ibaPDA-Interface-LMI-Gocator

Data Interface for LMI-Gocator sensors

Manual

Issue 2.0

Measurement Systems for Industry and Energy

www.iba-ag.com

Manufacturer

iba AG

Koenigswarterstrasse 44

90762 Fuerth

Germany

Contacts

Main office	+49 911 97282-0
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	Author	Version SW
2.0	11-2023	New version ibaPDA v8	rm	8.5.0

Windows® is a brand and registered trademark of Microsoft Corporation. Other product and company names mentioned in this manual can be labels or registered trademarks of the corresponding owners.

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 more recent

- 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-LMI-Gocator 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).

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.

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

Table 1: Available LMI-Gocator interface licenses

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 devices and *ibaPDA* can be established via the computer's standard Ethernet ports.

No further software is necessary for operation.

Note

It is recommended carrying out the TCP/IP communication on a separate network segment to exclude a mutual influence by other network components.

3.3 Configuration and engineering 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 options:
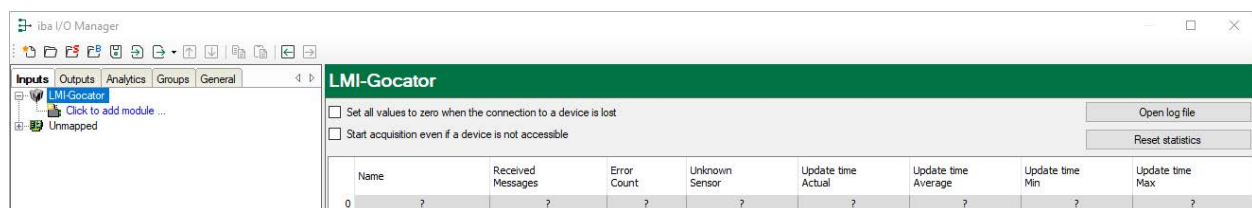
Set all values to zero when the connection to a device is lost
If enabled, all measured values of the device are set to zero as soon as the connection is lost. If this option is disabled, *ibaPDA* will keep the last valid measured data in memory at the time the connection was lost.

Start acquisition even if a device is not accessible
If this option is enabled, the acquisition will start even if the device is not accessible. In case of an error, a warning is indicated in the validation dialog. If the system has been started without a connection to the device, *ibaPDA* will periodically try to connect the device.

Connection table

The table shows the cycle times and error counters for the individual connections during data measurement. Each table row corresponds to a configured Gocator module. To reset the calculated times and error counters, simply click on the <Reset counters> button. See ↗ *Connection table*, page 20

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
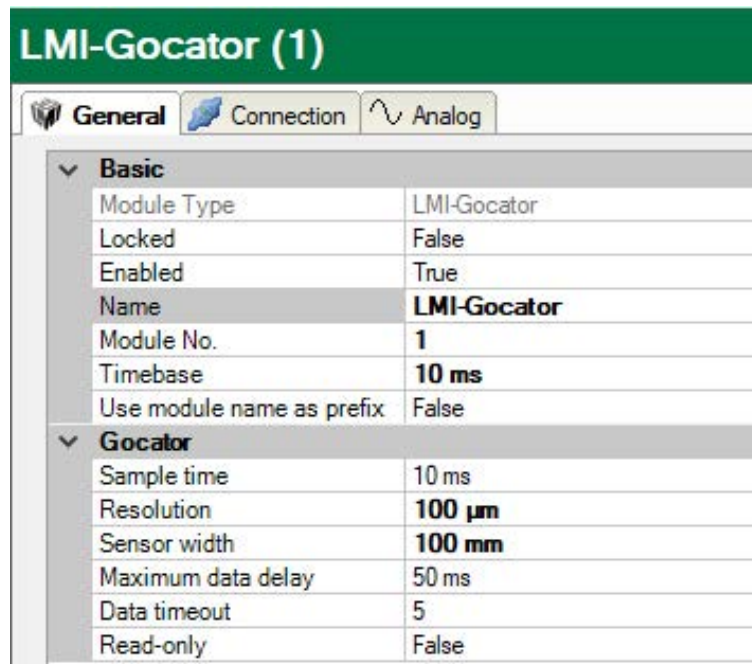
1. Click on the blue command *Click to add module...* located under each data interface in the *Inputs* or *Outputs* tab.

