ibaPDA-Interface-EGD

Data Interface to EGD (Ethernet Global Data)

Manual

Issue 1.5

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

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
1.5	02-2021	Note: EGD broadcasts	st/ip	6.31

Content

1 About this Manual

This document 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 manual 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>
File names, paths	"Filename", "Path" Example: "Test.doc"

1.3 Used symbols

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

Danger!

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

- Observe the specified measures.

Warning!

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

- Observe the specified measures.

Caution!

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

- Observe the specified measures

Note

A note specifies special requirements or actions to be observed.

Tip

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

Other documentation

Reference to additional documentation or further reading.

2 System requirements

The following system requirements are necessary for the use of the EGD data interface:

- *ibaPDA* v6.33.2 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 …).

Licences

Order No.	Product name	Description
31.001070	ibaPDA-Interface-EGD	Extension license for an *ibaPDA* system providing an additional EGD interface Number of connections: 64
31.101070	one-step-up-Interface-EGD	Extension license for an existing interface *ibaPDA-Interface-EGD* by another 64 EGD connections, max. 3 permitted

3 The EGD interface

iba has implemented a UDP driver able to handle the EGD (Ethernet Global Data) protocol over UDP/IP.

3.1 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)

Table 1: Network parameters for EGD

4 Configuration of an EGD exchange

Explanatory example with the GE Energy HPCi system.

This documentation will be extended as more systems are applied.

4.1 Step 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.

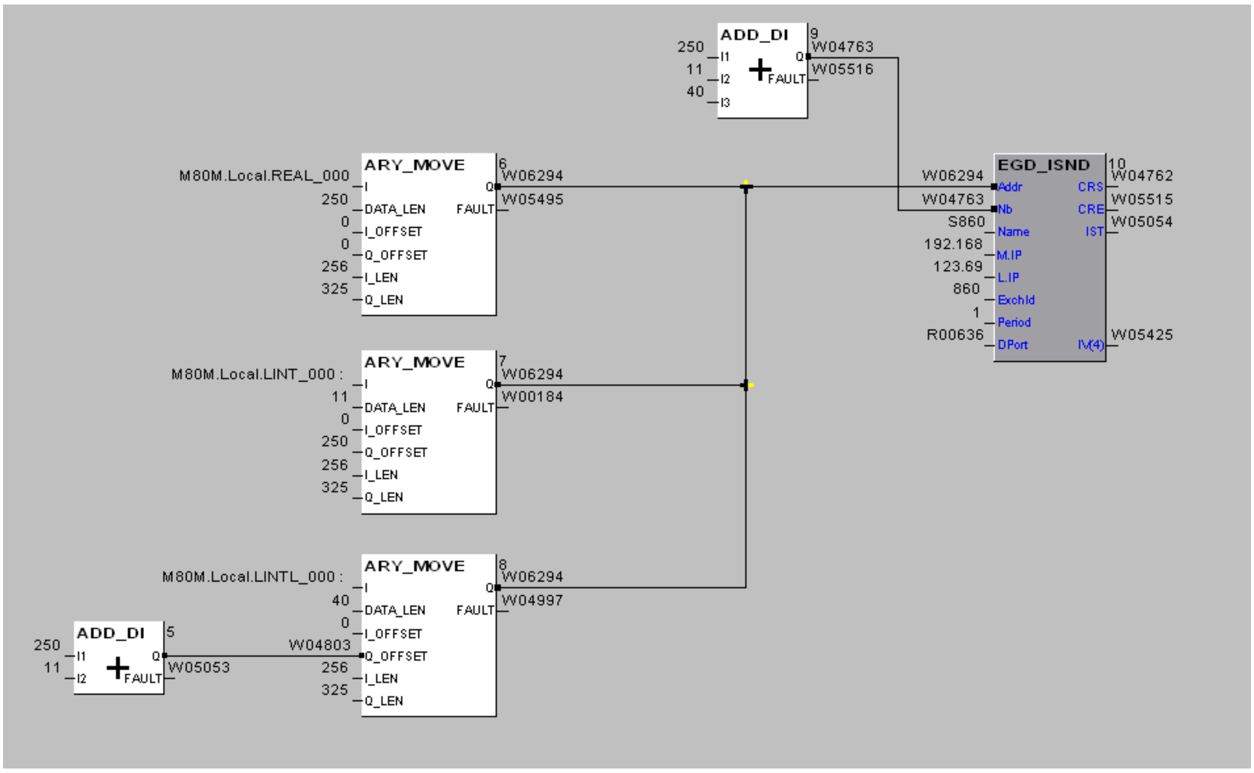


Fig. 1: Configuration of the GE Energy HPCi controller

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 Step 2: Configuration of ibaPDA

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
- EGD multicast

4.2.1 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”.

