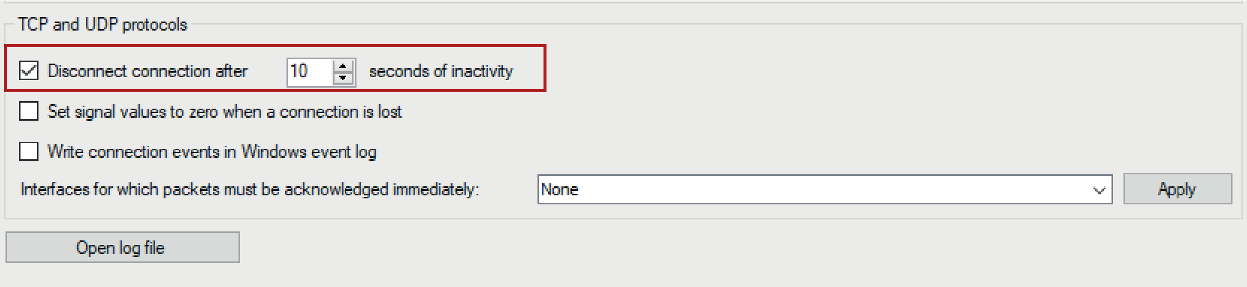
Messages received that can't be mapped:

Messages received with invalid length:

	IP Address	Producer ID	Exchange ID	Message counter	Sequence errors	Data length	Incorrect message type	Multicast join errors
0	**127.0.0.1**	**0.0.0.0**	**100**	**132794**	**2**	**120**	**0**	**-**
1	127.0.0.1	192.168.123.201	101	152	0	120	0	-
2	?	?	?	?	?	?	?	?
3	?	?	?	?	?	?	?	?

Timeout setting

In the I/O Manager under *General - Settings* there is an option that allows you to delete unused exchanges.



TCP and UDP protocols

☑ Disconnect connection after 10 seconds of inactivity

☐ Set signal values to zero when a connection is lost

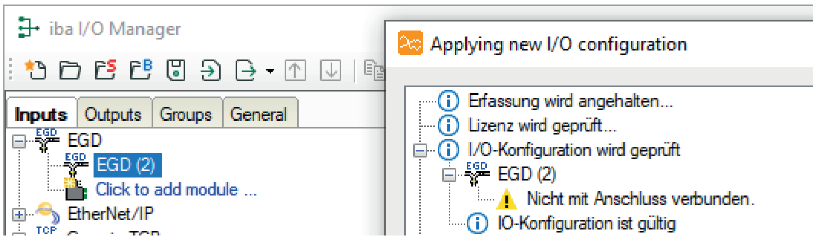
☐ Write connection events in Windows event log

Interfaces for which packets must be acknowledged immediately: None

Disconnect connectiotion after... seconds of inactivity

Enabling of this option will activate a timeout for all TCP/IP connections, including EGD exchanges. If no messages are received over the EGD exchange within the time (set in the field next to the checkbox), then the corresponding exchange will be closed and the last received production data for that exchange will be reset to 0. Closing unused exchanges saves resources and free up unused, dead exchanges.

Once the data-acquisition is running each of the 64 possible exchanges can have following states:

Color	Text	Meaning
Green	bold	the exchange is running and there is an EGD module assigned to it. The module is locked by the data acquisition system. Even when the producer stops sending data and the exchange is closed by the timeout, the exchange remains locked to reserve this entry in case the producer restarts sending data samples.
	not bold	the exchange is running but there is no EGD module assigned to it.
Red	-	The connection has failed or been interrupted.
Gray	bold	the exchange is not running but locked because an EGD module with the indicated Producer ID and Exchange ID is defined. If the exchange is not active at the time the data acquisition is started, a warning appears in the validate dialog as shown in the figure below.  iba I/O Manager Inputs Outputs Groups General EGD EGD (2) Click to add module ... EtherNet/IP Applying new I/O configuration Erfassung wird angehalten... Lizenz wird geprüft... I/O-Konfiguration wird geprüft EGD (2) Nicht mit Anschluss verbunden. IO-Konfiguration ist gültig
	not bold	No connection configured.