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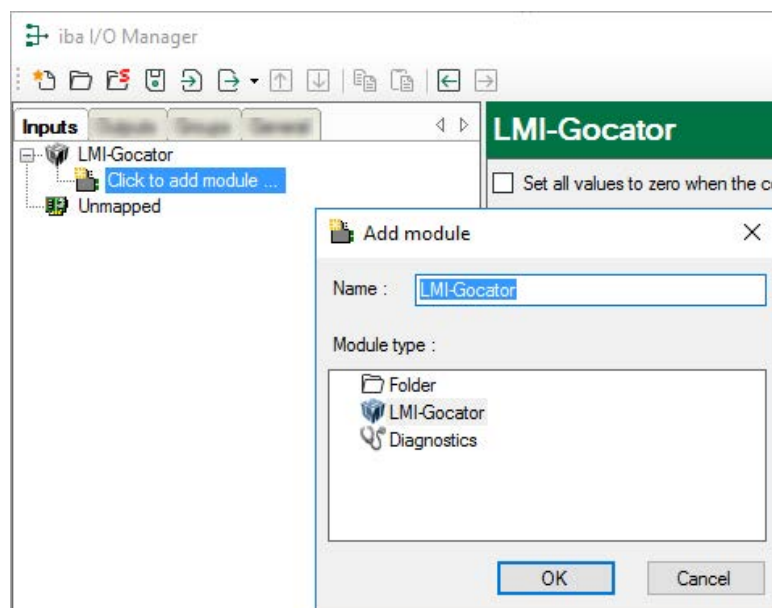
One Gocator module corresponds to a full profile, which is typically, but not necessarily, the composition of the profiles of several adjacent Gocator sensors.

3.3.3   General module settings

In the *General* tab the following module settings can be configured:

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.

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

One Gocator module corresponds to a full profile which is typically (but not necessarily) the composition of the profiles of several, adjacent Gocator sensors. Apart from the standard options, the following specific settings can be configured:

Sample time
This determines how fast profile data is generated by the Gocator sensors. Although in theory, it is possible to set the sample time as low as 1 ms, in practice it might be limited by the sensors.

Resolution
This determines the distance in microns between two neighbouring profile samples. Allowed values are 100 µm, 200 µm, 250 µm, 500 µm and 1000 µm. When changing the resolution, *ibaPDA* will automatically adjust 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 one Gocator sensor. This implies 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 one relatively wide virtual sensor; the sensor width then corresponds to the total profile width of all Gocator sensors combined.

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

Change this parameter only after consulting the iba support.

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

Change this parameter only after consulting the iba support.

Read-only
When enabled, *ibaPDA* will not change any settings in the Gocator sensors. This must be en-abled when using Gocator sensors in a buddy configuration.

3.3.4   Configuration of sensors

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

Sensors can be added and removed manually by using the ✚ button and ✕ button respectively. In addition the discovery functionality (🔍) will list all available sensors in the network. Discovered sensors which are not necessary for the configuration can be removed from the table. For sensors in a buddy mode group the master of the group is discovered and listed.

For each sensor, the following settings can be configured:

Name
This is an identifier used internally in *ibaPDA* and is not related to any setting in the sensor. The default value when discovering a new sensor is the serial number.

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 and stored in the sensor (if the read-only module property 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

|	
ℹ	The "X offset" value must be a multiple of the Gocator module's resolution. For example, if the resolution is set to 500 µm, a value of 10.486 mm will be automatically set to 10.500 mm.

Z offset

The offset of a sensor in the height direction. This value is retrieved from and stored in the sensor (if the read-only module property is disabled). Typically this parameter is obtained by using the alignment function of the sensor (see below).

Angle

The angle between the object to be measured and the sensor plane. This value is retrieved from and stored in the sensor (if the read-only module property is disabled). Typically this parameter is obtained by using the alignment function of the sensor (see below).