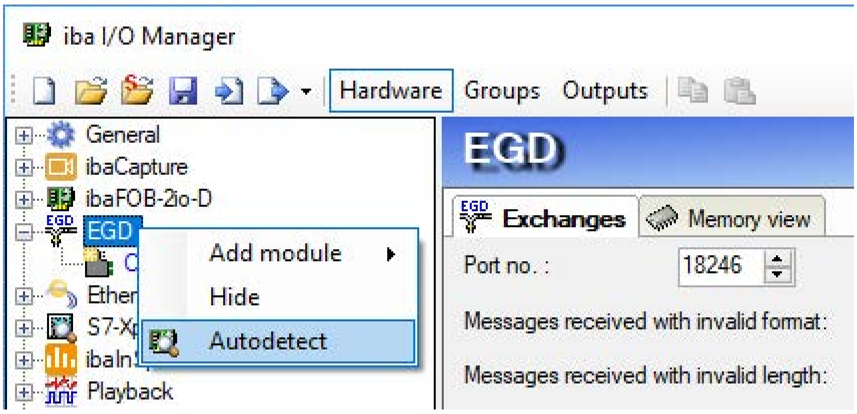


Fig. 2: I/O Manager - Autodetect

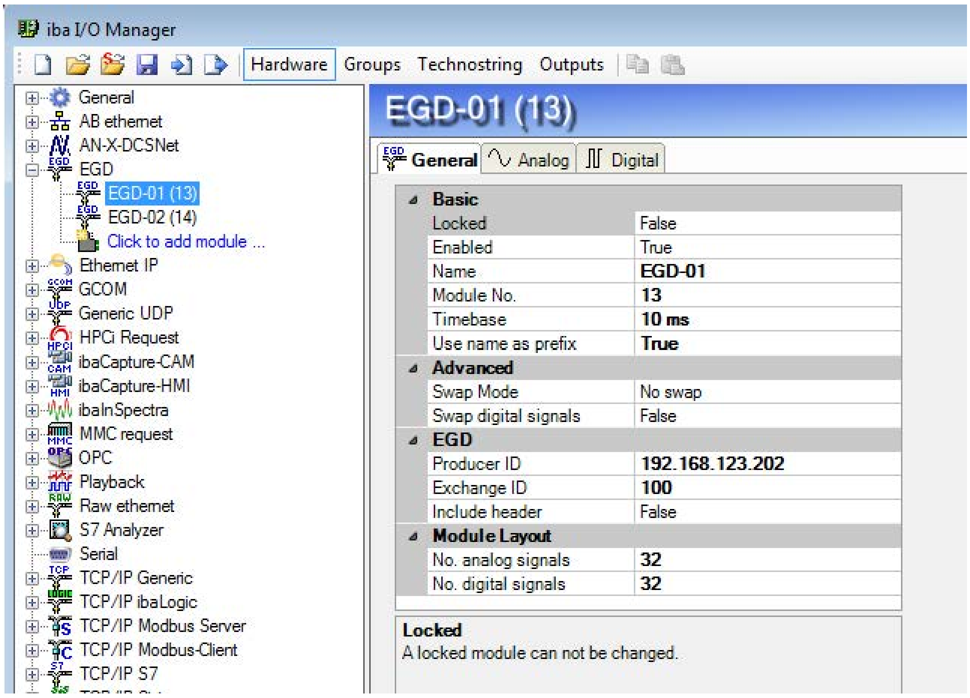
By applying the “autodetect”, an EGD module with 32 analog and 32 digital signals is created for each active exchange.

As an alternative you can create these modules manually without an existing exchange by right clicking the tree item or the blue “Click to add module…” item.

Note

The module can also be used to receive EGD broadcasts.

To configure a module, you should mark it in the tree view and use the following dialog:



Basic	
Locked	False
Enabled	True
Name	**EGD-01**
Module No.	**13**
Timebase	**10 ms**
Use name as prefix	**True**
Advanced	
Swap Mode	No swap
Swap digital signals	False
EGD	
Producer ID	**192.168.123.202**
Exchange ID	**100**
Include header	False
Module Layout	
No. analog signals	**32**
No. digital signals	**32**

Locked
A locked module can not be changed.

Fig. 3: 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.

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

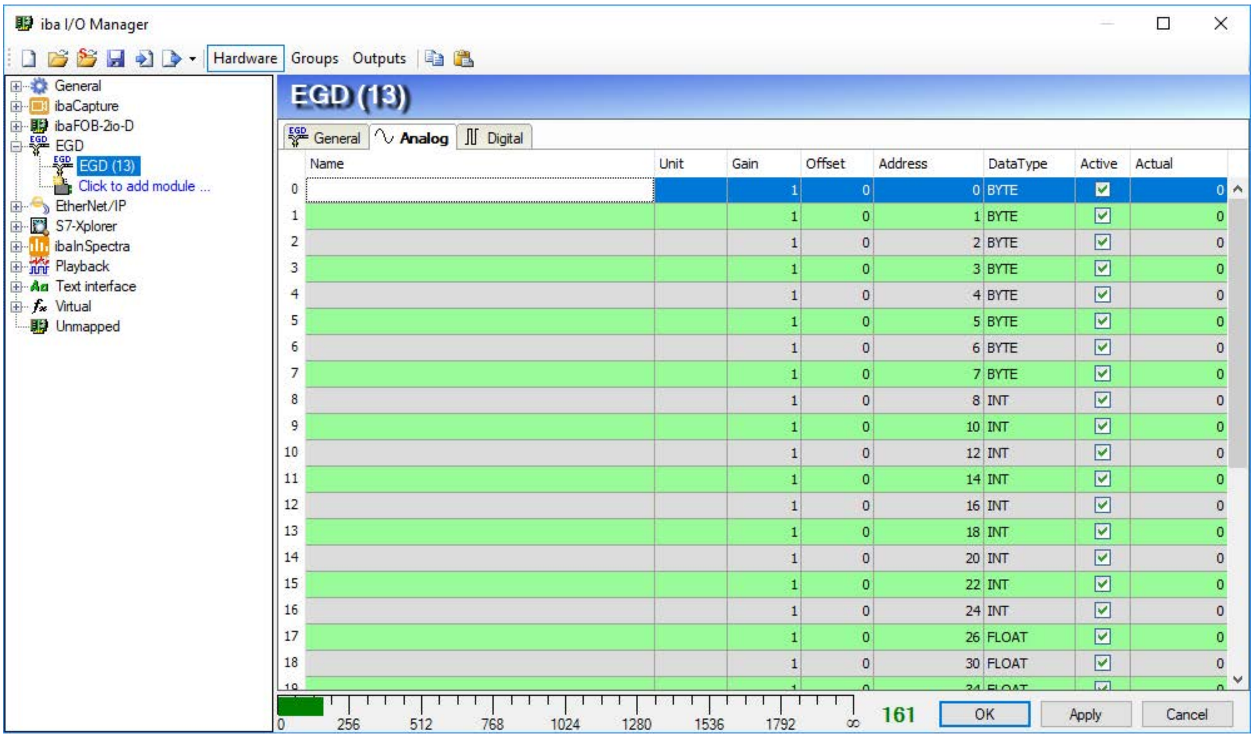
Module Layout

No. analog/digital signals

Defines the number of analog/digital signals in the signal table. The default value is 32 for each. The maximum value is 1000.

