ibaPDA-Interface-LMI-Gocator

Data interface for LMI-Gocator sensors

Manual

Issue 2.1

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
2.1	05-2024	new version ibaPDA v8.7, new module "LMI-Go-nmcator Health"		8.7.0

Contents

1 About this documentation

This documentation describes the function and application of the software interface

ibaPDA-Interface-LMI-Gocator.

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

For the handling of *ibaPDA-Interface-LMI-Gocator* the following basic knowledge is required and/or useful:

- Windows operating system
- Basic knowledge of *ibaPDA*
- Knowledge of configuration and operation of the relevant LMI-Gocator sensors

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 LMI-Gocator data interface:

- *ibaPDA* v8.0.0 or higher
- License for *ibaPDA-Interface-LMI-Gocator* (supports up to 2 sensors, i. e. 2 connections)
- If you need more than 2 connections, you will require additional *one-step-up-Interface-LMIGocator* licenses for each additional 2 connections. The total limit is 16 connections.

Also Gocator sensors used in a buddy mode setup require a license.

A minimum firmware version "Gocator Release 4.6 SR1" (v4.6.7.17) is required. Sensors of older series available only with firmware versions < v4.6 are not supported (e. g. Gocator 1100, 2000 series).

Other documentation

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

Note

The 2D top view is particularly suitable for displaying the measured values. This display is possible with live data but only with the trend graph and HD trend graph objects of the *ibaQPanel* software. Therefore it is recommended to purchase additional licenses for *ibaQPanel* and/or *ibaHD-Server*.

In the offline analysis with *ibaAnalyzer*, the 2D top view is included as standard.

License information

Order no.	Product name	Description
31.001012	ibaPDA-Interface-LMI-Gocator	ibaPDA data interface for connecting up to 2 Gocator sensors
31.101012	one-step-up-Interface-LMI-Gocator	Extension license for 2 more Gocator sensor connections; a maximum of 7 permissible
30.670040	ibaQPanel-Add-on	Additional package for an ibaPDA client to display process/quality data in an HMI image

3 LMI-Gocator interface

3.1 General information

The LMI-Gocator interface can be used to measure profile data from Gocator® sensors (LMI Technologies Inc.). Data from several, adjacent sensors can be collected and merged into a single profile.

3.2 System topologies

The connections between the sensors and *ibaPDA* can be established via standard Ethernet interfaces of the computer.

No further software is required for operation.

Note

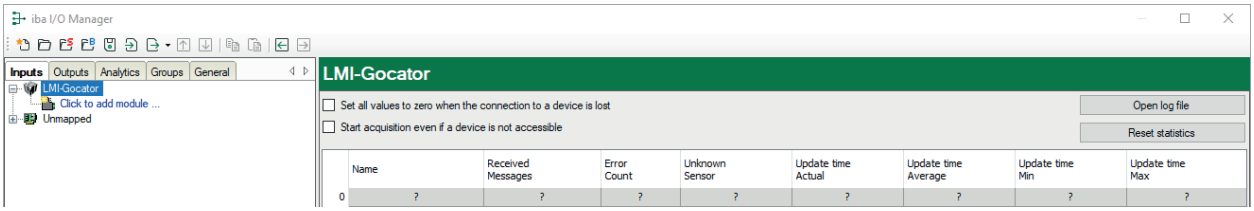
iba AG recommends carrying out TCP/IP communication on a separate network segment in order to prevent mutual interference from other network components.

3.3 Configuration and engineering in ibaPDA

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

3.3.1 Interface settings

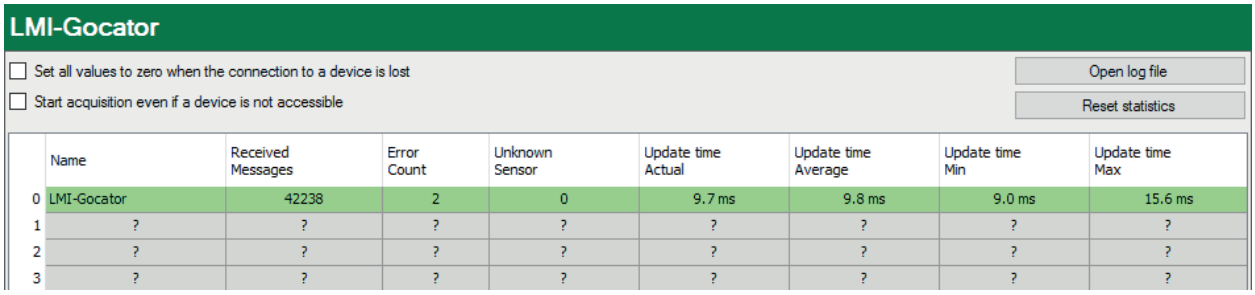
The interface itself has the following functions and configuration possibilities:



Set all values to zero when the connection to a device is lost
If this option is activated, all measured values of the sensor are set to zero as soon as the connection is lost. If this option is disabled, *ibaPDA* will keep the last valid measured value at the time the connection was lost in the memory.

Start acquisition even if a device is not accessible
If this option is enabled, the acquisition will start even if the sensor is not accessible. Instead of an error, an alert is indicated in the validation dialog. If the system was started without a connection to the sensor, *ibaPDA* will attempt to establish a connection to the sensor at regular intervals.

Connection table
The table shows the cycle times and error counters of the individual connections during data measurement. Each table row corresponds to a configured Gocator module. To reset the calculated times and error counters to zero, simply click on the <Reset statistics> button.



The table columns and their meaning:

- Name: Name of the module
- Received Massages: Number of messages coming from configured/used sensors
- Error Count: Number of communication errors that occurred
- Unknown Sensor: Number of messages coming from sensors which are not configured or used. This number should be "0" during normal operation.
- Update time Actual, Average, Min, Max: The update time indicates the time between 2 read operations.

Additional information is provided by the background color of the table rows:

Color	Meaning
Green	The connection is OK and the data are read.
Red	The connection has failed or been interrupted.
Gray	No connection configured.

<Open log file>
Messages related to the LMI-Gocator interface are written in a separate log file. To open the current log file click <Open log file>.

<Reset statistics>
Click this button to reset the calculated times and error counters in the table to 0.

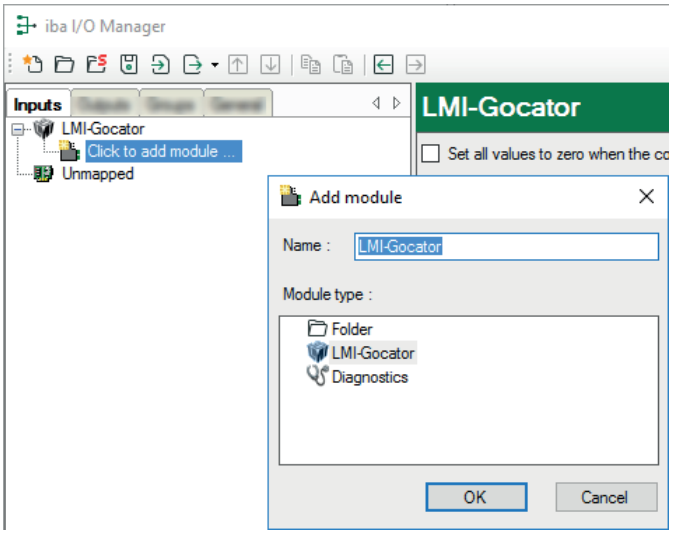
3.3.2 Adding a module

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

- ↗ *LMI-Gocator module*, page 9
- ↗ *LMI-Gocator Health module*, page 20
- ↗ *Diagnostic modules*, page 24

