ibaPDA-Interface-GCOM

Measurement Data Acquisition via ABB GCOM

Manual

Issue 3.0

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

Version	Date	Revision	Author	Version SW
3.0	08-2024	GUI new	rm, mm	8.8.0

Contents

1 About this documentation

This documentation describes the function and application of the software interface *ibaPDA-Interface-GCOM*.

Other documentation

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.

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-Interface-GCOM* the following basic knowledge is required and/or useful:

- Windows operating system
- Basic knowledge of *ibaPDA*
- Knowledge of configuration and operation of the relevant measuring device/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 to use the GCOM data interface:

- *ibaPDA* v6.36.1 or higher
- Base license for *ibaPDA* + license for *ibaPDA-Interface-GCOM*
- A free Ethernet network interface, ideally only for GCOM network

Note

It is recommended to carry out the GCOM communication on a separate network.
Another network interface map may be required to avoid influencing the GCOM telegrams by the Ethernet data traffic between *ibaPDA* and other nodes in the network (file server, measurement file requests, etc.).

For further requirements for the used computer hardware and the supported operating systems, refer to the *ibaPDA* documentation.

License information

Order no.	Product name	Description
31.001080	ibaPDA-Interface-GCOM	Extension license for an *ibaPDA* system for connecting to an ABB master system via GCOM

Table 1: Available GCOM interface licenses

3 GCOM Interface

3.1 General information

iba has developed an NDIS driver for implementing the ABB GCOM protocol.

The GCOM Multidrop bus (GCOM subnet) is a serial half-duplex bus for synchronous data transmission via medium communication distances.

Due to the three-layer structure, the multidrop bus connection and the segmentation that is not necessary, GCOM can easily run on external computers.

Application tasks	Application
GCOM Flow Control	Transport
LLC IEEE 802.3	Data Link
MAC IEEE 802.3	
PHY IEEE 802.3	Physical

Table 2: Three-layer communication model

The protocol is executed as a data link user on an IEEE 802.3 standard bus. The multidrop bus has a connection-free, unconfirmed data link service according to IEEE 802.2 class 1. The bus does not have a specific master, i.e. all stations have the same access to the bus.

Due to the sequence control and the handling of the retransmission in the GCOM layer, the GCOM subnet has a high level of security.

The GCOM subnet does not support broadcasting.

On the ABB master side, the GCOM communication software runs on a microprocessor-based communication board (DSCS 150 in MG 230/1 and MP 200/1, SC530 in AC 450).

The GCOM subnet (bus) supports up to 4 external computers on the same bus. The transmission rate is up to 10 Mbit/s.

GCOM Multidrop offers a powerful communication connection from external computers to the ABB master system (MP200, AC450).

ibaPDA supports up to 4 GCOM connections.

References

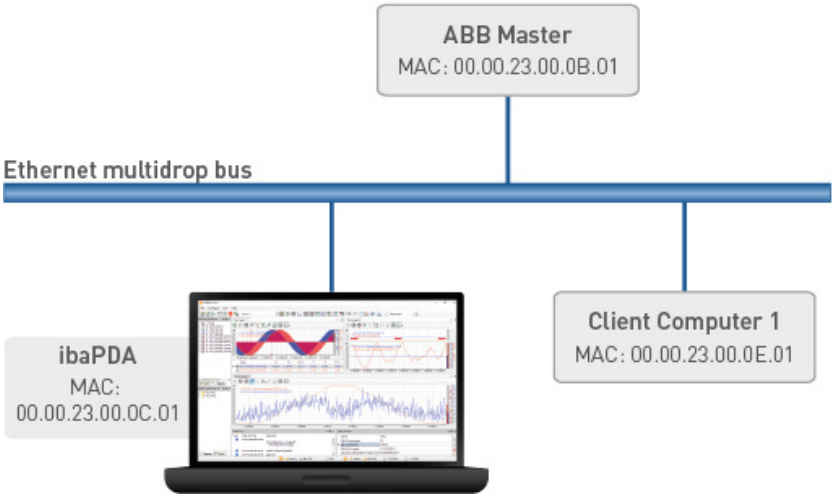
Other documentation

Additional documentation and further literature:

- ABB Master GCOM Multidrop User’s Guide 3BSE 000 165R0001
- ABB Stressometer Flatness Logger User’s Manual V4.0 3BSE 000 350R0801

3.2 System topologies

The following figure provides an overview of one of the possible configurations in which 2 computers are connected to the ABB master GCOM network.



3.3 ibaPDA-specific implementation

The individual messages from the ABB master to the *ibaPDA* system are sent using the Data Set Communication (DSC).

Other documentation

You can find additional details in the ABB Master GCOM Multidrop User's Guide, chapter 3.5.4 *Data Part Application Function*.