Bank

This is the number of a bank, the sensor is assigned to. A bank is a subgroup of sensors that can generate the laser line and measure the profile simultaneously without interfering with each other. The following figure illustrates an array of 6 sensors measuring a slab:
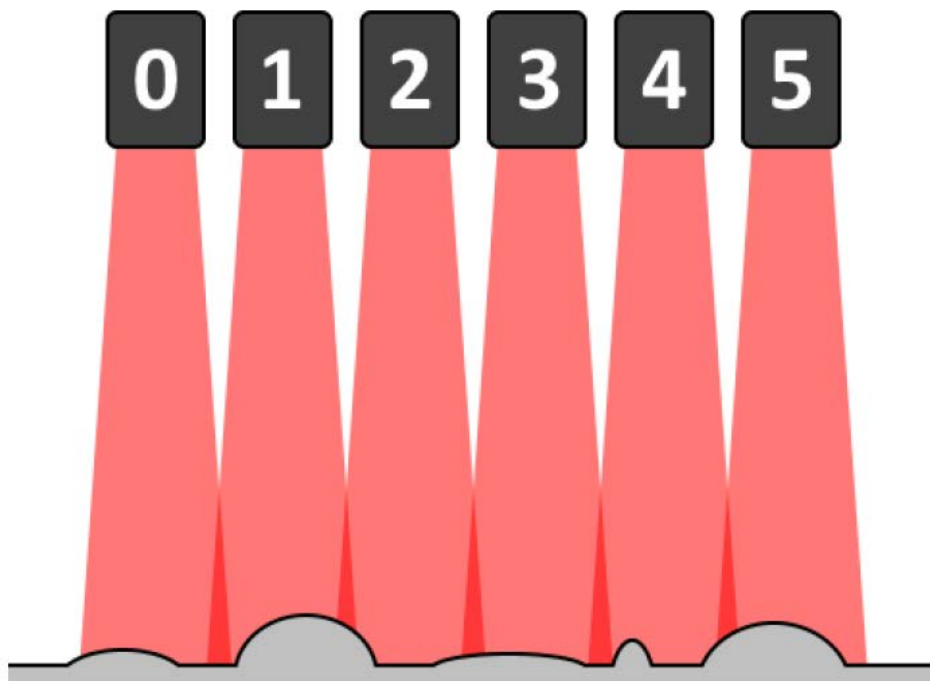
In the ideal case when the sensors can be positioned precisely next to one another, there will be no overlap of laser lines. However for high features in the slab, the projected width of the laser line of the sensor is smaller than for low features. Thus if an entire slab with relatively high and low features should be covered, the sensors must be positioned with some overlap between the laser lines.

In case all sensors would be generating the laser line and measuring the profile at the same time, the laser line from sensor 1 would interfere with the one from sensor 0 and would affect the measurement data. Note that sensor 1 would also interfere with sensor 2. To prevent this unwanted behavior, it is possible to time multiplex the exposure.

Example:

Assuming you would like to obtain a full profile every millisecond (i.e. you can divide a continuous measurement in time slots of 1 ms) and the exposure time (i.e. the time required to get a good measurement of the profile) of one sensor is 400 µs. In each time slot, sensor 0 could generate the laser line and measure the profile from 0 µs to 400 µs and sensor 1 could do so from 400 µs to 800 µs, thus still leaving a margin of 200 µs. There is no longer any interference, since the sensors that normally overlap now generate their laser lines at different points in time. Since sensor 1 and sensor 2 also overlap, they should also time multiplex the exposure. However, since sensor 0 and sensor 2 do not overlap, they can generate the laser line simultaneously.

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

Since the Gocator sensors operate independently and are unaware of each other, this setting is not saved in the sensor.

Exposure time

The required time in microseconds to generate a valid profile for this sensor. This value is retrieved from and stored in the sensor (if the read-only module property is disabled). Typically it is obtained by configuring the sensor in 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 using the discovery function or every time a connection test is done.

Checking a sensor's status and connection

To check whether a connection to a sensor can be created or to obtain some basic diagnostic information, select the sensor in question and click the ⬙ button to test the connection to the sensor. You may click on the ⬙ button to test the connection to all listed sensors.

Select the *Status* tab below the list. Testing the connection will display the current status, model, firmware version and serial number of the sensor and, if applicable, the status information of its buddy sensors.

Aligning a sensor (not available in read-only mode)

Before using the sensors in a measurement the "Z offset" and "Angle" need to be configured properly. To do so, place a flat surface below the sensors (i.e. where the measurement object will eventually be placed) and click the ⬭ button to align the selected sensor or the ⬭ button to align all sensors. The "Z offset" and "Angle" will be updated automatically.