Procedure:

1. Click on the blue command *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>.

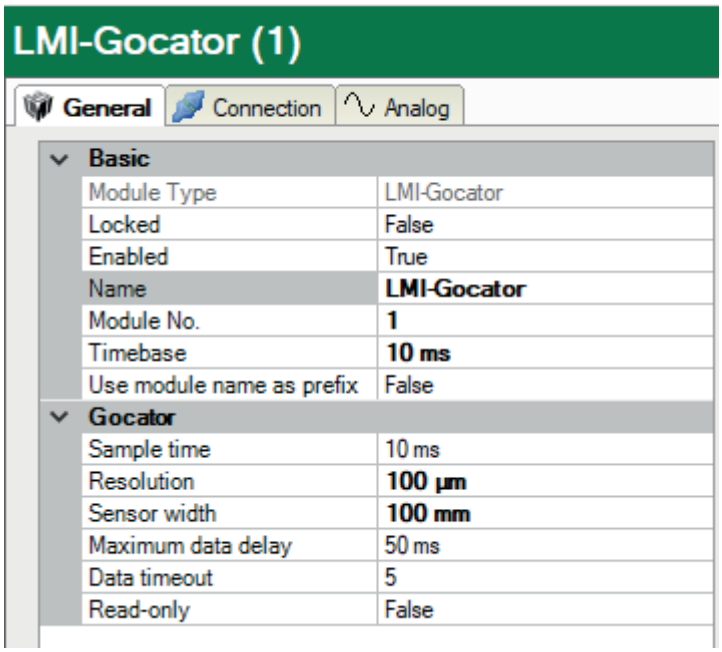


3.3.3 LMI-Gocator module

One Gocator module corresponds to a full profile which is typically (but not necessarily) the composition of the profiles of several, adjacent Gocator sensors.

3.3.3.1 LMI-Gocator – General tab

The following module settings can be configured on *General* tab:



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.

Gocator settings

In addition to the basic settings, the following settings can also be configured:

Sample time
Definition of how quickly the profile data is generated by the Gocator sensors

Theoretically, it is possible to set the sampling time to less than 1 ms, although in practical applications this may be limited by the sensors.

Resolution
Distance between two neighboring samples of the profile in micrometers

When the resolution value is changed, *ibaPDA* automatically adjusts the number of analog signals.

Sensor width
The sensor width determines the width of the measured profile on the X-axis (as defined in the Gocator documentation) of a Gocator sensor. This means that the profile width is the same for all Gocator sensors in the module. When using Gocator sensors in buddy mode, the master sensor acts as a relatively wide, virtual sensor; the sensor width then corresponds to the total profile width of all Gocator sensors together.

Maximum data delay
Maximum allowed difference (in ms) between the requested timestamp and the aczual time-stamp of the (reconstructed) profile

This parameter only needs to be changed in the event of network problems.

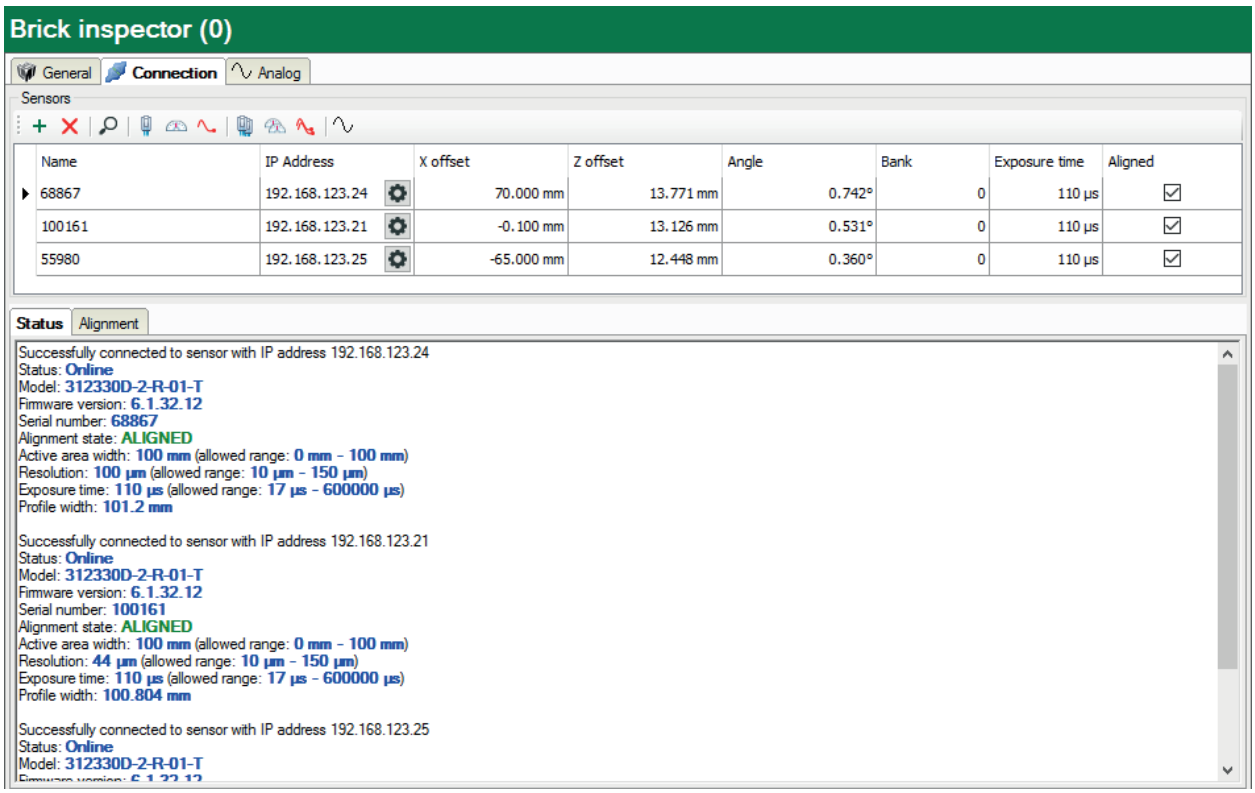
Data timeout
Number of sample times that *ibaPDA* can wait without receiving a new (reconstructed) profile.

This parameter only needs to be changed in the event of network problems.

Read-only
If this option is activated, *ibaPDA* does not change any settings in the Gocator sensors. If Gocator sensors are used in a buddy configuration, this option must be activated.

3.3.3.2 LMI-Gocator – Connection tab

In the *Connection* tab you can configure the sensors associated with a module.



Use the button + to add sensors manually and the button ✕ to remove sensors. You can also use the search function to list all available sensors in the network. Sensors that were found but are not required for the configuration can be removed from the table. For sensors in a buddy mode group, the master of the group is displayed.

The following settings can be configured for each sensor:

Name
This name is used internally by *ibaPDA* for identification and is not related to the sensor settings. When a new sensor is detected, the respective serial number is used as the default setting.

IP Address
The IP address at which a sensor can be reached for communication. By using the ⚙ button, the web interface of the sensor can be opened (standard web browser).

X offset
The offset of a sensor in mm along the axis of the sensor's laser line. The *X offset* value is retrieved from the sensor and saved in the sensor if the *read-only* module setting is disabled. This parameter is required when constructing a profile based on the data of multiple sensors. It can be measured manually or using *ibaPDA* when acquiring a test profile of all sensors (see below).