Since every message contains up to 24 values each of 32 bits, several data messages are sent for each logical message group. The *ibaPDA* driver buffers data messages until all components of a logical message group have arrived. Only then is the entire message transferred to the general GCOM buffer in the *ibaPDA* driver.

The general GCOM buffer has the following layout:

```
typedef struct _MEMGCOM
{
unsigned char GCOMMessages[256][100]; // 256 messages of 100 bytes
unsigned long GCOMMessageCounter[256]; // 256 message counters
unsigned long GCOMGroupCounter[256]; // 256 message group counters
unsigned long GCOMOwnMacMessageCounter;
unsigned long GCOMIAmHereMessageCounter;
unsigned long GCOMNotOwnMacMessageCounter;
unsigned long SendCounterAck;
unsigned long SendCounterIAmHere;
} MEMGCOM, *PMEMGCOM;
```

The buffer reserves 100 bytes of storage capacity for the last message of a possible message ID (according to the GCOM manual, the highest message ID value is 255). If you use logical message groups, the last message is only updated when all components of the message have been received. Note that, due to their layout, the logically grouped messages are not always stored in a contiguous buffer block (e.g. if a logical message group consists of messages 1, 3 and 5).

The buffer block is followed by an array of 256 32-bit message counters; one for each message ID. The message counters count up regardless of the logical message grouping.

Then there is another array of 256 32-bit values for the logical message group counters.

In addition, the following diagnose counters are available, which are generated by the *ibaPDA* driver:

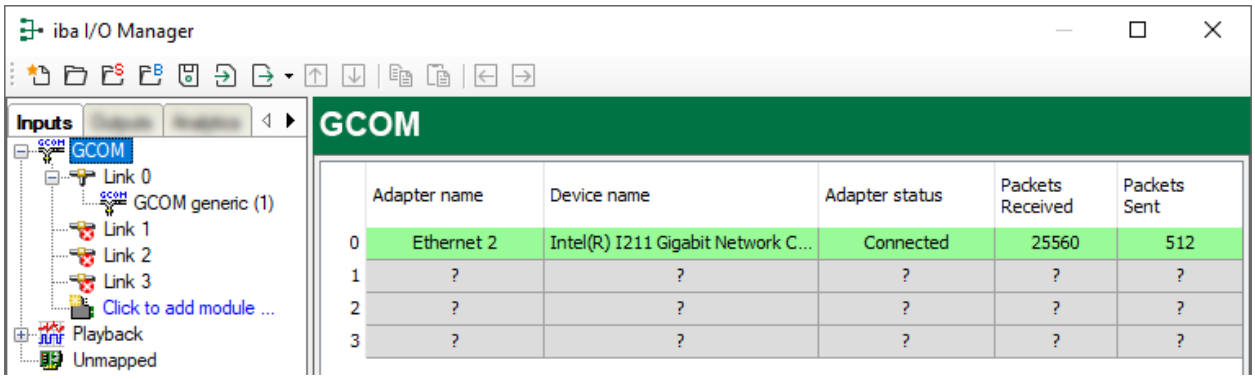
- GCOMOwnMacMessageCounter
  Number of data messages that are intended for the configured MAC address.
- GCOMIAmHereMessageCounter
  Number of received "I am here" messages (both the messages sent from *ibaPDA* in active mode as well as the messages that are received at other network nodes).
- GCOMNotOwnMacMessageCounter
  Unrecognized packets that come from a MAC address that is not the own.
- SendCounterAck
  ACK messages that are sent in active mode.
- SendCounterIAmHere
  "I am here" messages that are sent in active mode.

3.4 Configuration and engineering ibaPDA

The engineering for *ibaPDA* is described in the following. If all system requirements are fulfilled, *ibaPDA* displays the *GCOM* interface in the interface tree of the I/O Manager.

3.4.1 General interface settings

If you select the data interface in the tree, you can see an overview of diagnostics information about the configured connections. In the respective connection node, you configure the individual connections.



3.4.2 Connection settings

Configure the GCOM connections in nodes *Connection 0* to *4*.