4.2.1.1 Analog tab

On the tab *Analog* you can define the analog signals in your exchange:



	Name	Unit	Gain	Offset	Address	DataType	Active	Actual
0			1	0	0	BYTE	☑	0
1			1	0	1	BYTE	☑	0
2			1	0	2	BYTE	☑	0
3			1	0	3	BYTE	☑	0
4			1	0	4	BYTE	☑	0
5			1	0	5	BYTE	☑	0
6			1	0	6	BYTE	☑	0
7			1	0	7	BYTE	☑	0
8			1	0	8	INT	☑	0
9			1	0	10	INT	☑	0
10			1	0	12	INT	☑	0
11			1	0	14	INT	☑	0
12			1	0	16	INT	☑	0
13			1	0	18	INT	☑	0
14			1	0	20	INT	☑	0
15			1	0	22	INT	☑	0
16			1	0	24	INT	☑	0
17			1	0	26	FLOAT	☑	0
18			1	0	30	FLOAT	☑	0
19			1	0	34	FLOAT	☑	0

Fig. 4: Analog tab

Name

In the *Name* column enter a comprehensive name for each signal.

It is recommended to use an application-specific naming convention for a better clearness and comprehension, particularly with vast numbers of signals. The name may refer to a technological purpose, the module name or a special location in the plant, where the signal comes from.

The number of characters in the name is unlimited. The names of the signals are stored in the data file and can be visualized in *ibaAnalyzer*.

Tip

A useful feature is the automatic fill function: If you enter a signal name and click on the column header as long as the cursor is still in the name field then all empty fields below will be filled with that name. If the name is ending with a number you will get names with an increasing number like an index. You may use this function in any row of the table. Fields which already have names won't be overwritten.

Unit

Assignment of an physical unit (such as Ampere, Volt, etc.) for the signal. This entry can be up to 11 characters long and is regarded as a comment field only. It is always displayed in conjunction with a numerical display of the values.

Gain and Offset

The values for gain and offset describe a linear characteristic curve for scaling. If incoming values are given in physical units, gain and offset can be ignored, i.e. set gain = 1 and offset = 0.

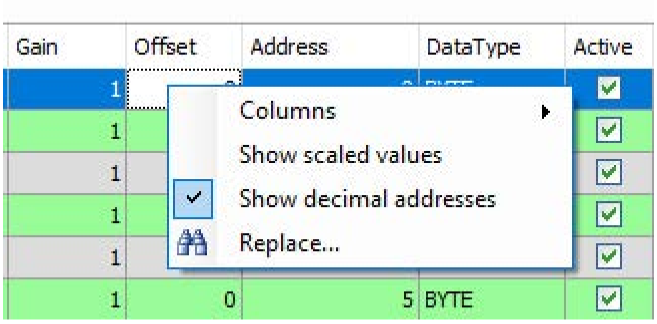
However, control applications in the automation systems which supply the signals often use normalized values for analog signals, ranging between 0.0 and 1.0 or -1.0 and +1.0. In order to get a correct scale for display in terms of physical units, *ibaPDA* must use a normalize factor. This factor can be evaluated by means of gain and offset parameters.

Gain and offset can be entered directly in the corresponding fields or by means of the two-point-scaling dialog with two pairs of applicable values.

You can open the two-point-scaling dialog with a click on the little tool button ⊞ in the fields gain or offset. (Cursor must be on the fields to see the button.)

Address

In the Address column, the offset of the first byte of this value within the raw data stream may be specified by the user. The offset can be entered as hexadecimal or decimal values by selecting the desired setting in the context menu.



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
- Digital signals (EGD module) increase of Bit no. by 1, from 0 to 31, then increase of address 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
In the fields of this column you can select the data type of each signal. Just click in the corresponding field and select the data type from the drop-down list.

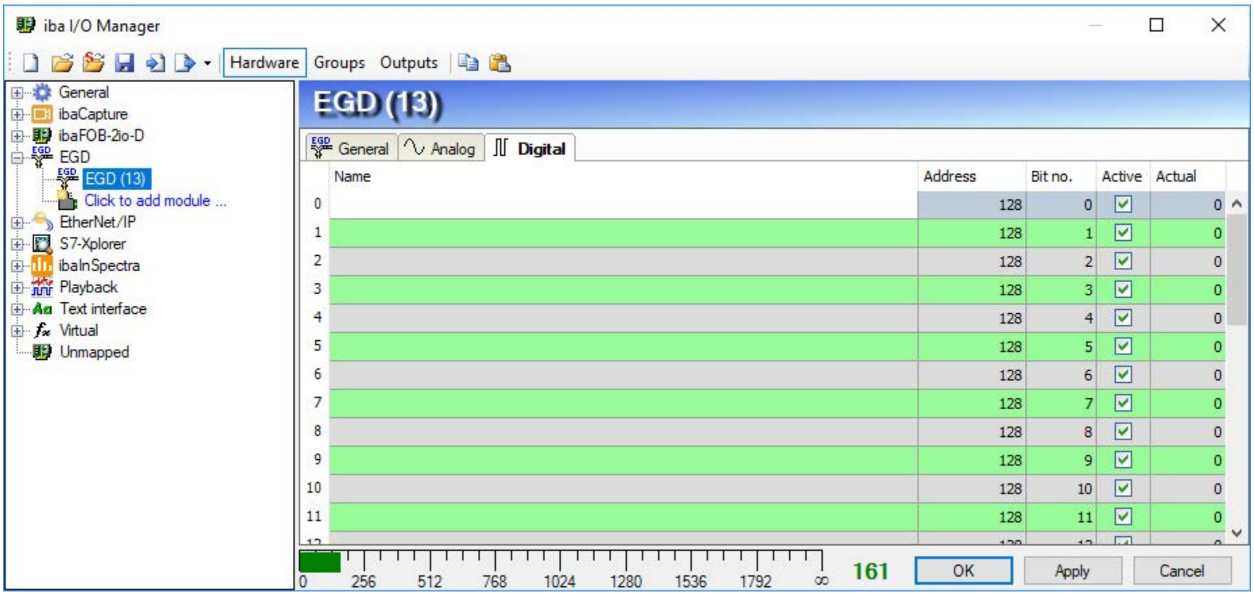
The address space is depending on the data type. Hence, an adjustment of address entries may be necessary after change of data types.

Active
Enable/disable the signal for acquisition in *ibaPDA*.