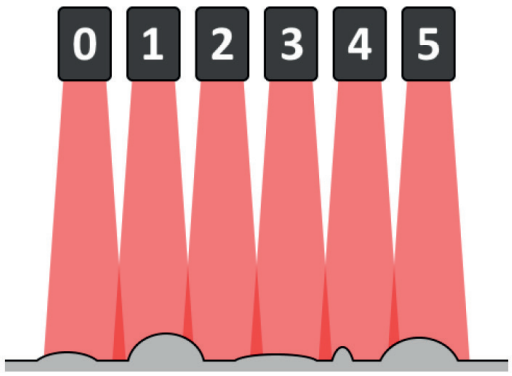
Note

The *X offset* value must be a multiple of the resolution of the Gocator module. For example, if the resolution is set to 500 µm, the value 10.486 mm is automatically set to 10.500 mm.

Z offset
The offset of a sensor in height direction. The value is retrieved from and stored in the sensor if the *read-only* module setting is disabled. Typically, this value is obtained by using the sensor's alignment function (see below).

Angle
The angle between the object to be measured and the sensor plane. The value is retrieved from and stored in the sensor if the *read-only* module setting is disabled. Typically, this value is obtained by using the sensor's alignment function (see below).

Bank
The number of the bank to which the sensor is assigned. A bench is a subgroup of sensors that can generate a laser line and measure the profiles simultaneously without interfering with each other. The following illustration shows an arrangement of 6 sensors measuring a slab:



In the ideal case, if the sensors can be positioned exactly next to each other, there is no overlap of the laser lines. However, if the measured object has high contours, the covered width of the laser line is smaller than with low contours. If the entire slab with its high and low contours is to be covered, the sensors must be positioned with a certain overlap of their laser lines.

If all sensors were to generate a laser line and measure a profile at the same time, the laser line from sensor 1, for example, would interfere with the laser line from sensor 0 and thus also influence the measurement data. Sensor 1 would also interfere with sensor 2 at the same time. To prevent this, the exposure can be staggered over time in groups, the so-called banks.

Example:
Assuming, you want to obtain a full profile every millisecond (i. e. continuous measurements can be divided into time windows of 1 ms), whereby the exposure time (i. e. the time required to perform a valid profile measurement) of a laser is 400 µs. For this purpose, in each time window, sensor 0 can generate the laser line and measure the profile from 0 µs to 400 µs, while sensor 1 does this for the range from 400 µs to 800 µs. This leaves a span of 200 µs and there is no longer any mutual interference, as the sensors, which would normally overlap, generate their laser lines at different times. Since sensor 1 and sensor 2 also overlap, the time-division multiplexing method should also be used for the exposure time. Sensor 0 and sensor 2, on the other hand, can generate the laser line at the same time as they do not overlap.

In the figure above, this implies that sensors 0, 2 and 4 can generate the laser line at the same time (e.g. in the subslot from 0 µs to 400 µs) and sensors 1, 3 and 5 can generate the laser line at a different time (e.g. in the subslot from 400 µs to 800 µs). Sensors 0, 2 and 4 form the first bank, while sensors 1, 3 and 5 form the second bank.

Since the Gocator sensors work independently of each other and do not know each other, this setting is not saved in the sensor.

Exposure time
The time required to perform a valid profile measurement for this sensor. This value is retrieved from and stored in the sensor. This value is normally obtained by configuring the sensor via its web interface (by checking the live image).

Aligned
A read-only field that indicates whether the sensor has been successfully aligned. This field is updated when a sensor is automatically added via the discovery function or a connection test is performed.

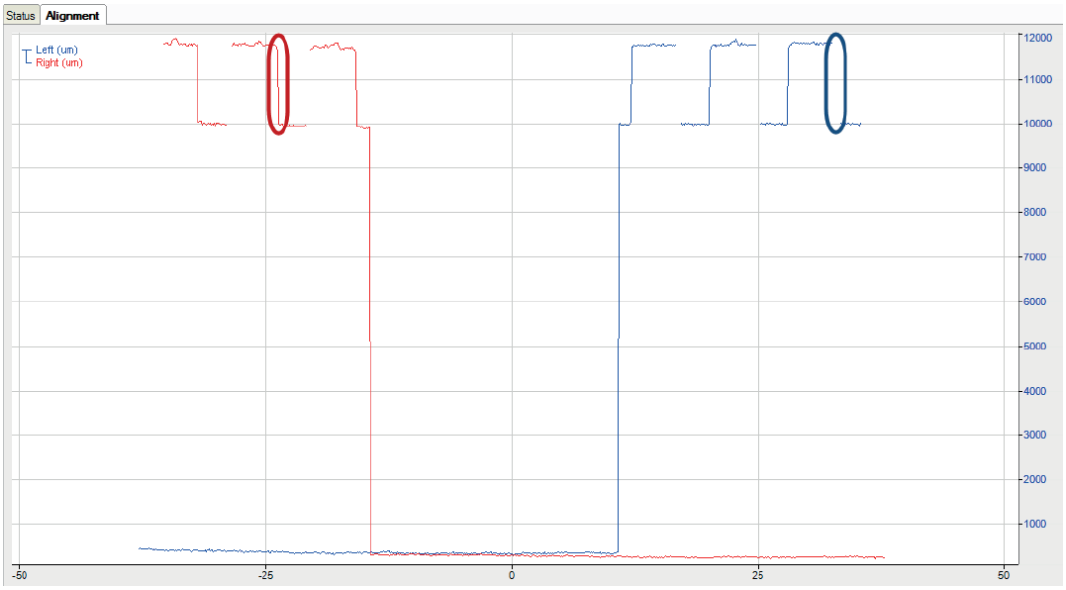
Checking a sensor's status and connection
To check whether a connection to a sensor can be established or to obtain basic diagnostic information, select the respective sensor and click on the button  . To check the connection to all listed sensors, click on the button  .

Select the *Status* tab below the list. When the connection is checked, the current status, the model, the firmware version, the serial number of the sensor and, if applicable, the status information of its buddy sensors are displayed.

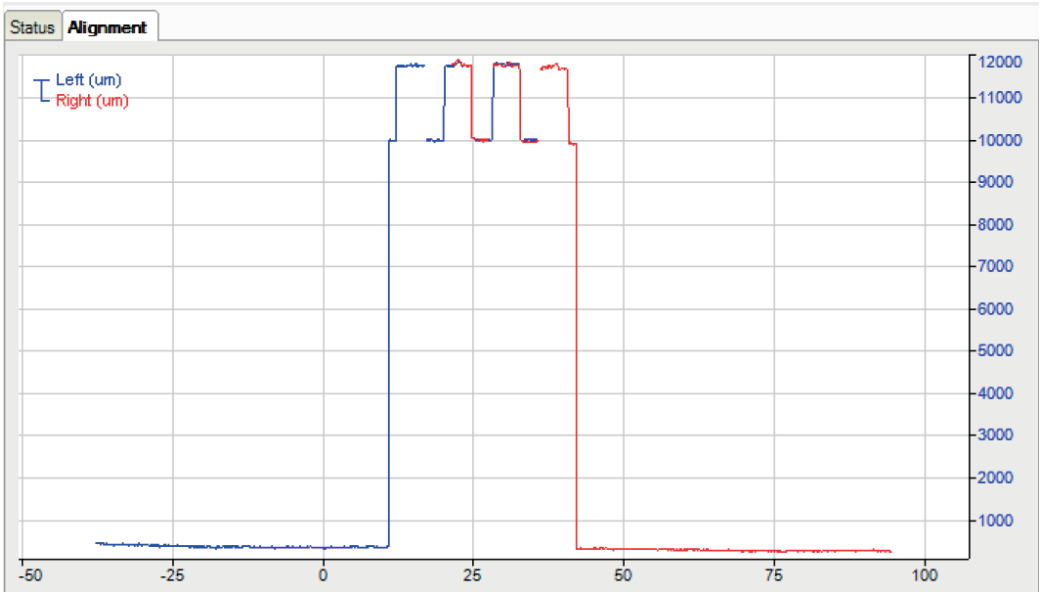
Aligning a sensor (not available in read-only mode)
Before the sensors can be used for a measurement, the *Z offset* and *Angle* values must be configured correctly. To do this, place a flat surface underneath the sensor (i.e. the position on which the measurement object will later be placed) and click the  button to align the selected sensor or the  button to align all sensors. The *Z offset* and *Angle* values are updated automatically.