5.5 Diagnostic modules

Diagnostic modules are available for most Ethernet based interfaces and Xplorer interfaces. Using a diagnostic module, information from the diagnostic displays (e.g. diagnostic tabs and connection tables of an interface) can be acquired as signals.

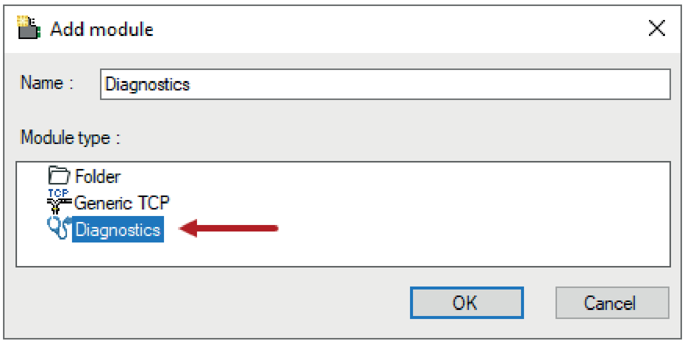
A diagnostic module is always assigned to a data acquisition module of the same interface and supplies its connection information. By using a diagnostic module, you can record and analyze the diagnostic information continuously in the *ibaPDA* system.

Diagnostic modules do not consume any license connections because they do not establish their own connection but refer to another module.

Example for the use of diagnostic modules:

- A notification can be generated, whenever the error counter of a communication connection exceeds a certain value or the connection gets lost.
- In case of a disturbance, the current response times in the telegram traffic may be documented in an incident report.
- The connection status can be visualized in *ibaQPanel*.
- You can forward diagnostic information via the SNMP server integrated in *ibaPDA* or via OPC DA/UA server to superordinate monitoring systems like network management tools.

In case the diagnostic module is available for an interface, a "Diagnostics" module type is shown in the "Add module" dialog (example: Generic TCP).

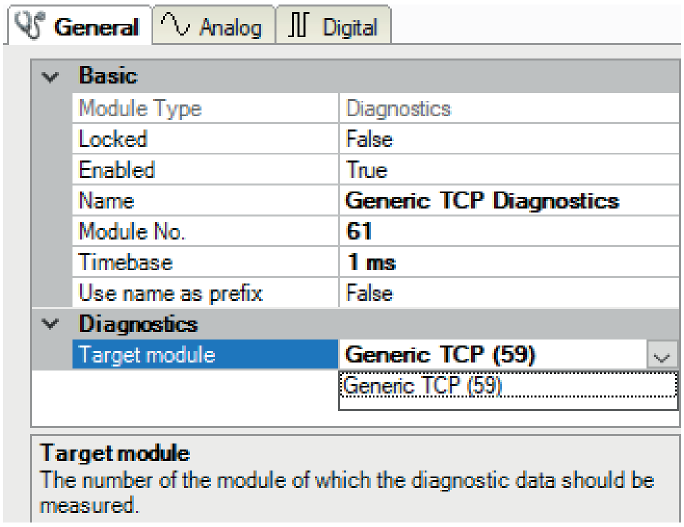

Add module

Name : Diagnostics

Module type :

Module settings diagnostic module

For a diagnostic module, you can make the following settings (example: Generic TCP):