A click on the column heading "Active" enables (checkmark) and disables (no checkmark) all the signals at the same time. Individual signals can be activated in the signal-specific activation box. No acquisition takes place for channels which are not activated. Therefore these channels are available neither for display nor for storage.

Actual (Value)
The fields in this column show the actual value of the signals. Even if the acquisition is not running yet the actual value may be displayed if the hardware is already connected and working (diagnostic feature).

For digital signals only the values 0 and 1 are permitted.



	Name	Address	Bit no.	Active	Actual
0		128	0	☑	0
1		128	1	☑	0
2		128	2	☑	0
3		128	3	☑	0
4		128	4	☑	0
5		128	5	☑	0
6		128	6	☑	0
7		128	7	☑	0
8		128	8	☑	0
9		128	9	☑	0
10		128	10	☑	0
11		128	11	☑	0

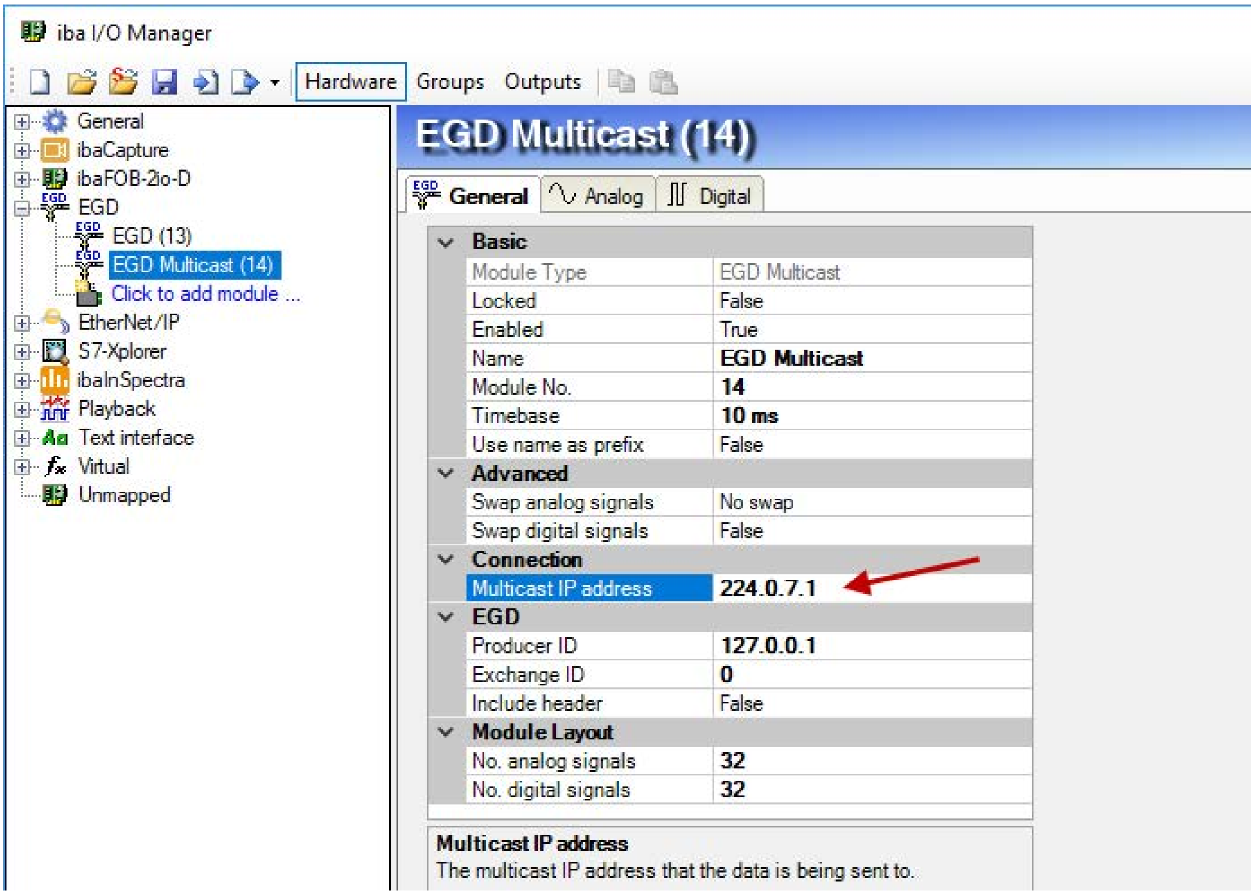
Fig. 5: Digital tab

On the digital tab you can define these options for the signals. For description see the figure above.

4.2.2 EGD multicast module

This type of module should be created manually.

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	False
Enabled	True
Name	**EGD Multicast**
Module No.	**14**
Timebase	**10 ms**
Use 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 that the data is being sent to.

Fig. 6: EGD multicast module, General tab

5 Troubleshooting and additional information

5.1 Data message layout

5.1.1 Data Production

The “Data_Production” PDU supports the production of Global Ethernet Data. This packet contains the user data being produced and sent to the consuming node(s). 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.

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

Table 2: Data Production PDU Format

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
Timestamp	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 timestamp is not synchronized on producer node
Configuration Signature	2	2 byte integer indicating relative version of the user data contained in the sample (see section 2.5).

PDU field	Size (Bytes)	Description
Production Data	Up to 1400	User data sample matching configuration specified in Configuration Signature