Acquiring a single profile

Although the *X offset* parameter is also set when the sensors are aligned, this will not initially be the correct value: "X-Offset" refers to the distance from a sensor to a reference sensor. However, as the sensors do not know each other, it is not possible to determine this value automatically. However, the following method makes it possible to determine a relatively accurate value for "X-Offset": by clicking the button (for a single sensor or sensors in buddy mode) or the button (for all sensors), the current profile is retrieved and displayed in the *Alignment* tab.



The graphic above shows 2 aligned sensors; however, the "X-Offset" value is not yet set correctly. As the sensors overlap, their profiles should partially match. In this example, the area circled in red and the area circled in blue should overlap. Markers can be used to measure the distance between the two areas in the graphic; this value can then be used as the "X offset" for one of the sensors. If this value is entered in the sensor table and then the profiles are queried again, the following result is obtained:



The profiles now overlap correctly and the correct value for X-Offset has been found.

Note

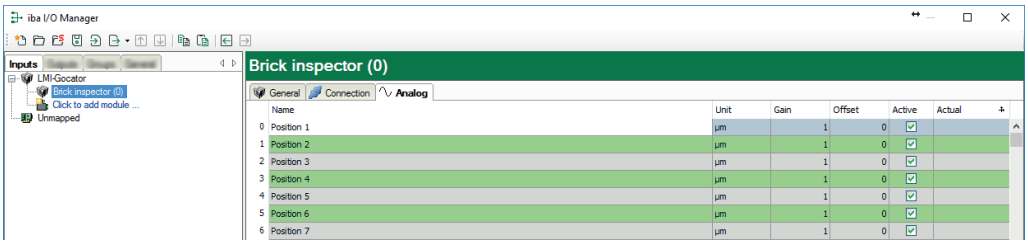
Markers can only be made visible after exiting live mode. You have the following options for exiting live mode:

1. Right-click in the graph (context menu) - select *Live mode* or
2. Zoom into the graph or
3. Press <F6>.

To return to live mode, press <F6> again or use the context menu.

As soon as the *X offset* parameter has been adjusted, the number of signals must be updated. To perform the update, click on the button . Based on the resolution of the module and the *X offset* parameters of all sensors, *ibaPDA* generates the required number of signals. Each signal corresponds to a single data point of the entire profile.

3.3.3.3 LMI-Gocator – Analog tab

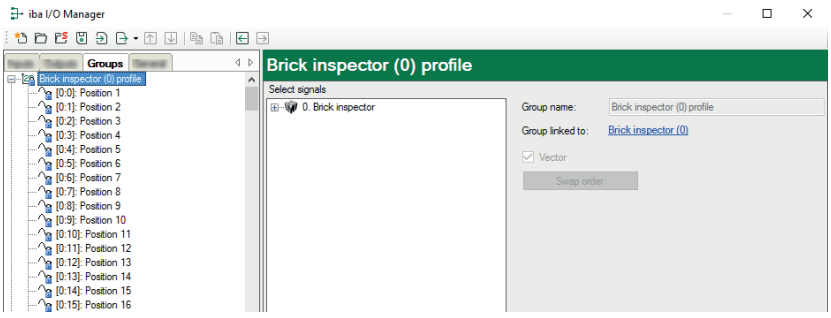


No further changes need to be made in the *Analog* tab. If desired, you can optionally change the signal names *Gain* and *Offset*.

3.3.4 Vector signal

ibaPDA automatically generates a vector signal with the profile data for each LMI-Gocator module. The vector signal can be fornd in the I/O Manager under *Groups*.

The name consists of the module name, the module number and the suffix "profile".

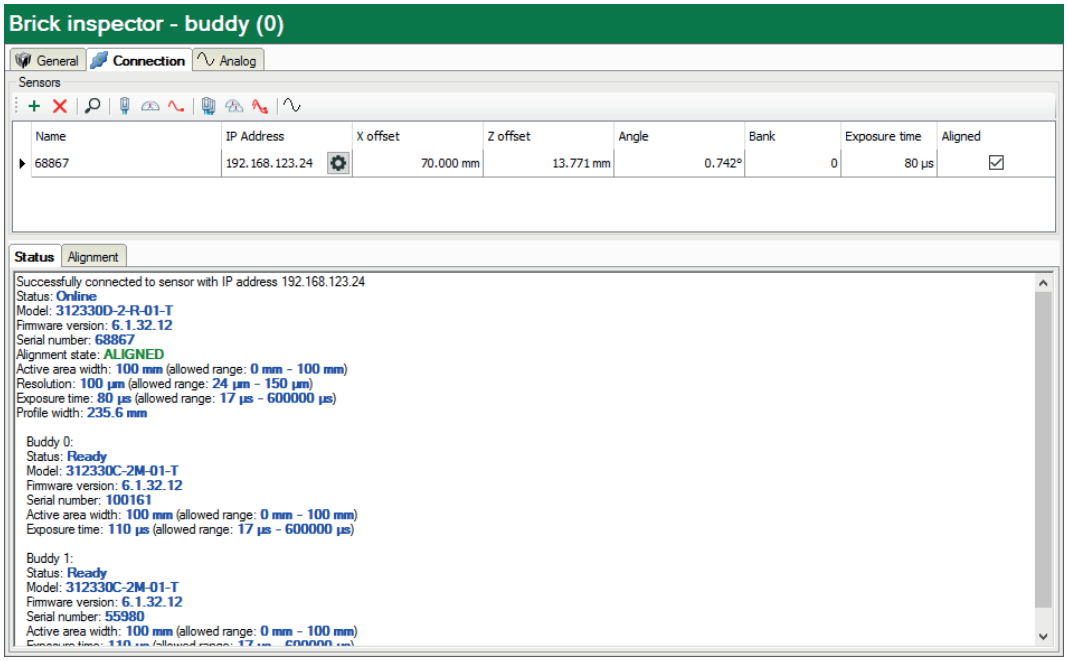


Tip

The vector signal can be used directly in the 2D color-coded display of the surface profile in *ibaQPanel* or *ibaAnalyzer*.