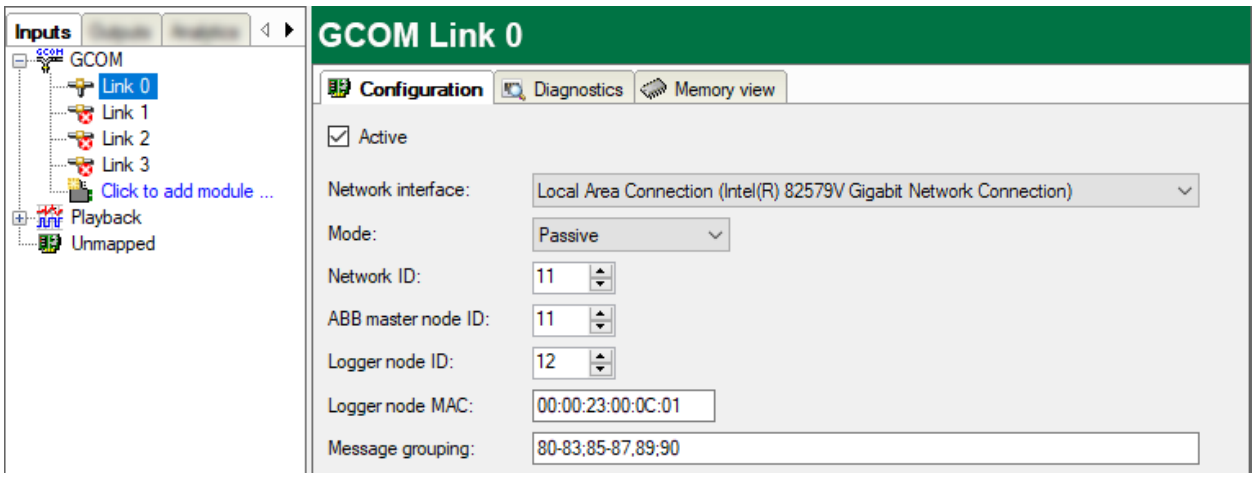
Select a connection node.

In the right window of the I/O Manager, you can find three tabs which are used for configuration and diagnostics of the connection.

Later, you can add the modules for signal acquisition to the connection.

3.4.2.1 Link – Configuration tab

Set the GCOM connection parameters in the *Configuration* tab.



Caution

Network disruptions possible

If you use the same network interface and settings for Network ID and Logger Node ID for different connections in active mode, this leads to unpredictable results and network disruptions due to the use of the same MAC address.

Use different settings for Network ID and Logger Node ID for different connections via the same network interface.

Active
Activate the connection to use it.

Network interface
Select the NIC (network interface card) for the GCOM connection. The drop-down list contains all of the registered network interfaces.

Mode
Select whether the connection runs in active or passive mode:

- *Active* mode: The *ibaPDA* system functions as an active GCOM network node and sends ACK messages as well as "I am here" messages on the network on a cyclical basis.
  For detailed information, see also the ABB Master GCOM Multidrop User's Guide, chapter 1.8.4 *Transit Handling* to 1.8.6 *Acknowledge and Sequence Handling*.
- *Passive* mode: The *ibaPDA* system behaves completely passively on the network. It collects data in this mode but does not send any messages in the network. In this mode, you can use *ibaPDA* in parallel with an existing flatness logger.

Network ID
Enter the ABB master subnet that is used for the communication with the ABB master system.

ABB master node ID
Enter the ABB master node that is used in your network.

Logger node ID
Specify the node ID of the *ibaPDA* system in the ABB master subnet.

Logger node MAC
Network address to which the data messages are sent from the ABB master. Note that this address may differ from the current MAC address of the NIC that is connected to the ABB network. The *ibaPDA* driver only evaluates messages that are intended for this MAC address.

Message grouping
Define the messages to be synchronized here. Use the following characters:

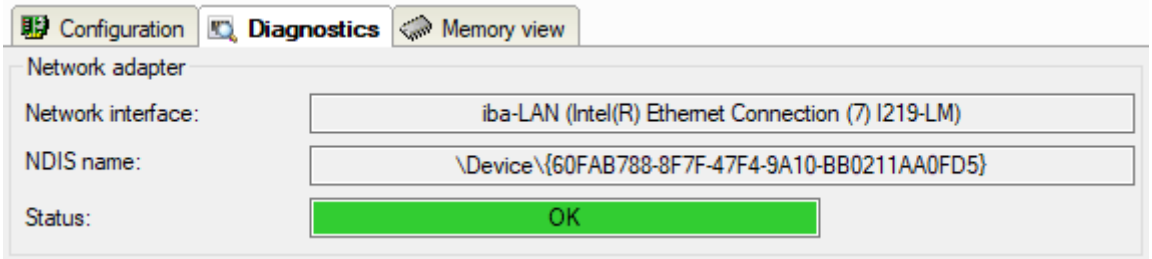
- **";"** to separate two groups
- **","** to separate two message IDs within a group

Define a range of message IDs in a group:
You can define a maximum of 256 groups, where each group can contain 32 messages.
Default setting: 80-83;85-87;89;90