Table 3: Data Production specific fields

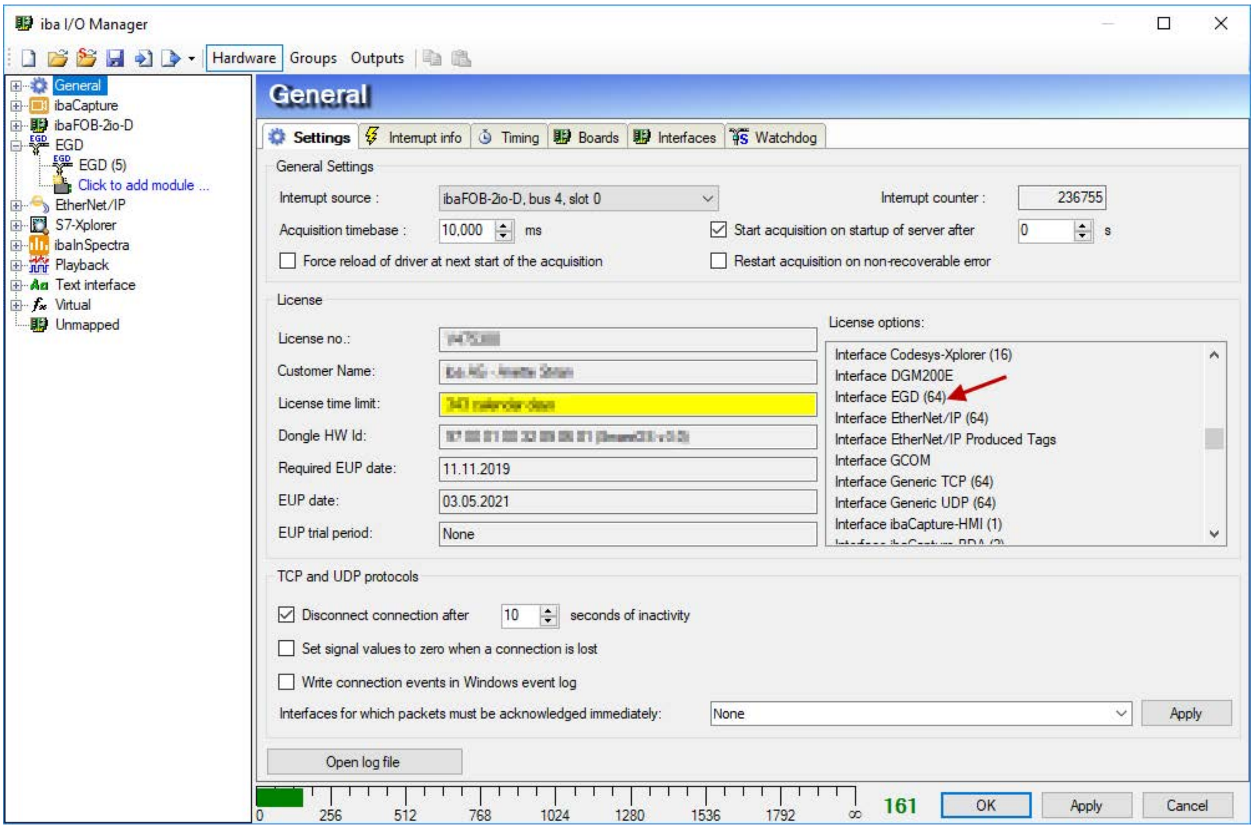
5.1.2 C-style data message declaration

```
#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 sec-
onds

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 packet

typedef struct _EGD_DATA_PACKET {

EGD_DATA_HEADER header;

unsigned char ProductionData[EGD_MAX_PRODUCTION_DATA];

} EGD_DATA_PACKET,*PEGD_DATA_PACKET;

#pragma pack (pop)
```

6 Diagnostics

6.1 License

If the "EGD" interface is not displayed in the signal tree, you can either check in *ibaPDA* under General - Settings- License info in the I/O Manager or in the *ibaPDA* service status application, whether your license "Interface EGD" has been properly recognized. The number of licensed connections is indicated in brackets.



General Settings

Interrupt source : ibaFOB-2io-D, bus 4, slot 0

Interrupt counter : 236755

Acquisition timebase : 10,000 ms

☑ Start acquisition on startup of server after 0 s

☐ Force reload of driver at next start of the acquisition

☐ Restart acquisition on non-recoverable error

License

License no.: [redacted]

Customer Name: [redacted]

License time limit: [redacted]

Dongle HW Id: [redacted]

Required EUP date: 11.11.2019

EUP date: 03.05.2021

EUP trial period: None

License options:

- Interface Codesys-Xplorer (16)
- Interface DGM200E
- Interface EGD (64)
- Interface EtherNet/IP (64)
- Interface EtherNet/IP Produced Tags
- Interface GCOM
- Interface Generic TCP (64)
- Interface Generic UDP (64)
- Interface ibaCapture-HMI (1)

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

Apply

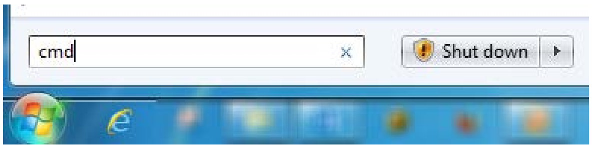
Open log file

Fig. 7: License display in the ibaPDA I/O Manager

6.2 Connection diagnostics with PING

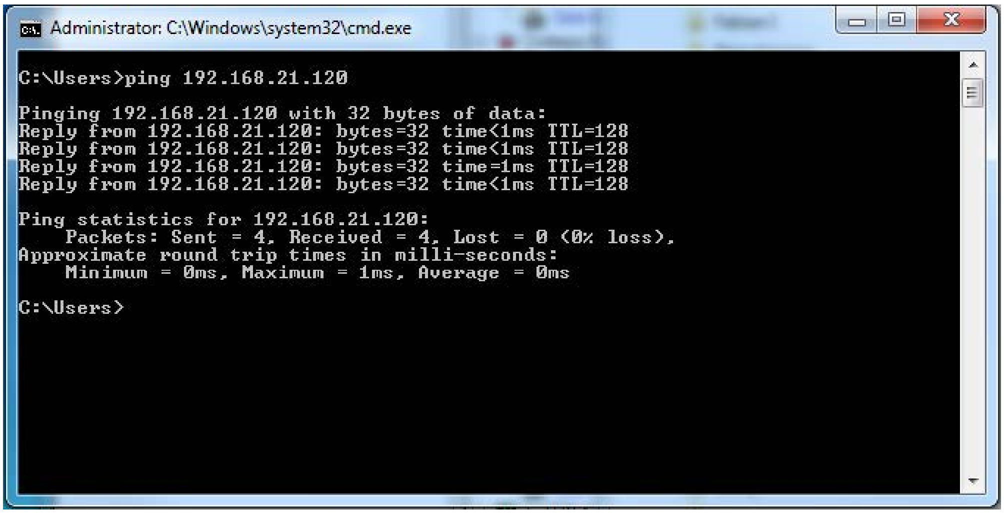
PING is a system command with which you can check if a certain communication partner can be reached in an IP network.