Acquiring a single profile

Though the "X offset" will also be set by aligning the sensors, this will not be the correct value: the "X offset" is the distance of one sensor to another reference sensor. Since sensors are unaware of each other, there is no way to determine this automatically. However the following method can make it easier to get a relatively precise value for this "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.
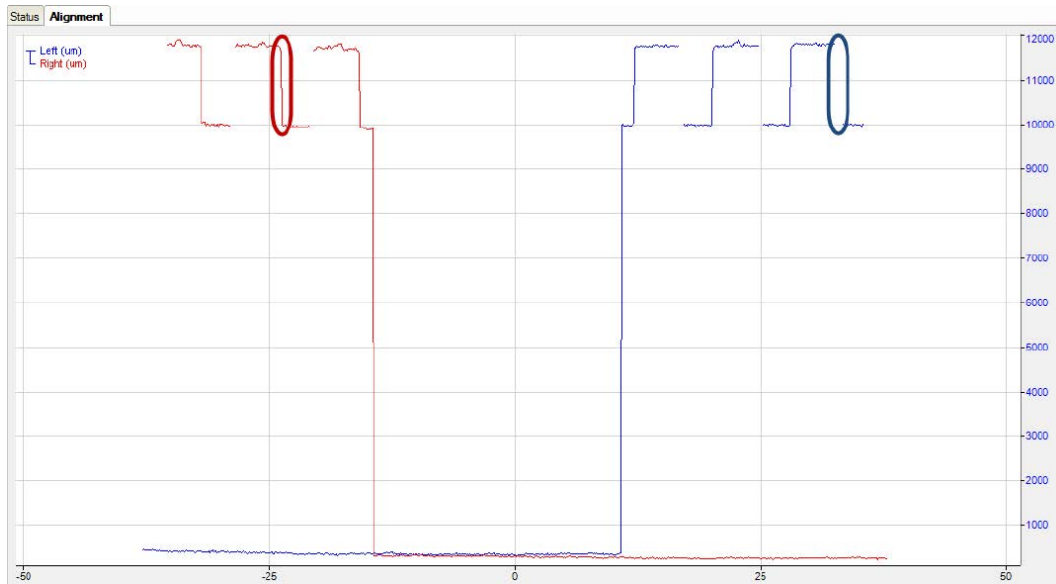
In the graph above, the profile for two aligned sensors is displayed; the "X offset" however is not yet set correctly. Since the sensors overlap, part of their profiles should match. In this example, the red circled region should overlap with the blue circled region. Using markers in the graph you can measure the distance between these two regions and use those value as the "X offset" of one of the sensors. When you enter this value in the sensor table and acquire the profiles again, you get the following result:
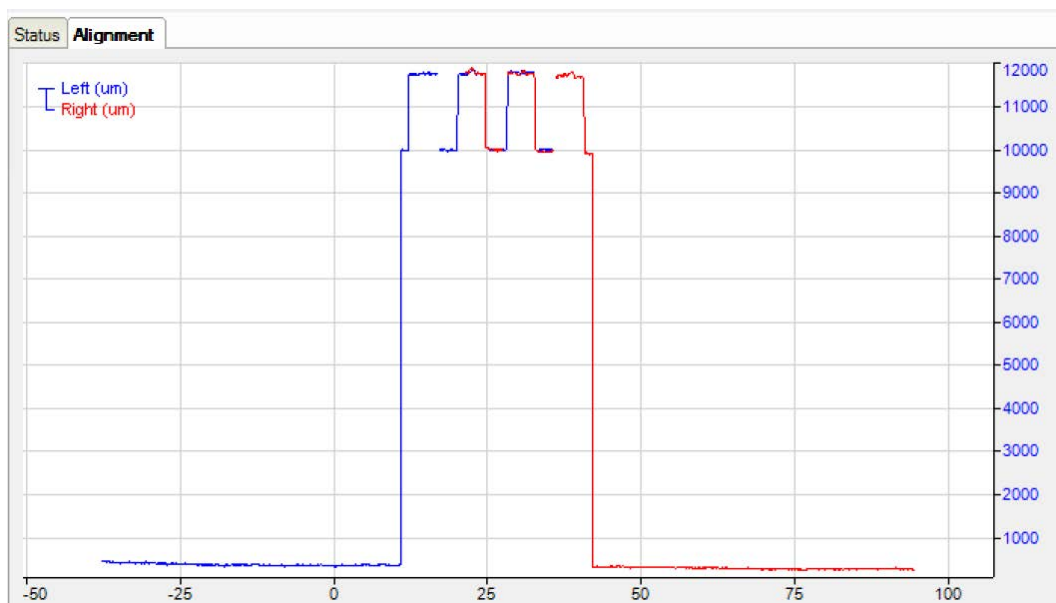
Now the profiles overlap as they should and apparently the correct value for the "X offset" has been found.
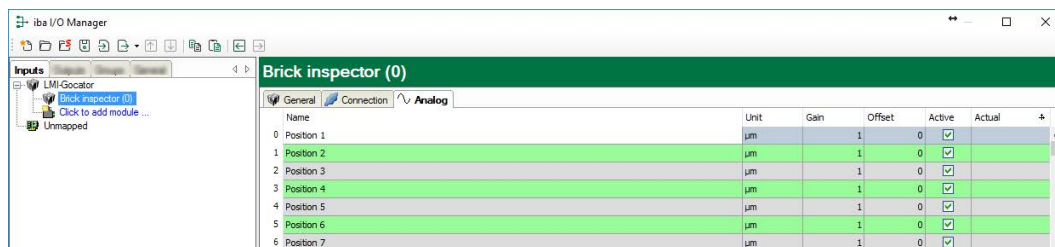
Note