3.4.2.2 Link – Diagnostic tab

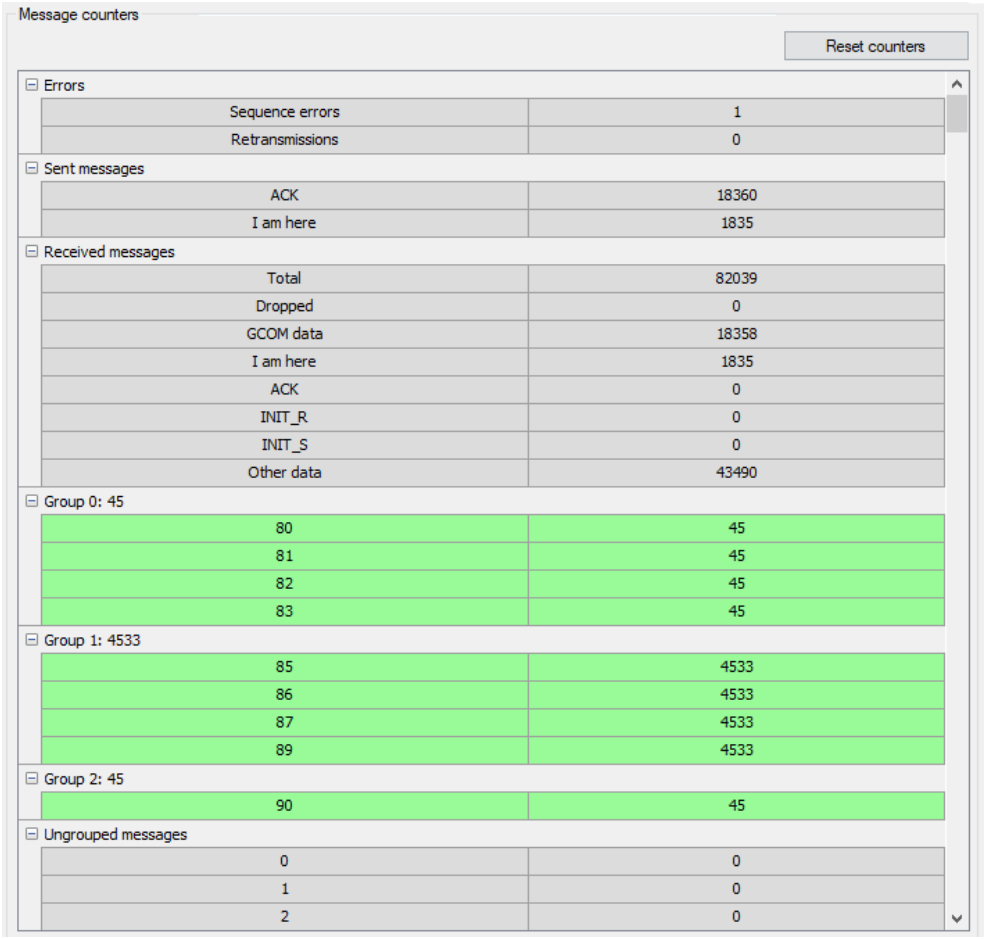
If you select an active connection, a variety of diagnostic information is available. This information is divided into two areas: *Network adapter* and *Message counter*.

Network adapter
This area shows the selected network interface, the associated NDIS name and the status of the connection.



Message counters

This area shows several counters for diagnostic purposes. The table contains three categories of counters. In addition, there is one further category per logical group, as defined in the *Configuration* tab of the connection, as well as a category for messages that do not belong to the logical message grouping.



For every logical message group, a category is created in the table containing all configured message IDs (left column), including the number of received messages (right column). The header of the respective category shows the number of completely received logical messages (i.e. the minimum of all message ID counters in the group).

Message IDs that do not belong to any logical group are listed in the category *Ungrouped messages*.

Error

- *Sequence errors*
  Total number of messages that arrive with a sequence error.
  An INIT_S message is sent to the ABB Master unit. The sequence counter for the received messages is reset and the signal for the received data is dropped.
- *Retransmissions*
  Total number of retransmitted messages received by *ibaPDA*. If a data message with the same sequence counter as the previous message is received, this is a data retransmission of the ABB Master. The *ibaPDA* system confirms the data message with the same sequence counter.

Sent messages

- *ACK*
  Total number of ACK messages, which are sent to the ABB Master in the *Active* mode. The counter remains on 0 in *Passive* mode.
- *I am here*
  Total number of "I am here" messages, which are sent from *ibaPDA* in the *Active* mode. These messages must be sent every second to establish a cross reference table between station addresses on the GCOM subnet and the logic addresses (network and node number) in the ABB Master system. The counter remains on 0 in *Passive* mode.

Received messages

- *Total*
  Total number of messages that were received on the selected network interface.
- *Dropped*
  Total number of messages that could not be processed due to internal problems (memory allocation).
- *GCOM data*
  Total number of received GCOM messages.
- *I am here*
  Total number of received "I am here" messages.
- *ACK*
  Total number of ACK messages received by the ABB Master. Normally, this counter remains on 0 because *ibaPDA* does not send any data messages to the ABB Master.
- *INIT_R*
  Total number of INIT_R messages received by the ABB Master. If an INIT_R message is received, the *ibaPDA* sequence counter for received messages must be reset to 0 in order to re-synchronize the sequence counter.
- *INIT_S*
  Total number of INIT_S messages received by the ABB Master. If an INIT_S message is received, the *ibaPDA* sequence counter for sent messages must be reset to 0 in order to re-synchronize the sequence counter. However, this is normally not necessary because *ibaPDA* does not send any data messages to the ABB Master.
- *Other data*
  Total number of received messages that are not GCOM messages.