Open a Windows command prompt.



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

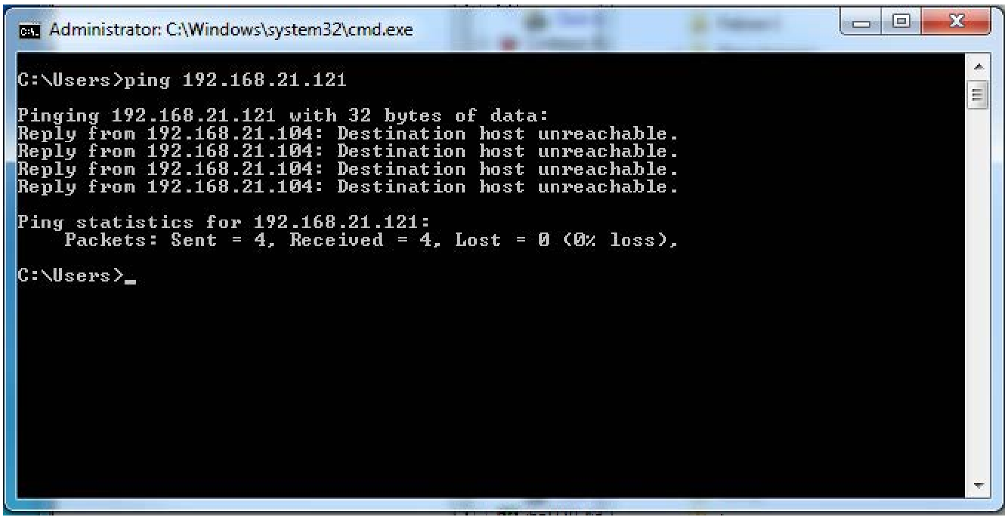
With an existing connection you receive several replies.



```
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. 8: PING successful

With no existing connection you receive error messages.