Markers can be made visible by exiting the live mode. You have the following options to exit live mode:

1. Right click on the graph (context menu) - select *Live mode* or

2. Zoom into graphic or

3. Press <F6>

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

Once the "X offset" has been adjusted, the number of signals needs to be updated. This can be done by clicking the ∿ button. Based on the module's resolution and "X offset" parameters of all sensors, *ibaPDA* will generate the required number of signals with each signal corresponding to a single data point of the entire profile.

3.3.5 "Analog" tab

In the *Analog* tab, no further changes need to be made (apart from changing the signal names "Gain" and "Offset", if desired).

3.3.6 Vector signal

For each LMI-Gocator module *ibaPDA* generates automatically a vector signal with the measured line data. The vector signal can be found in the "Groups" section of the I/O Manager.

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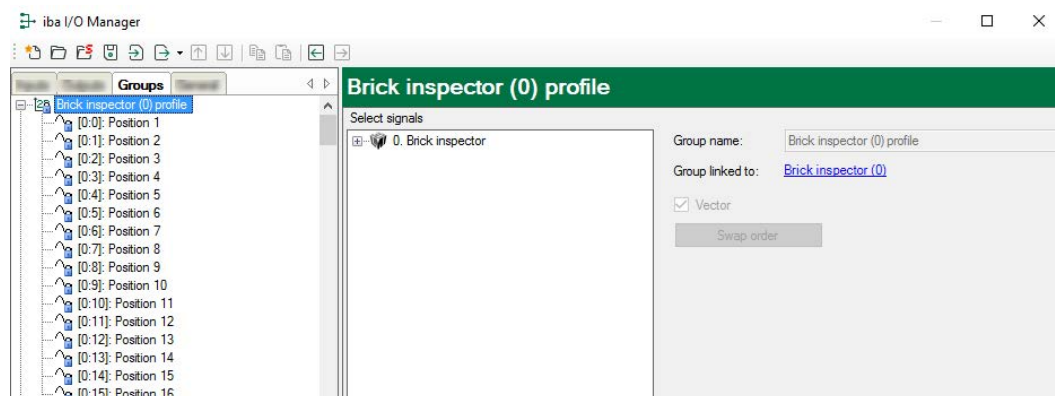
Tip