3.3.5 Buddy mode

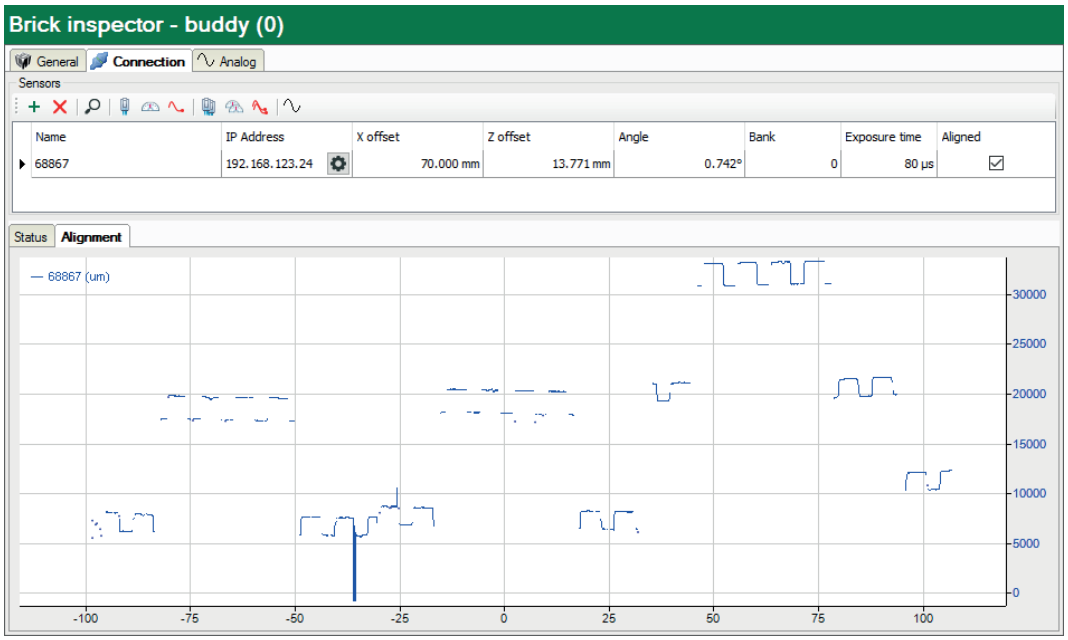
In case the firmware of the Gocator sensors allows for it, it is advised to configure multiple Gocator sensors in buddy mode and to set all relevant parameters using their web interface. In *ibaPDA* it is then only required to configure the general module settings (note that read-only mode must be enabled) and adding one sensor (i.e. the main sensor of the buddy group) with its IP address (other settings like X and Z offset are not relevant). Only the main sensor is discovered and accessible when using multiple Gocator sensors in buddy mode (see the figure below):



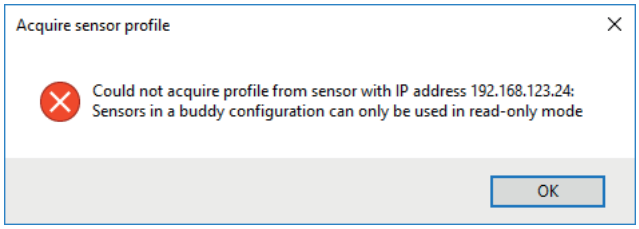
When doing a connection test the status information of the main sensor will be displayed in the *Status* tab along with some basic information concerning its configured buddies.

The total profile width (i.e. the profile width of all buddy sensors combined) is also displayed and can be used to configure the *Sensor width* property in the *General* tab of the module.

In the *Alignment* tab you can see the measured values of all sensors belonging to the buddy group and check their alignment.



When read-only mode has not been enabled in the Gocator module and you try to acquire a profile or align the sensors, an error will appear in *ibaPDA* indicating that buddy sensors can only be used when read-only mode is enabled.

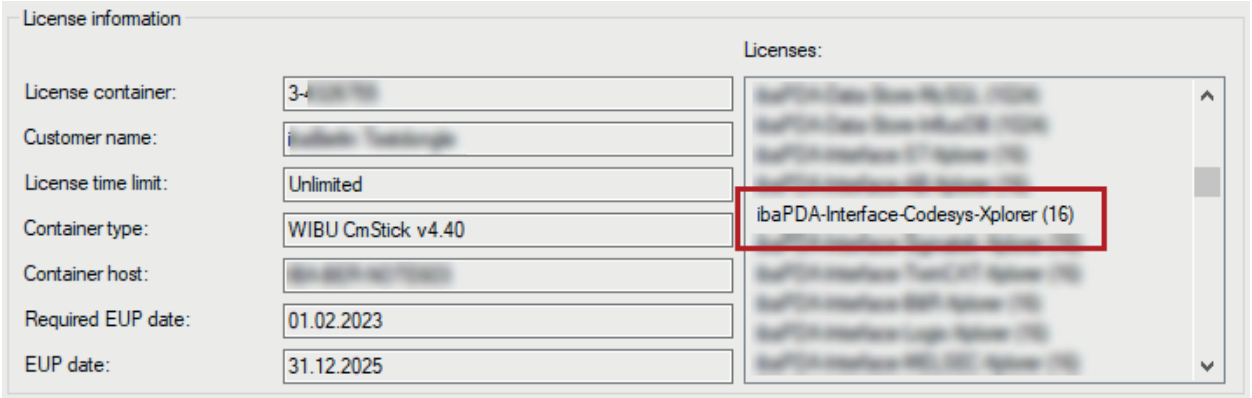


4 Diagnostics

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 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.



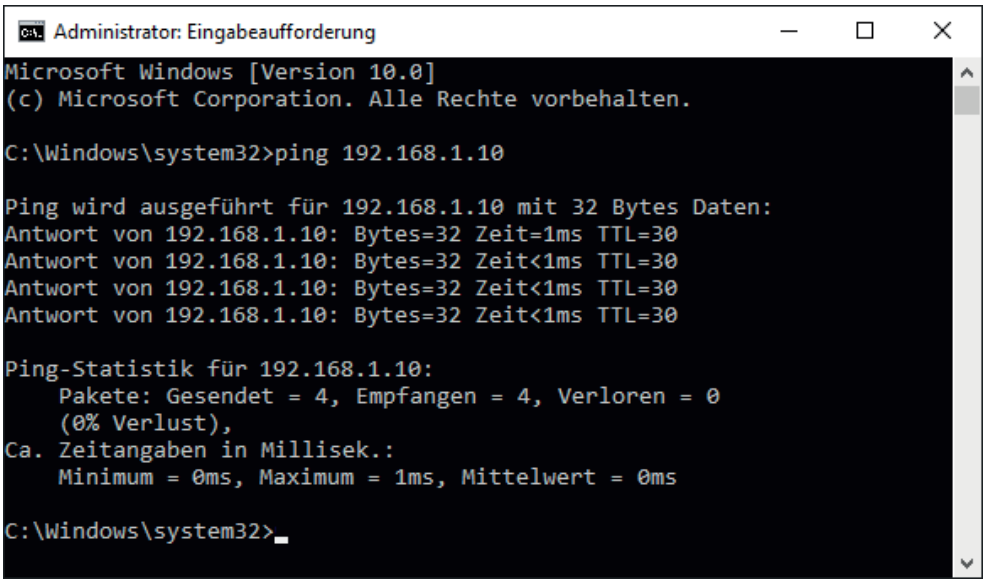
4.2 Connection diagnostics with PING

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

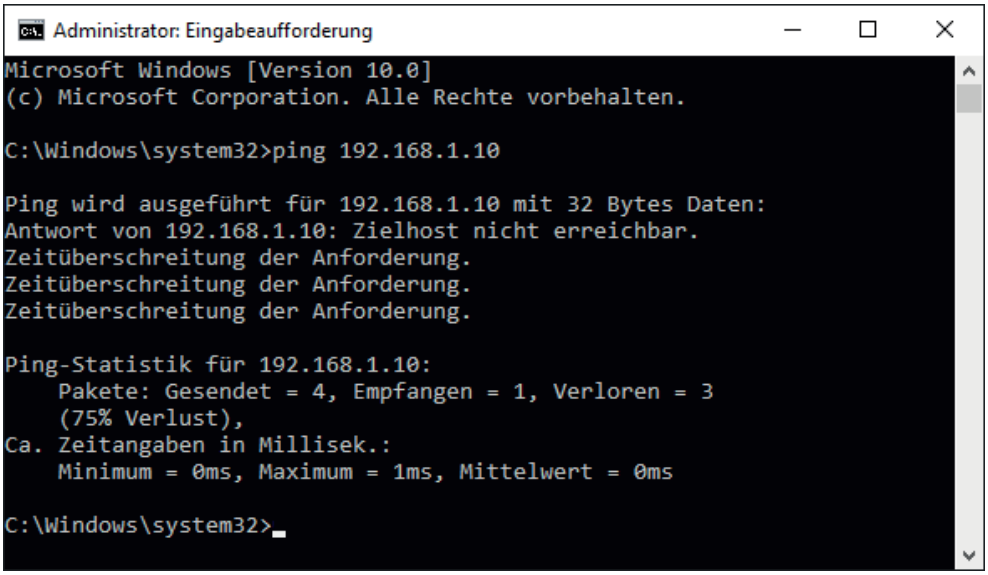
1. Open a Windows command prompt.