```
C:\Users>ping 192.168.21.121

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

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

C:\Users>_
```

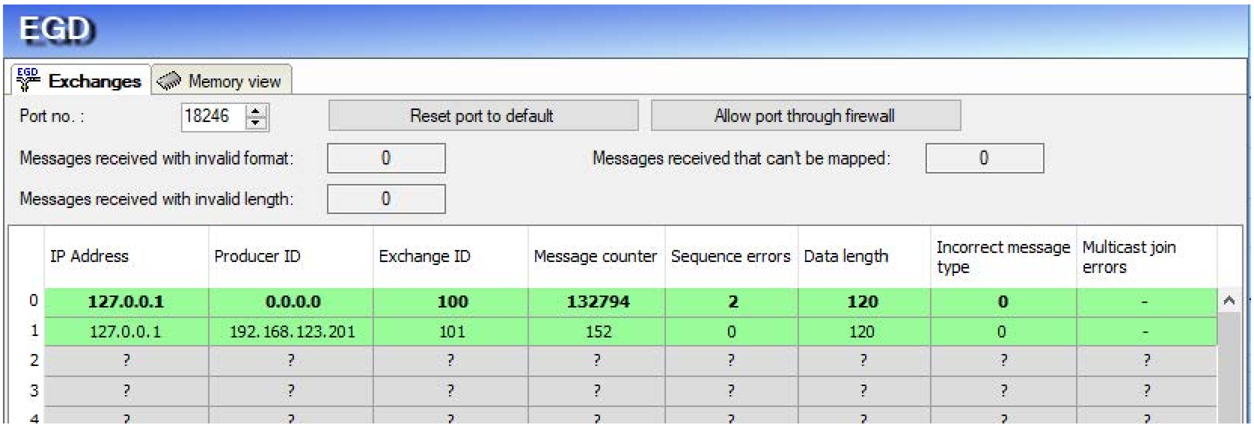
Fig. 9: PING unsuccessful

6.3 Connection table

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.

On the EGD interface examine the table of the different exchanges.



Port no. : 18246

Messages received with invalid format: 0

Messages received that can't be mapped: 0

Messages received with invalid length: 0

	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	?	?	?	?	?	?	?	?
4	?	?	?	?	?	?	?	?

Fig. 10: Connection table

On top of the grid which contains 64 possible exchanges, 3 global error counters are shown:

(See data message layout for explanation of the different data message fields)

Messages received with invalid format
Increments on the reception of a message

- where PDU_type not equal to 13 or
- PVN not equal to 1 or
- Request ID is not in valid range.

Messages received with invalid length
Increments on the reception of a message where the total length

- is shorter than the data message header (32 bytes) or
- longer then the maximum data message length (1432 bytes).

Messages received that cannot be mapped
Increments on the reception of a data sample of a new exchange but all 64 available exchange entries are used.

The active (connected) exchanges are marked with a green background color. In the table you can see either the IP address of the producer or the Multicast address, the Producer ID and the Exchange ID.

The “Message counter” is incremented on every data sample received. This message counter is also used to control if the producer of this exchange is still alive.

The “Sequence errors” counter is incremented when the Request ID of the previous data sample incremented by 1 does not match the Request ID of the newly received data sample. This can indicate that a data sample is lost.

The “Data length” column indicates the length of the received data sample.

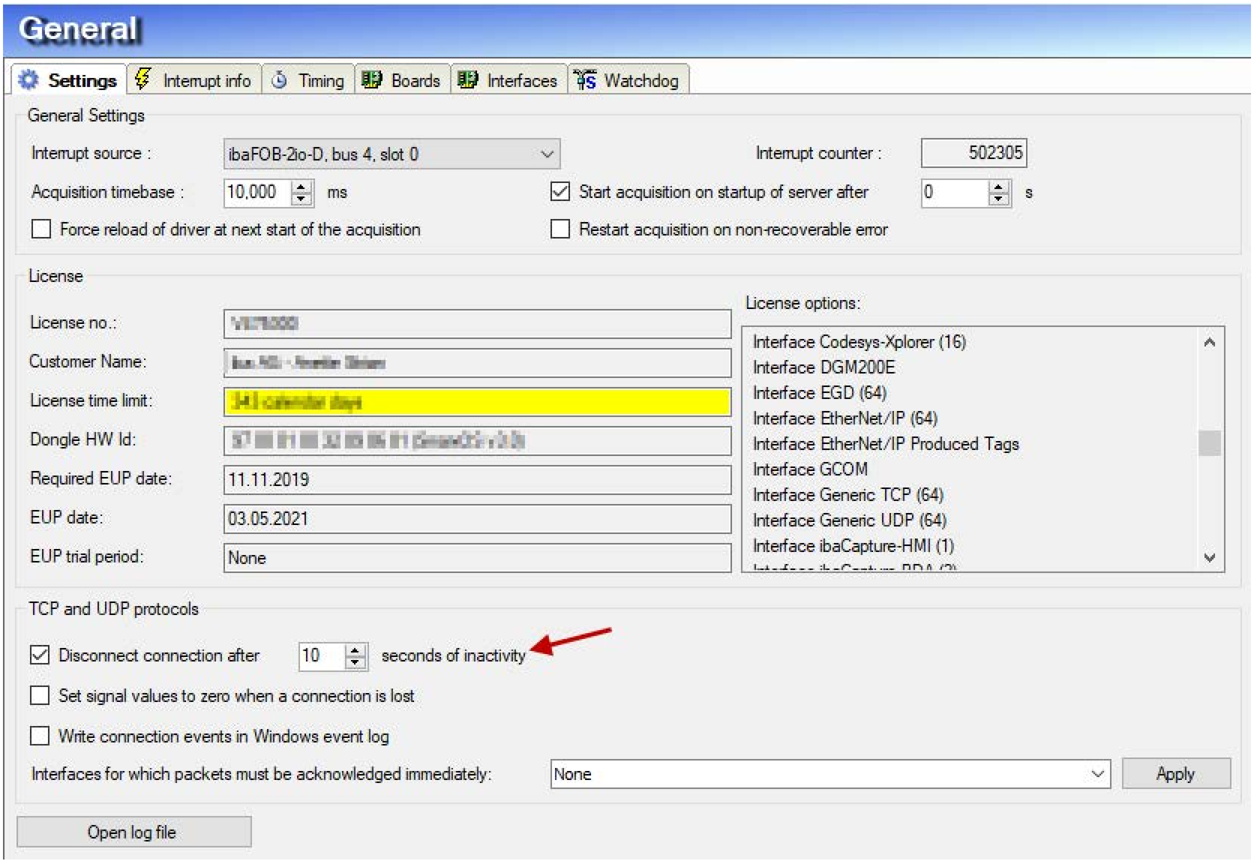
The “Incorrect message type” counter will be counting if

- unicast messages are received on an exchange that is configured for multicast or
- when multicast messages are received on an exchange that is configured for unicast.

The “Multicast join errors” counter will count each time ibaPDA fails to join the multicast group. It will have a ’-‘ symbol when this exchange is a unicast exchange.

Timeout setting

In the I/O manager there is an option to wipe the dead exchanges. See figure below.



General

Settings | Interrupt info | Timing | Boards | Interfaces | Watchdog

General Settings

Interrupt source : ibaFOB-2io-D, bus 4, slot 0

Interrupt counter : 502305

Acquisition timebase : 10,000 ms

☑ Start acquisition on startup of server after 0 s

□ Force reload of driver at next start of the acquisition

□ Restart acquisition on non-recoverable error

License

License no.: [redacted]

Customer Name: [redacted]

License time limit: [redacted]

Dongle HW Id: [redacted]

Required EUP date: 11.11.2019

EUP date: 03.05.2021

EUP trial period: None

License options:

Interface Codesys-Xplorer (16)
Interface DGM200E
Interface EGD (64)
Interface EtherNet/IP (64)
Interface EtherNet/IP Produced Tags
Interface GCOM
Interface Generic TCP (64)
Interface Generic UDP (64)
Interface ibaCapture-HMI (1)

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

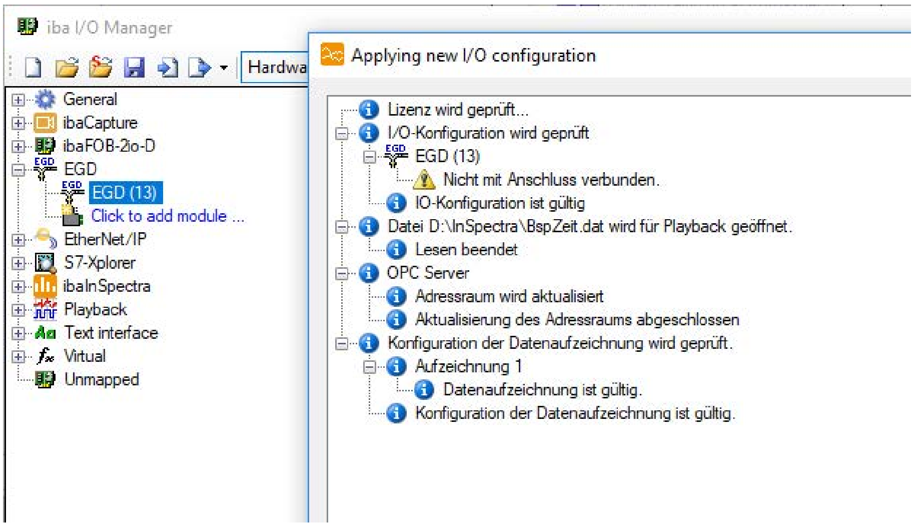
Fig. 11: Timeout setting

Disconnect TCP/IP connection 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:

- Green background and non-bold text: the exchange is running but there is no EGD module assigned to it.
- Green background and bold text: the exchange is running and there is an EGD module assigned to it. The bold text means 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.
- Grey background and bold text: the exchange is not running but locked since 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.