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

3.3.7 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):
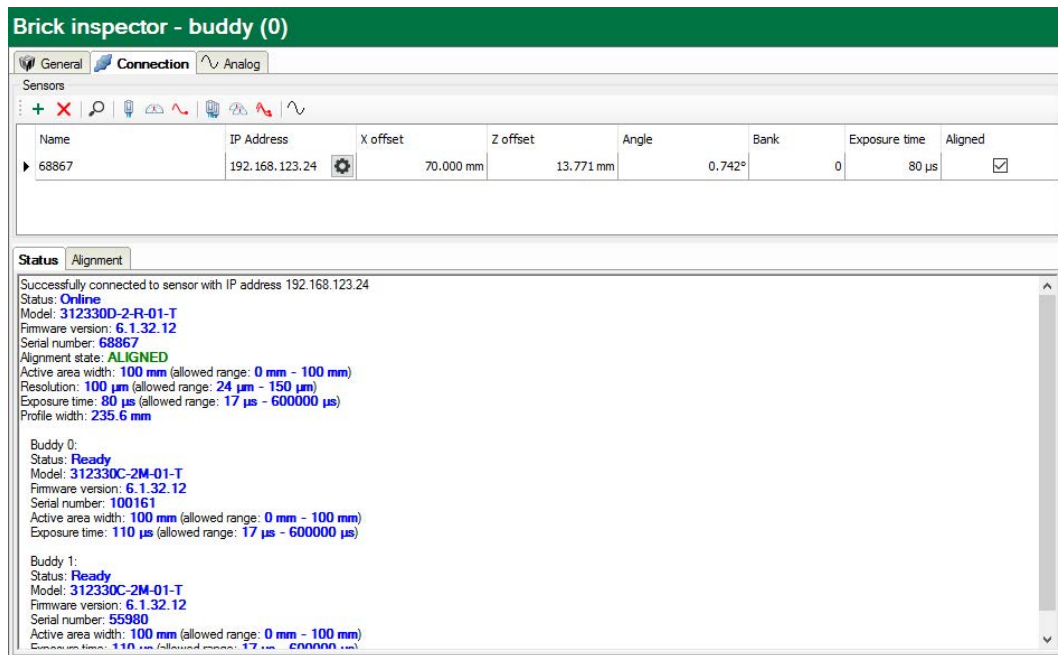
When doing a connection test the status information of the main sensor will be displayed in the *Status* tab along with some basic information concerning its configured buddies.

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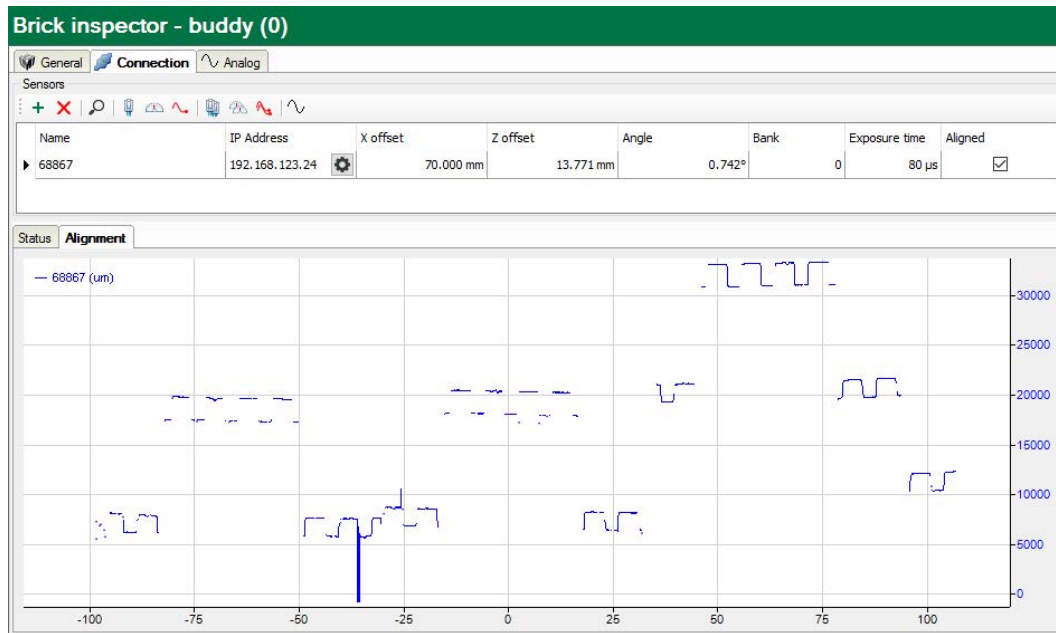
When read-only mode has not been enabled in the Gocator module and you try to acquire a profile or align the sensors, an error will appear in *ibaPDA* indicating that buddy sensors can only be used when read-only mode is enabled.
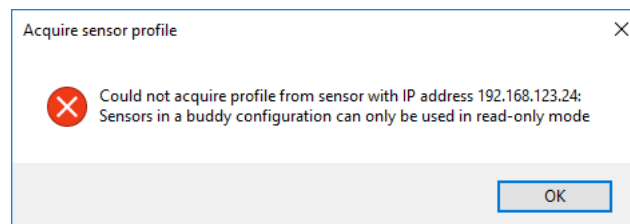
4   Diagnostics