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.



4.3 Check connection

You can check the status of the connections via the connection table of the interface. The background color of the table rows provides information on the status of the connection, see chapter ↗ *Interface settings*, page 7.

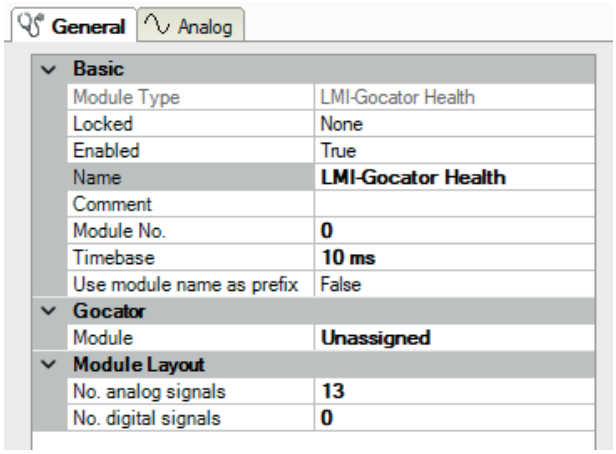
You can also check the connection status of individual sensors, see ↗ *Checking a sensor's status and connection*, page 13.

4.4 LMI-Gocator Health module

The "LMI-Gocator Health" module allows you to record diagnostic and status information that is sent from the Gocator sensors with the measurement data to *ibaPDA*.

4.4.1 LMI-Gocator Health – General tab

The following module settings can be configured on *General* tab:



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.

Gocator settings

In addition to the basic settings, the following settings can also be configured:

Module
LMI-Gocator module to which the status module is linked

A status module must always be linked to a regular LMI-Gocator module.

Sensor
Selection of the sensor for which the diagnostic and status data is to be measured

The field is only displayed if several Gocator sensors are configured in the LMI Gocator module.

Module Layout

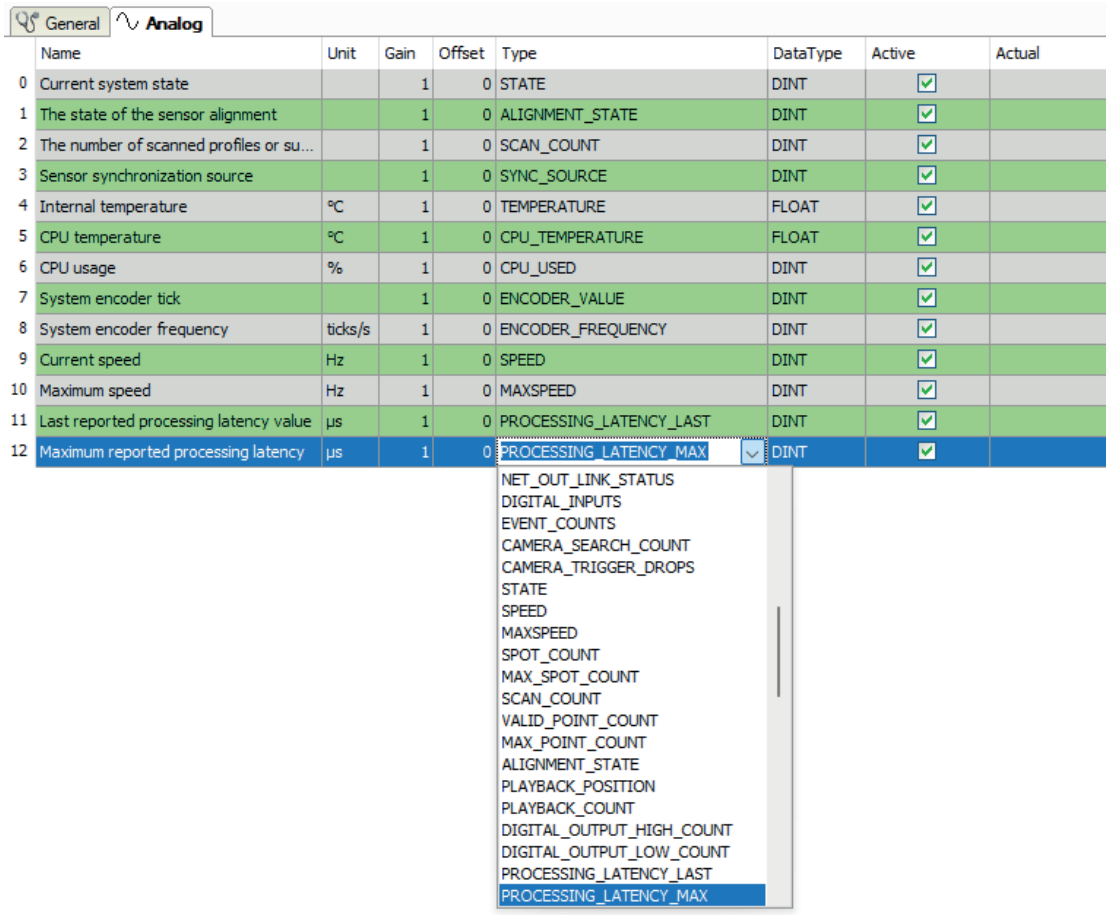
Number of analog signals
Number of analog signals measured with the health module

For further information on displaying the analog signals, see chapter ↗ *LMI-Gocator Health – Analog and Digital tab*, page 22.

Number of digital signals
Number of digital signals measured with the health module

4.4.2 LMI-Gocator Health – Analog and Digital tab

When you add an LMI-Gocator Health module, a standard list of frequent signals is automatically created in the *Analog* tab.



Name

Here, you can enter a signal name and additionally two comments when clicking the  symbol in the *Name* field.

Unit

Here, you can enter the physical unit of the analog value.

Gain / Offset

Gradient (Gain) and y axis intercept (Offset) of a linear equation. You can convert a standardized and unitless transmitted value into a physical value.

Type

Type of status signal

The selection list contains all signal types that were known at the time of implementation. Each signal type corresponds to a unique integer value as defined in the LMI-Gocator SDK. A Gocator sensor sends a list of value pairs every second, consisting of an ID (integer value that corresponds to the status signal type) and the actual diagnostic value. *ibaPDA* goes through this list and checks for each entry whether it should be measured.

If an additional signal type is required, you can manually enter an integer value in the column.

Data type
Data type of the status signal

The data type is automatically assigned for the status signal types stored in *ibaPDA*. If you define your own signal types, select the corresponding data types from the selection list.

Active
Only when this option is selected, the signal is acquired and considered when checking the number of licensed signals.

More columns can be displayed or hidden, using the context menu (right mouse-click on the header).

Insert list of all existing signals
You can add all signals stored in *ibaPDA* to the Analog tab.

1. In the I/O Manager, select the LMI-Gocator Health module by holding down the <Shift> key.

   If the module was already selected, first select another node to deselect it.

→ An additional link "Add all known signals" is displayed in the *General* tab.