```
Lizenz wird geprüft...
I/O-Konfiguration wird geprüft
    EGD (13)
        Nicht mit Anschluss verbunden.
    IO-Konfiguration ist gültig
Datei D:\InSpectra\BspZeit.dat wird für Playback geöffnet.
    Lesen beendet
OPC Server
    Adressraum wird aktualisiert
    Aktualisierung des Adressraums abgeschlossen
Konfiguration der Datenaufzeichnung wird geprüft.
    Aufzeichnung 1
        Datenaufzeichnung ist gültig.
    Konfiguration der Datenaufzeichnung ist gültig.
```

Fig. 12: Configuration validation

6.4 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, since 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.

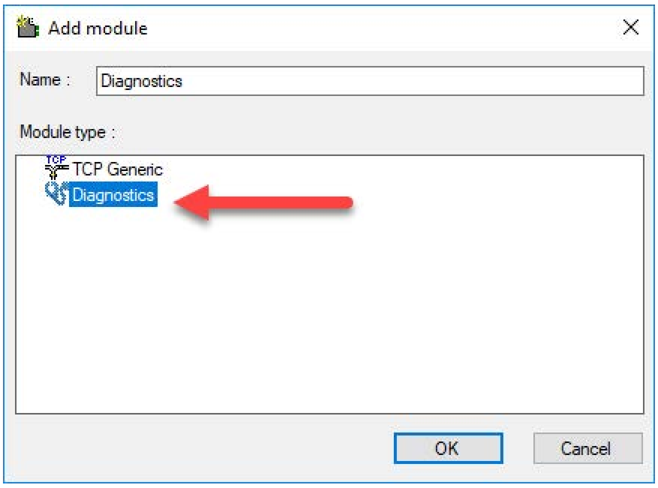
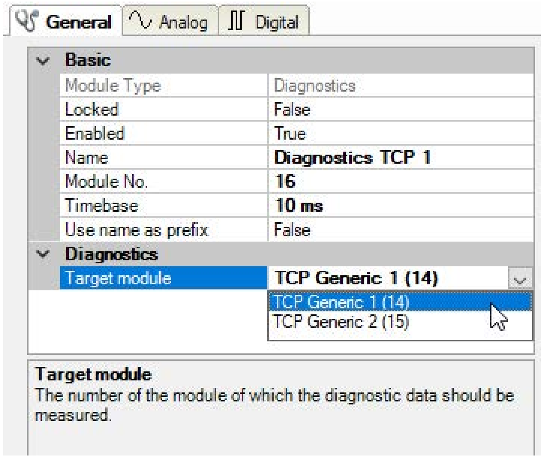


Fig. 13: Add diagnostic module, example Generic TCP

Module settings diagnostic module

For a diagnostic module, you can make the following settings:



Basic	
Module Type	Diagnostics
Locked	False
Enabled	True
Name	**Diagnostics TCP 1**
Module No.	**16**
Timebase	**10 ms**
Use name as prefix	False
Diagnostics	
Target module	**TCP Generic 1 (14)**
	TCP Generic 1 (14) TCP Generic 2 (15)

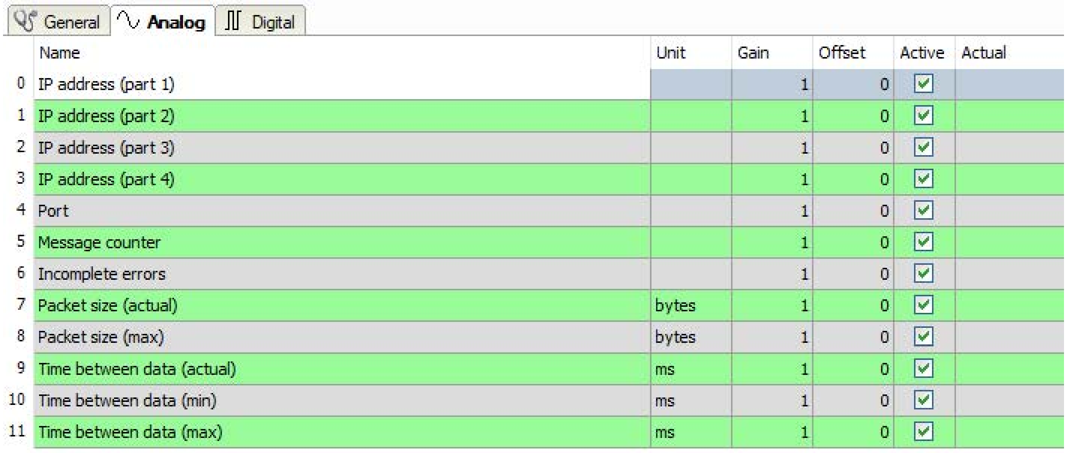
Target module
The number of the module of which the diagnostic data should be measured.

Fig. 14: Module settings diagnostic module, example TCP Generic

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.

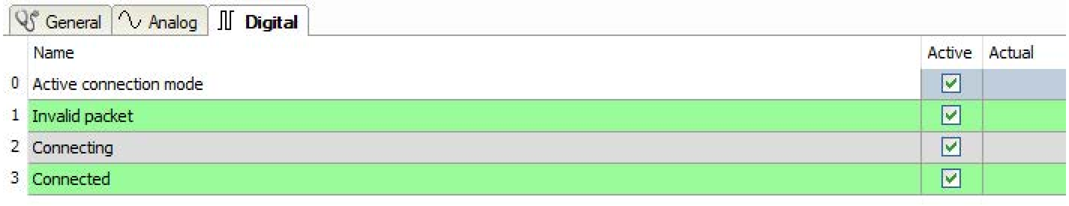
By selecting the target module, you assign the diagnostic module to the module on which you want to acquire information about the connection. You can select the supported modules of this interface in the drop down list of the setting. You can assign exactly one data acquisition module to each diagnostic 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.



	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	☑	
11	Time between data (max)	ms	1	0	☑	

Fig. 15: Example: Analog values of a diagnostic module for a TCP Generic module

For example, the IP (v4-) address of a TCP Generic 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.



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

Fig. 16: Example: Digital values of a diagnostic module for a TCP Generic module

7 Support and contact

Support

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

Note

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

Contact

Headquarters

iba AG
Koenigswarterstrasse 44
90762 Fuerth
Germany

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

Mailing address

iba AG
Postbox 1828
D-90708 Fuerth, Germany

Delivery address

iba AG
Gebhardtstrasse 10
90762 Fuerth, Germany

Regional and Worldwide

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

www.iba-ag.com.