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.

```
Administrator: Command Prompt                        —   □   ×
Microsoft Windows [Version 10.0]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>ping 192.168.1.10

Pinging 192.168.1.10 with 32 bytes of data:
Reply from 192.168.1.10: bytes=32 time=1ms TTL30
Reply from 192.168.1.10: bytes=32 time<1ms TTL30
Reply from 192.168.1.10: bytes=32 time<1ms TTL30
Reply from 192.168.1.10: bytes=32 time<1ms TTL30

Ping statistics for192.168.1.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\Windows\system32>_
```

→   With no existing connection you receive error messages.

```
Administrator: Command Prompt                        —   □   ×
Microsoft Windows [Version 10.0]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>ping 192.168.1.10

Pinging 192.168.1.10 with 32 bytes of data:
Reply from 192.168.1.10: Destination host unreachable.
Reply from 192.168.1.10: Destination host unreachable.
Reply from 192.168.1.10: Destination host unreachable.
Reply from 192.168.1.10: Destination host unreachable.

Ping statistics for 192.168.1.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

C:\Windows\system32>_
```

4.3 Connection table

The LMI-Gocator interface shows a table of the configured LMI-Gocator modules. Each line in the table corresponds to a configured LMI-Gocator module and connection respectively.

The columns in the table and their meaning:

- Name: Name of the module

- Received Messages: 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 is read.
Red	The connection has failed.
Grey	No connection configured.

Table 2: Meaning of background colors

4.4   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 because they do not establish their own connection, but refer to another module.

Example for the use of diagnostic modules:
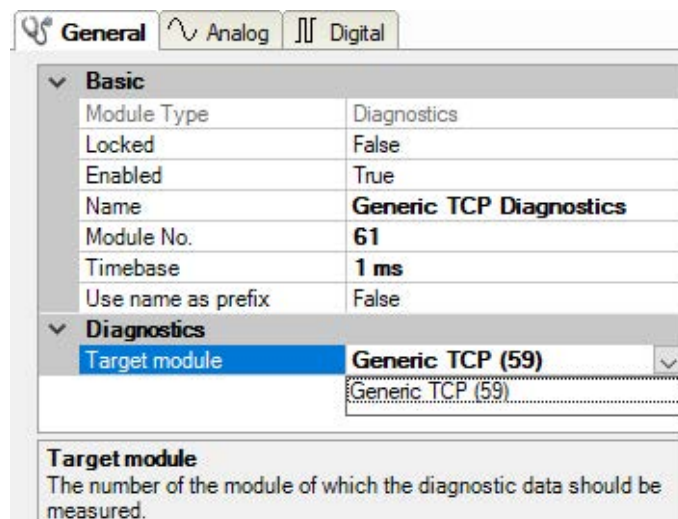
■ A notification can be generated, whenever the error counter of a communication connection exceeds a certain value or the connection gets lost.

■ 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).
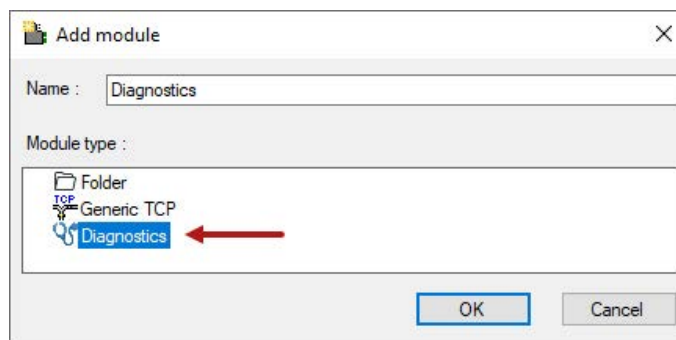
Module settings diagnostic module