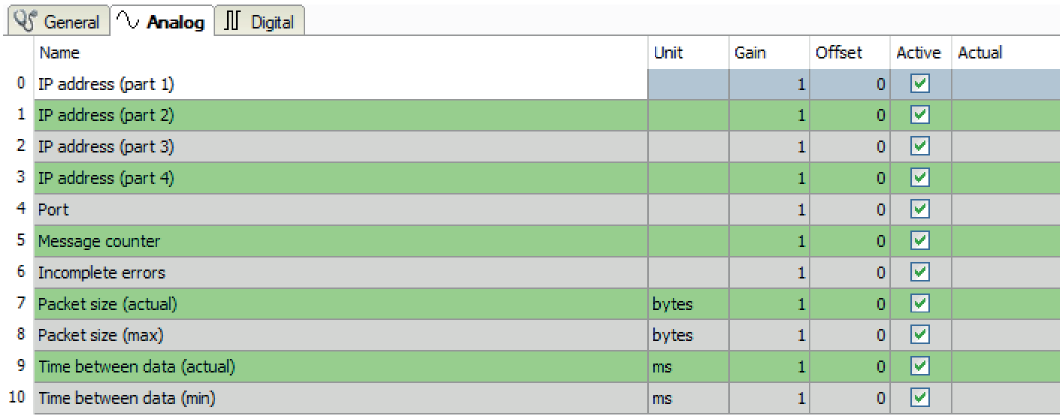
Basic	
Module Type	Diagnostics
Locked	False
Enabled	True
Name	**Generic TCP Diagnostics**
Module No.	**61**
Timebase	**1 ms**
Use name as prefix	False
Diagnostics	
Target module	**Generic TCP (59)**
	Generic TCP (59)

Target module
The number of the module of which the diagnostic data should be measured.

The basic settings of a diagnostic module equal those of other modules.

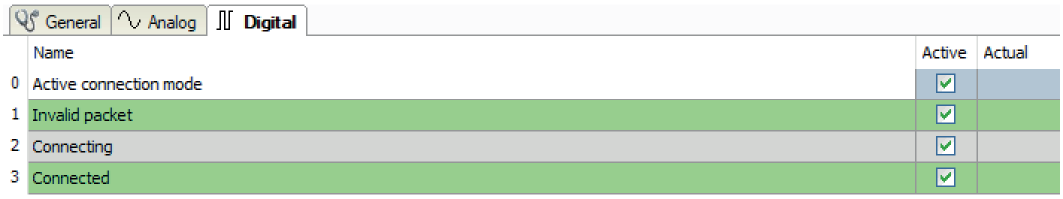
There is only one setting which is specific for the diagnostic module: the target module.

By selecting the target module, you assign the diagnostic module to the module on which you want to acquire information about the connection. You can select the supported modules of this interface in the drop-down list of the setting. You can assign exactly one data acquisition module to each diagnostic module. When having selected a module, the available diagnostic signals are immediately added to the *Analog* and *Digital* tabs. It depends on the type of interface, which signals exactly are added. The following example lists the analog values of a diagnostic module for a Generic TCP module.



	Name	Unit	Gain	Offset	Active	Actual
0	IP address (part 1)		1	0	☑	
1	IP address (part 2)		1	0	☑	
2	IP address (part 3)		1	0	☑	
3	IP address (part 4)		1	0	☑	
4	Port		1	0	☑	
5	Message counter		1	0	☑	
6	Incomplete errors		1	0	☑	
7	Packet size (actual)	bytes	1	0	☑	
8	Packet size (max)	bytes	1	0	☑	
9	Time between data (actual)	ms	1	0	☑	
10	Time between data (min)	ms	1	0	☑	

For example, the IP (v4) address of a Generic TCP module (see fig. above) will always be split into 4 parts derived from the dot-decimal notation, for better reading. Also other values are being determined, as there are port number, counters for telegrams and errors, data sizes and telegram cycle times. The following example lists the digital values of a diagnostic module for a Generic TCP module.



	Name	Active	Actual
0	Active connection mode	☑	
1	Invalid packet	☑	
2	Connecting	☑	
3	Connected	☑	

Diagnostic signals

Depending on the interface type, the following signals are available:

Signal name	Description
Active	Only relevant for redundant connections. Active means that the connection is used to measure data, i.e. for redundant standby connections the value is 0. For normal/non-redundant connections, the value is always 1.
Buffer file size (actual/avg/max)	Size of the file for buffering statements
Buffer memory size (actual/avg/max)	Size of the memory used by buffered statements
Buffered statements	Number of unprocessed statements in the buffer
Buffered statements lost	Number of buffered but unprocessed and lost statements
Connected	Connection is established
Connected (in)	A valid data connection for the reception (in) is available
Connected (out)	A valid data connection for sending (out) is available
Connecting	Connection being established
Connection attempts (in)	Number of attempts to establish the receive connection (in)
Connection attempts (out)	Number of attempts to establish the send connection (out)
Connection ID O->T	ID of the connection for output data (from the target system to *ibaPDA*). Corresponds to the assembly instance number
Connection ID T->O	ID of the connection for input data (from *ibaPDA* to target system). Corresponds to the assembly instance number
Connection phase (in)	Status of the ibaNet-E data connection for reception (in)
Connection phase (out)	Status of the ibaNet-E data connection for sending (out)
Connections established (in)	Number of currently valid data connections for reception (in)
Connections established (out)	Number of currently valid data connections for sending (out)
Data length	Length of the data message in bytes
Data length O->T	Size of the output message in byte
Data length T->O	Size of the input message in byte
Destination IP address (part 1-4) O->T	4 octets of the IP address of the target system Output data (from target system to *ibaPDA*)
Destination IP address (part 1-4) T->O	4 octets of the IP address of the target system Input data (from *ibaPDA* to target system)
Disconnects (in)	Number of currently interrupted data connections for reception (in)

Signal name	Description
Disconnects (out)	Number of currently interrupted data connections for sending (out)
Error counter	Communication error counter
Exchange ID	ID of the data exchange
Incomplete errors	Number of incomplete messages
Incorrect message type	Number of received messages with wrong message type
Input data length	Length of data messages with input signals in bytes (*ibaPDA* receives)
Invalid packet	Invalid data packet detected
IP address (part 1-4)	4 octets of the IP address of the target system
Keepalive counter	Number of KeepAlive messages received by the OPC UA Server
Lost images	Number of lost images (in) that were not received even after a retransmission
Lost Profiles	Number of incomplete/incorrect profiles
Message counter	Number of messages received
Messages per cycle	Number of messages in the cycle of the update time
Messages received since configuration	Number of received data telegrams (in) since start of acquisition
Messages received since connection start	Number of received data telegrams (in) since the start of the last connection setup. Reset with each connection loss.
Messages sent since configuration	Number of sent data telegrams (out) since start of acquisition
Messages sent since connection start	Number of sent data telegrams (out) since the start of the last connection setup. Reset with each connection loss.
Multicast join error	Number of multicast login errors
Number of request commands	Counter for request messages from *ibaPDA* to the PLC/CPU
Output data length	Length of the data messages with output signals in bytes (*ibaPDA* sends)
Packet size (actual)	Size of the currently received message
Packet size (max)	Size of the largest received message
Ping time (actual)	Response time for a ping telegram
Port	Port number for communication
Producer ID (part 1-4)	Producer ID as 4-byte unsigned integer
Profile Count	Number of completely recorded profiles
Read counter	Number of read accesses/data requests
Receive counter	Number of messages received

Signal name	Description
Response time (actual/average/max/min)	Response time is the time between measured value request from *ibaPDA* and response from the PLC or reception of the data. Actual: current value Average/max/min: static values of the update time since the last start of the acquisition or reset of the counters.
Retransmission requests	Number of data messages requested again if lost or delayed
Rows (last)	Number of resulting rows by the last SQL query (within the configured range of result rows)
Rows (maximum)	Maximum number of resulting rows by any SQL query since the last start of acquisition (possible maximum equals the configured number of result rows)
Send counter	Number of send messages
Sequence errors	Number of sequence errors
Source IP address (part 1-4) O->T	4 octets of the IP address of the target system Output data (from target system to *ibaPDA*)
Source IP address (part 1-4) T->O	4 octets of the IP address of the target system Input data (from *ibaPDA* to target system)
Statements processed	Number of executed statements since last start of acquisition
Synchronization	Device is synchronized for isochronous acquisition
Time between data (actual/max/min)	Time between two correctly received messages Actual: between the last two messages Max/min: statistical values since start of acquisition or reset of counters
Time offset (actual)	Measured time difference of synchronicity between *ibaPDA* and the ibaNet-E device
Topics Defined	Number of defined topics
Topics Updated	Number of updated topics
Unknown sensor	Number of unknown sensors
Update time (actual/average/configured/max/min)	Specifies the update time in which the data is to be retrieved from the PLC, the CPU or from the server (configured). Default is equal to the parameter "Timebase". During the measurement the real actual update time (actual) can be higher than the set value, if the PLC needs more time to transfer the data. How fast the data is really updated, you can check in the connection table. The minimum achievable update time is influenced by the number of signals. The more signals are acquired, the greater the update time becomes. Average/max/min: static values of the update time since the last start of the acquisition or reset of the counters.
Write counter	Number of successful write accesses
Write lost counter	Number of failed write accesses