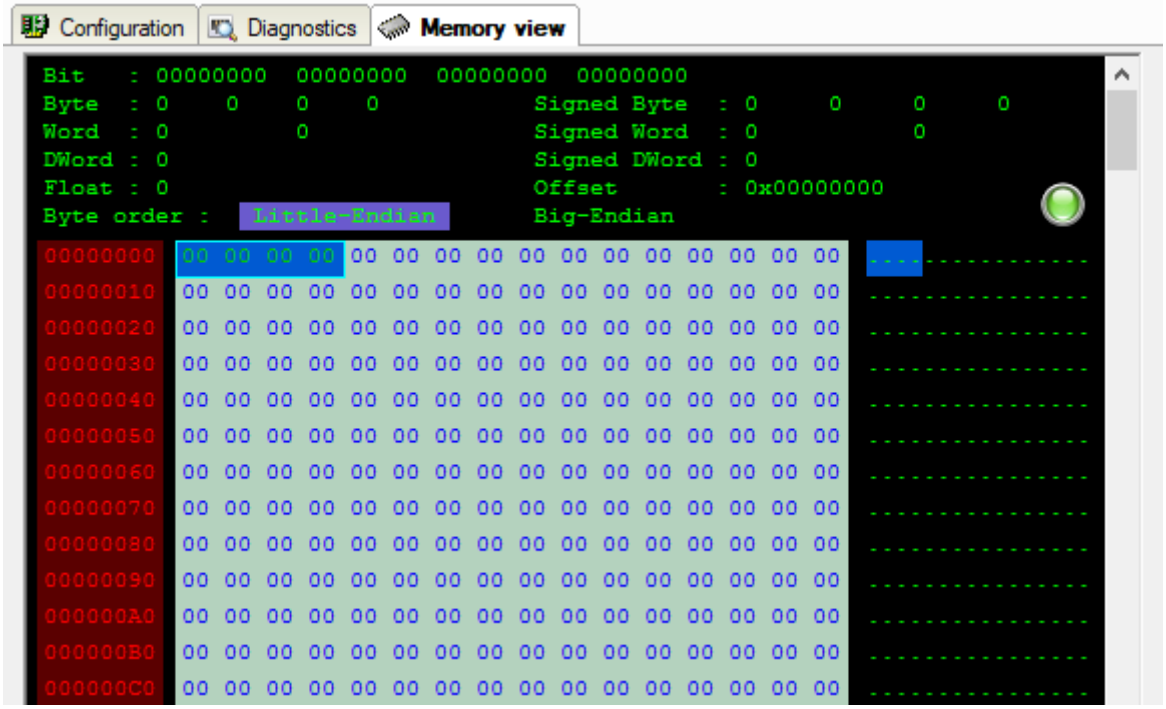
<Reset counters>
Click this button to reset the counters in the table to 0.

3.4.2.3 Link – Memory view tab

Select an active link and then the *Memory view* tab to view a hex viewer.

The viewer shows the memory region MEMGCOM where the recomposed data messages and other diagnostic information are stored.

For information about the layout of the MEMGCOM structure, see chapter ↗ *ibaPDA-specific implementation*, page 9.



Using the Memory view

With the appropriate experience, you can check the content of memory address cells.

1. Right-click with the mouse in the memory data display to open the context menu.
2. In the context menu, select *Go to offset* and enter a memory address that you want to check.
3. Click on <OK>.

→ The view jumps to the desired address (highlighted) and shows the content.

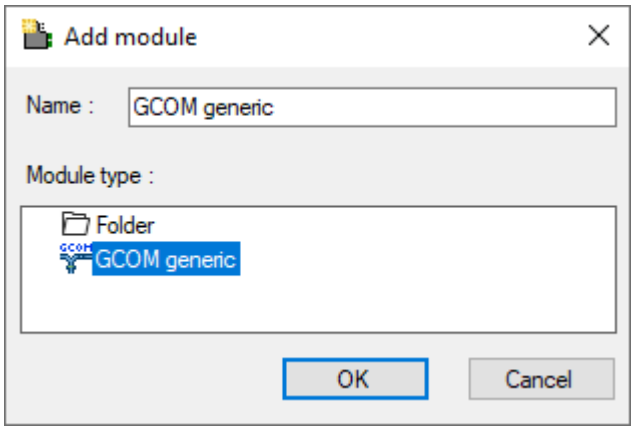
3.4.3 Adding a module

This chapter describes how to add a module to the GCOM interface. Detailed information on the module and the configuration can be found in the corresponding chapter:

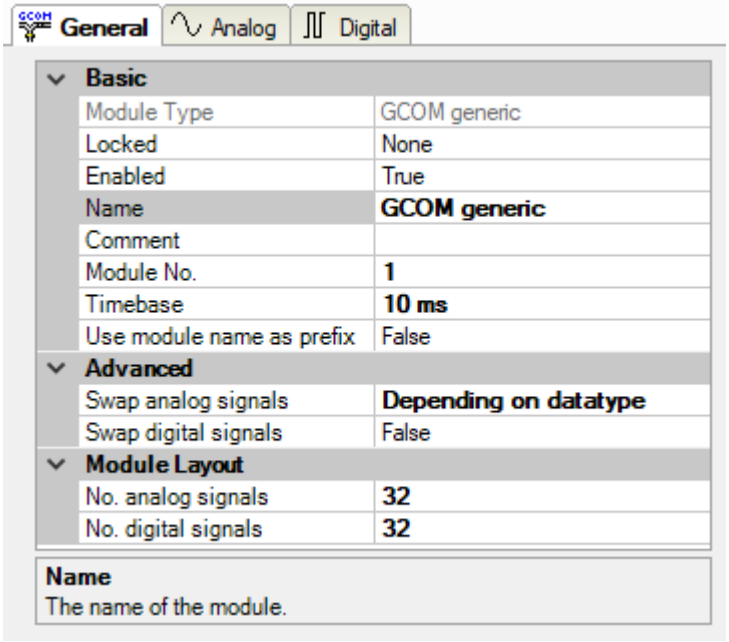
↗ *General module settings*, page 16

Procedure

1. Click on the blue link *Click to add module* located under each link of the data interface in the *Inputs* 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>.



3.4.4 General module settings



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.

Advanced

Swap analog signals
Set the swap mode according to the signal source.
You can select one of the following 4 options:

Mode	16 bit	32 bit
No swap	AB	ABCD
Depending on data type	BA	DCBA
Swap 16 bit	AB	CDAB
Swap 8 bit	BA	BADC

Table 3: Swap modes

The swap mode to be selected depends on the swap mode of the signal source.

Tip

Use the default setting *Depending on datatype*.

Swap digital signals
Select here whether to swap the digital signals on a 4-byte basis.

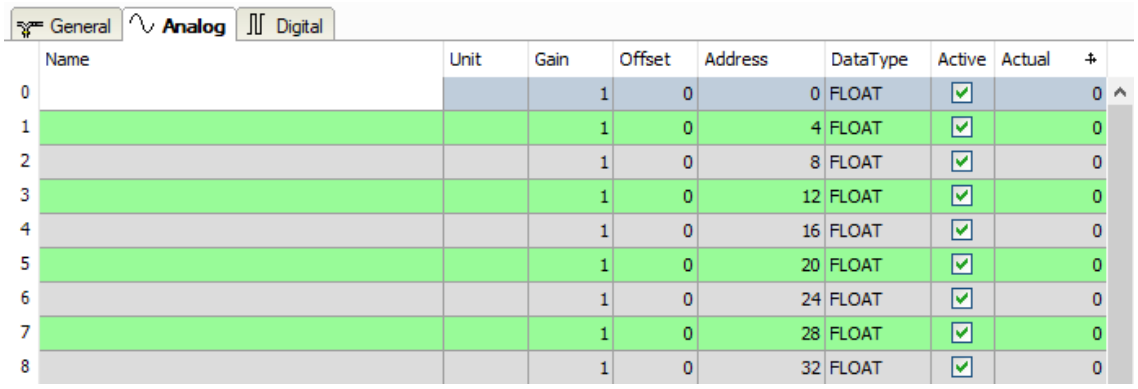
- False: no swap (default)
- True: byte order changes from ABCD to DCBA

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.

3.4.5 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	0	FLOAT	☑	0
1			1	0	4	FLOAT	☑	0
2			1	0	8	FLOAT	☑	0
3			1	0	12	FLOAT	☑	0
4			1	0	16	FLOAT	☑	0
5			1	0	20	FLOAT	☑	0
6			1	0	24	FLOAT	☑	0
7			1	0	28	FLOAT	☑	0
8			1	0	32	FLOAT	☑	0

Name
Enter a meaningful plain text name for the signal.

Unit (analog signals only)
Assignment of a physical unit for the signal

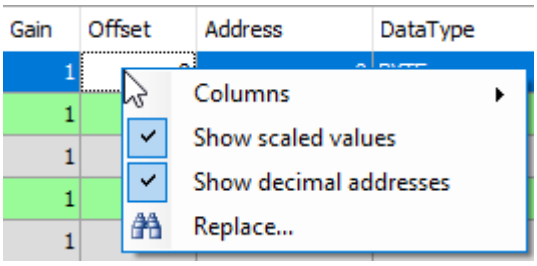
You can enter a maximum of 11 characters, the field is only considered a comment field. The unit is always displayed in conjunction with a numerical display of the values.