2. Click on the link.

   All existing analog and digital signals are entered in the corresponding registers.

4.5 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.

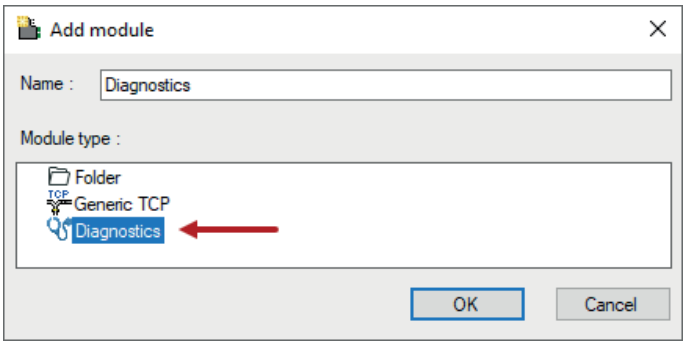
A diagnostic module is always assigned to a data acquisition module of the same interface and supplies its connection information. By using a diagnostic module you can record and analyze the diagnostic information continuously in the *ibaPDA* system.

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:

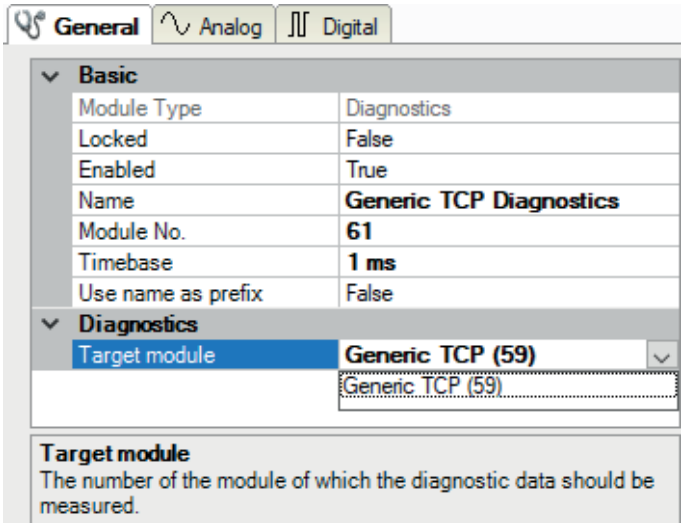
- A notification can be generated, whenever the error counter of a communication connection exceeds a certain value or the connection gets lost.
- In case of a disturbance, the current response times in the telegram traffic may be documented in an incident report.
- The connection status can be visualized 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).



Module settings diagnostic module

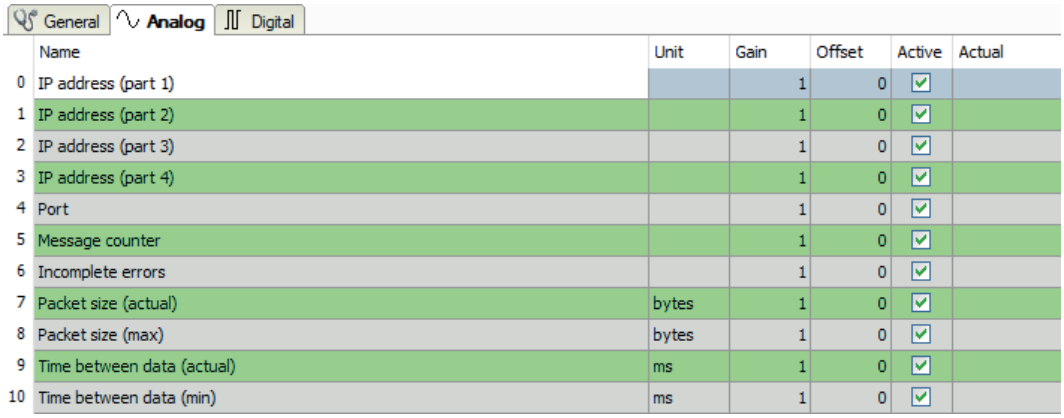
For a diagnostic module, you can make the following settings (example: Generic TCP):



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

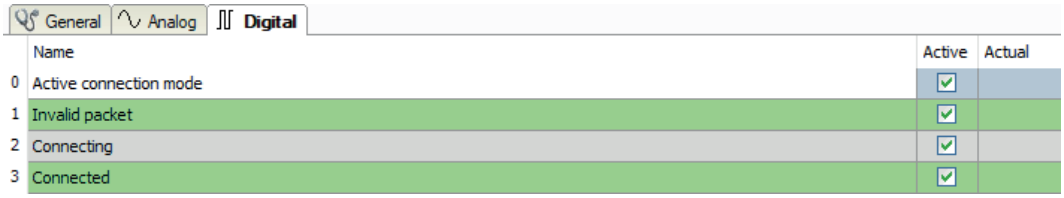
There is only one setting which is specific for the diagnostic module: the target module.

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.



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

Signal name	Description
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)
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

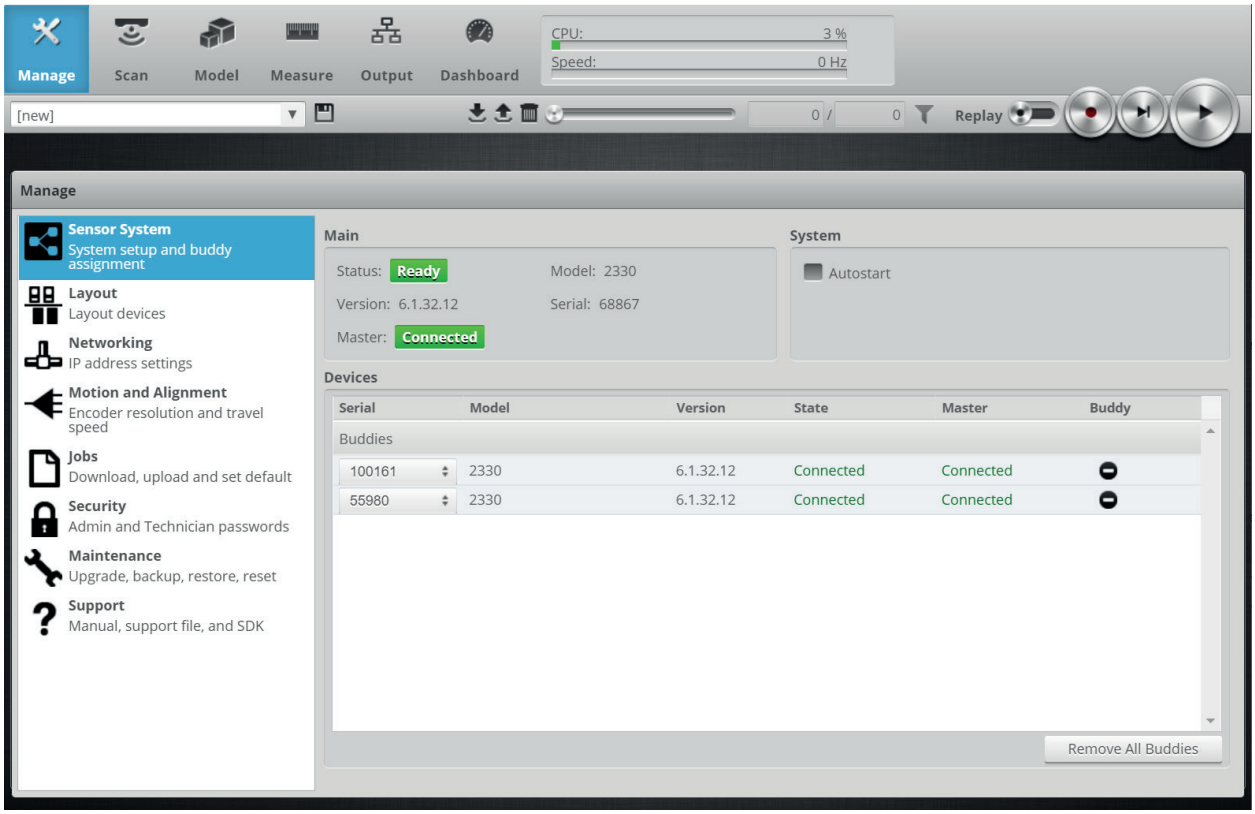
Signal name	Description
Profile Count	Number of completely recorded profiles
Read counter	Number of read accesses/data requests
Receive counter	Number of messages received
Response time (actual/average/max/min)	Response time is the time between measured value request from *ibaPDA* and response from the 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