6 Further information

Below you will find further information on the structure and layout of messages for the EGD interface.

6.1 Message layout - data generation

The *Data_Production* PDU supports the production of Global Ethernet Data. This packet contains the user data being produced and sent to the consumers. This message is unacknowledged, that means, it is transmitted by the producing node at the configured rate without delaying for a response from the consuming nodes. Receiving nodes that detect an encoding error in a received Data_Production PDU shall discard the PDU and take no action on its content.

Data Production PDU Format

0	PDU Type (13)	PVN = 1	Request ID
4	Producer ID		
8	Exchange ID		
12	Timestamp		
16			
20	Status		Reserved
24	Configuration Signature		Reserved
28	Reserved		
32	Production Data. up to 1400 bytes		

Specific fields for data production
The following table describes specifics of some of these fields:

PDU field	Size (bytes)	Description
PDU Type	1	Must be set to 13. Defines a data message.
PVN	1	Must be set to 1 for compatibility with older protocol versions.
Request ID	2	A 2 byte unsigned integer which is incremented each time a data sample is produced.
Producer ID	4	Producer identifier
Exchange ID	4	Exchange identifier
Time stamp	8	The time stamp value should correspond as closely as practical to when the data sample was captured.
Production status	2	Bit mask indicating the validity of the data sample produced: Bit 0: Set if production error or invalid (old) data Bit 1: Set if time stamp is not synchronized on producer node
Configuration Signature	2	2 byte integer indicating relative version of the user data contained in the sample*

PDU field	Size (bytes)	Description
Production Data	to	User data sample matching configuration specified in Configuration Signature

* Additional documentation/further reading: TCP/IP Tutorial, RFC1180 (http://tools.ietf.org/html/rfc1180.html)

6.2 C-style data message declaration

```
#define  EGD_MAX_PRODUCTION_DATA              1400

#pragma pack (push,1)

typedef struct _EGD_DATA_HEADER {

   unsigned char  PDU_type;         // datatype for datamessages  = 13

   unsigned char  PVN;              // Protocol Version Number    = 1

   unsigned short RequestID;        // a 2 byte unsigned integer which is incremented each time a data sample is produced

   unsigned long  ProducerID;       // a 4 byte unsigned integer , used to uniquely identify the producer

                                    // set to IP address on the producing node (in network byte order) by default

   unsigned long  ExchangeID;       // a 4 byte unsigned integer , used to uniquely identify a particular exchange

                                    // valid range 0 - 0x3FFF  (0 - 16383)

   unsigned long  TimeStampSec;     // a time values in POSIX 1003.4 format , 4 byte seconds

   unsigned long  TimeStampNanoSec; // a time values in POSIX 1003.4 format , 4 byte nanoseconds

   unsigned short Status;           // bit mask indicating the validity

                                    // bit 0: set if production error or invalid data

                                    // bit 1: set if timestamp not synchronized on producer node

   unsigned short Reserved1;

   unsigned short ConfigSignature;  // 2 byte integer indicating relative version of the user data

   unsigned short Reserved2;   unsigned long  Reserved3;

} EGD_DATA_HEADER , *PEGD_DATA_HEADER;


// receive structure UDP packettypedef struct _EGD_DATA_PACKET {


   EGD_DATA_HEADER header;

   unsigned char ProductionData[EGD_MAX_PRODUCTION_DATA];

} EGD_DATA_PACKET,*PEGD_DATA_PACKET;


#pragma pack (pop)
```

7 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