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

The values describe a linear characteristic curve for scaling. If incoming values are specified in physical units, you can ignore this function, i.e. Gain = 1 and Offset = 0.

Address
The address specifies the offset of the first byte of this value within the MEMGCOM structure.

You can enter the offset as hexadecimal or decimal value.



To enter the default values in the column, click on the column header. The offset values are automatically entered from the current position of the cursor or the first row in the table. The addresses are incremented depending on the data type. For digital signals, the *Bit no.* is automatically incremented.

- Analog signals as FLOAT, DINT or DWORD: 4-byte-steps
- Analog signals as INT or WORD: 2-byte-steps
- Analog signals 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.

The addresses of messages are calculated from the message ID:
Address = message ID * 100.
Example: Message ID 81 is from address 8100.

If you enter all signal definitions with name and data type and click on *Address*, *ibaPDA* automatically calculates 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 data type determines the address of the next signal.

Available data types:

Data type	Description	Value range
BYTE	8 bit without positive or negative sign	0 … 255
INT	16 bit with positive or negative sign	-32768 … 32767
WORD	16 bit without positive or negative sign	0 … 65535
DINT	32 bit with positive or negative sign	-2147483648 … 2147483647
DWORD	32 bit without positive or negative sign	0 … 4294967295
FLOAT	IEEE754; single precision; 32 bit floating point value	1.175·10-38 … 3.403·1038
DOUBLE	IEEE754; double precision; 64 bit floating point;	2.225E-308 … 1.798E+308
FP_REAL	Fixed point real; Q15.16; 15 integer bits and 16 fractional bits;	-32768 … 32767.9999

Active
Activation or deactivation of the respective signal

Actual
Display of the current actual value of the signal

Other documentation

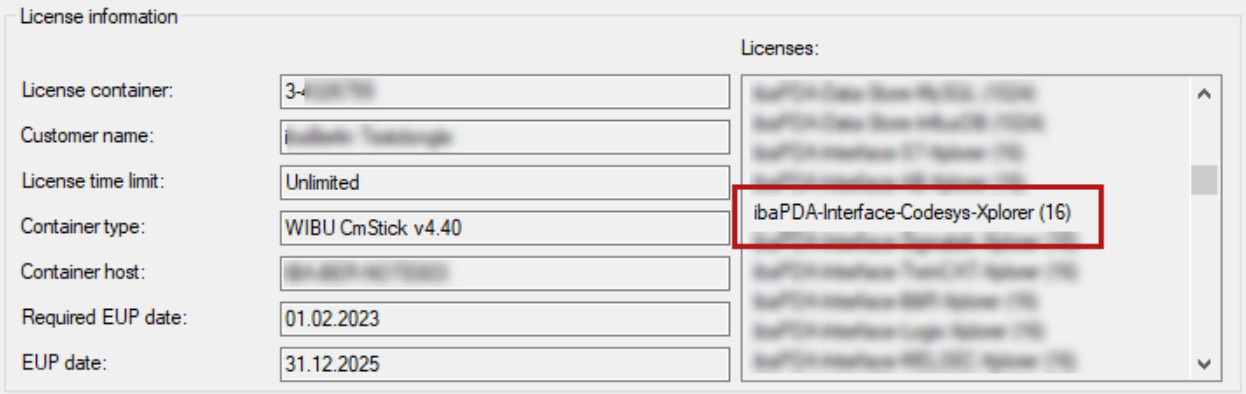
Detailed descriptions of the columns and how to fill in the signal tables can be found in the documentation for *ibaPDA*.

4 Diagnostics

4.1 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 the interface *ibaPDA-Interface-GCOM* 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.



4.2 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.

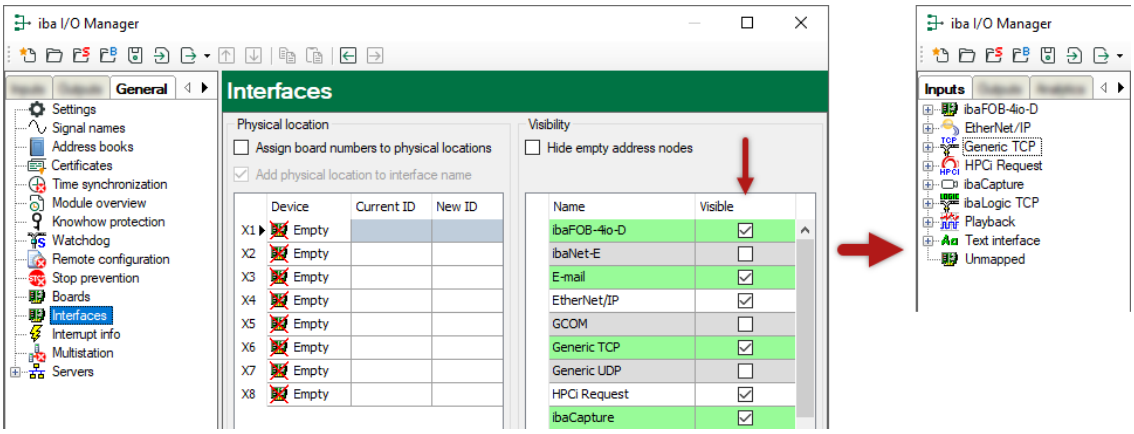
Visibility

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:



4.3 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.