Signal name	Description
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

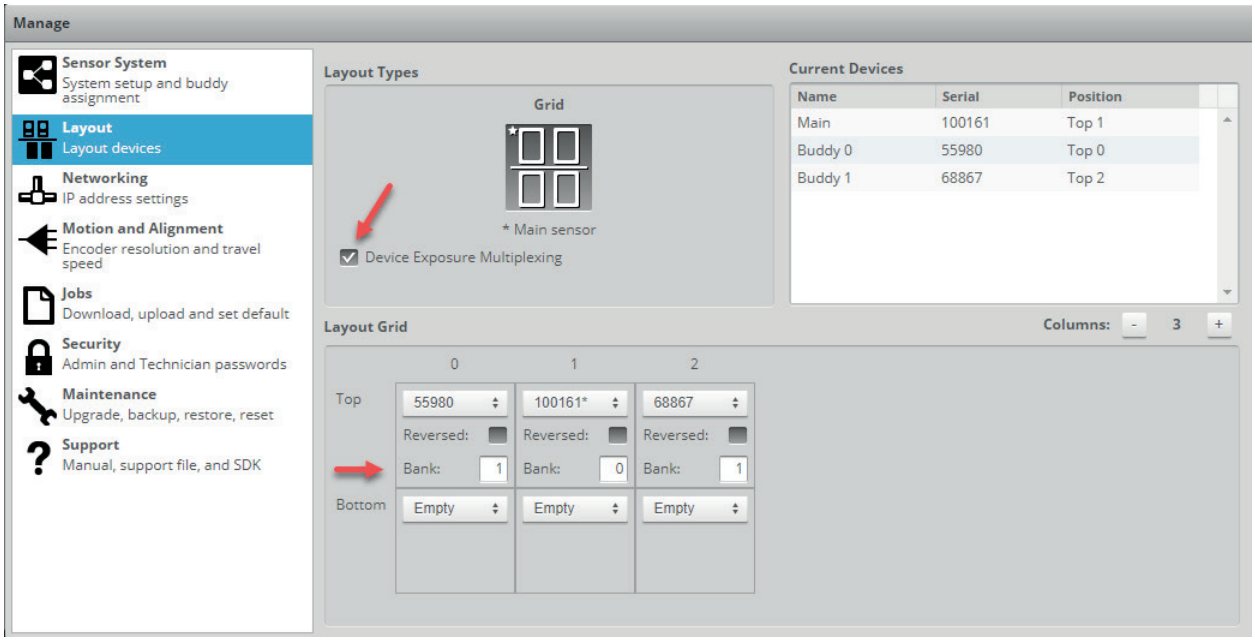
4.6 Gocator sensor web interface

Although configuring the Gocator sensors using their web interface is outside of the scope of this manual, some troubleshooting tips are listed here. Note that some settings are only available when accessing the sensor's web interface in advanced mode. To enter advanced mode, use the following URL format: http://[sensor_ip]/?advanced=1

Buddy mode: make sure the sensors are configured as buddies
Go to *Manage* → *Sensor System* and make sure the buddies are connected to the Master.

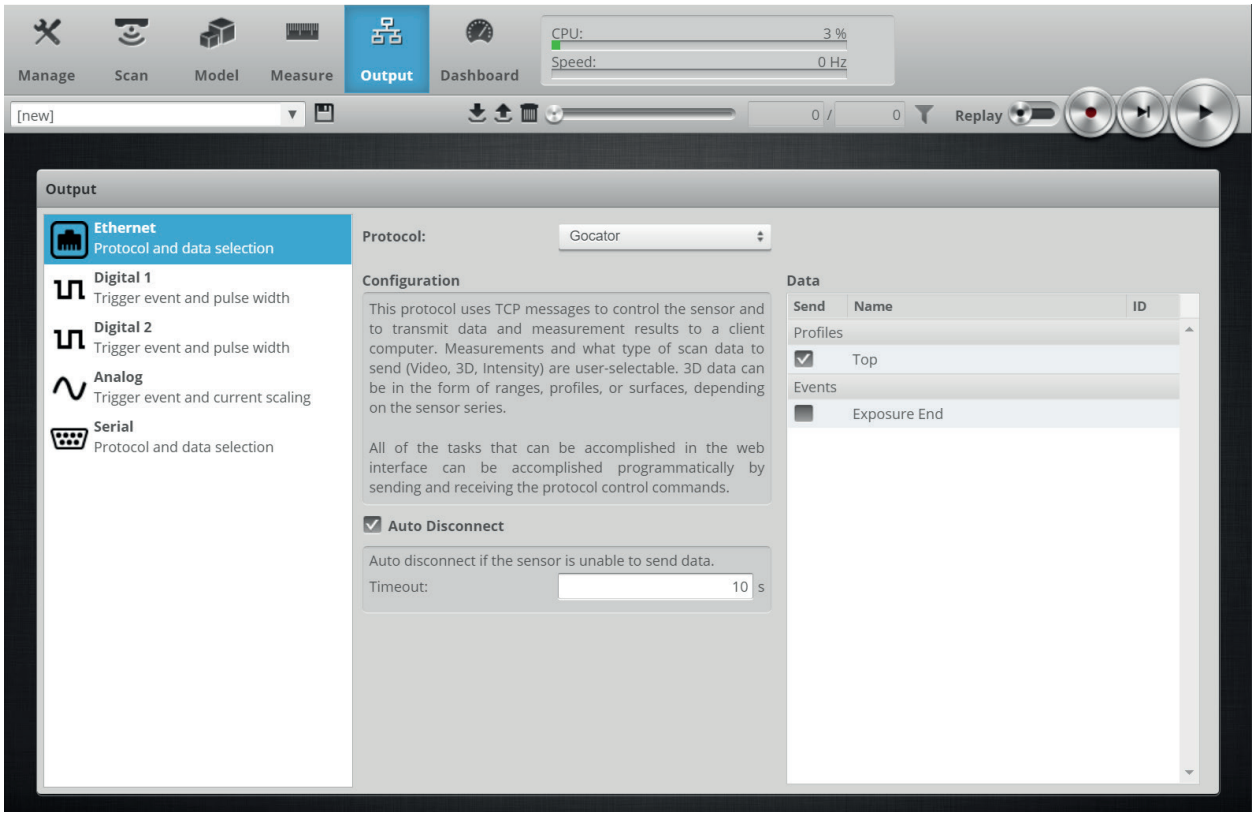


Also verify that all buddies are part of the layout (*Manage → Layout*). It is recommended to activate the "Device Exposure Multiplexing" option, as the banks can be set here, see ↗ *LMI-Gocator – Connection tab*, page 11.



No profile data can be obtained

Make sure that the Top profile is being sent by the (main) sensor.



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