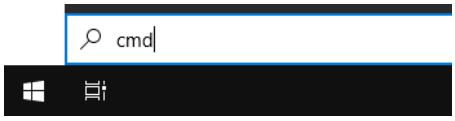
Examples:

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

4.4 Connection diagnostics with PING

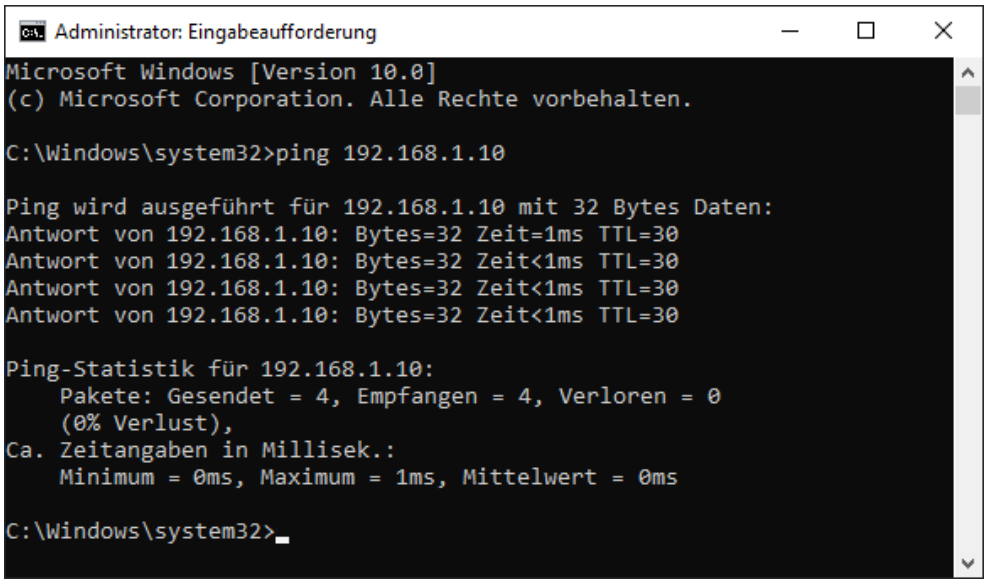
PING is a system command with which you can check if a certain communication partner can be reached in an IP network.

1. Open a Windows command prompt.

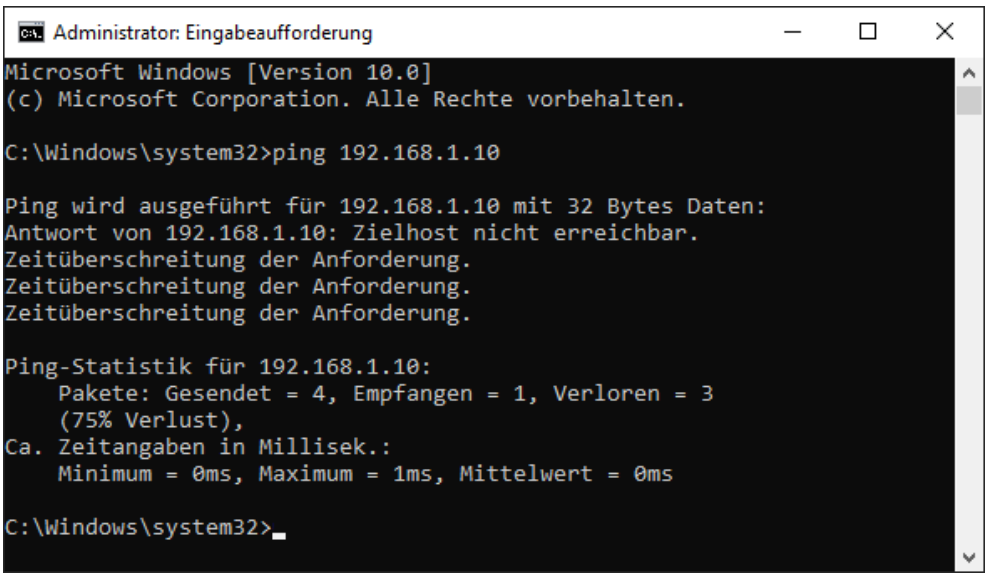


2. Enter the command "ping" followed by the IP address of the communication partner and press <ENTER>.

→ With an existing connection you receive several replies.



→ With no existing connection you receive error messages.



5 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