For a diagnostic module, you can make the following settings (example: Generic TCP):
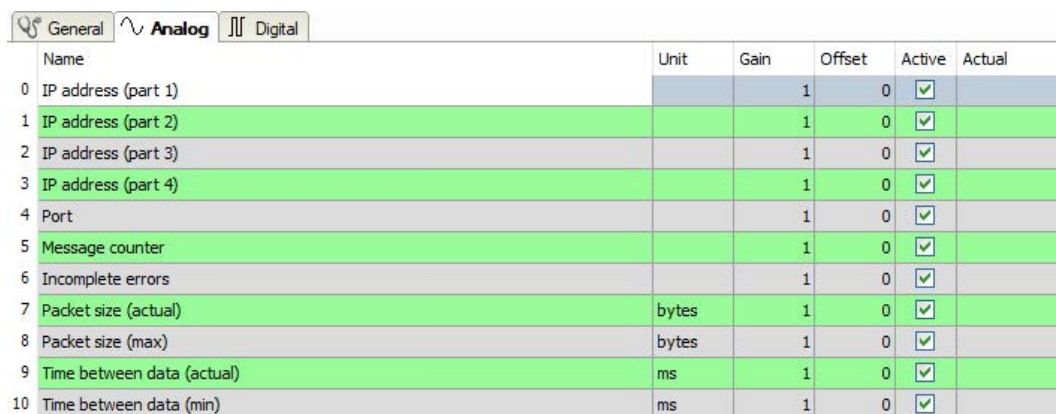
The basic settings of a diagnostic module equal those of other modules.

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.

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
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

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

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

4.5   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.
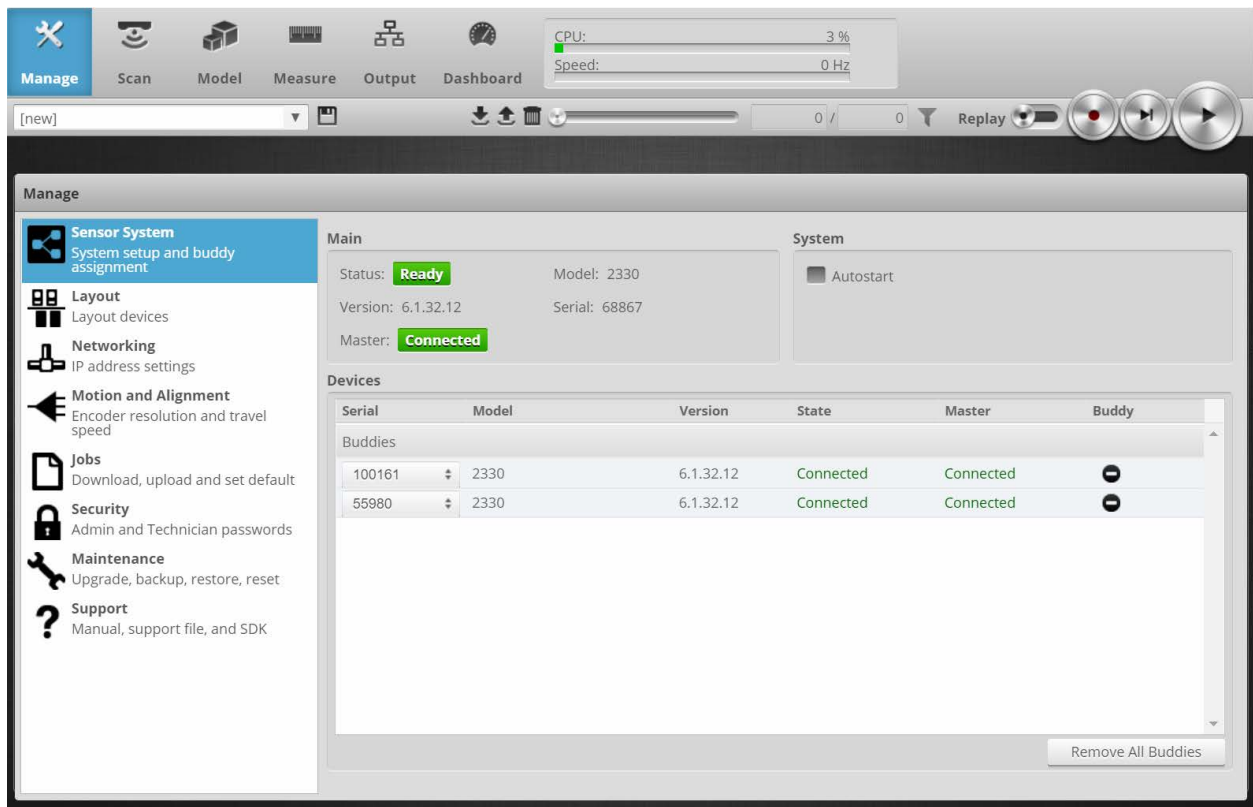
Also verify that all buddies are part of the layout (*Manage → Layout*). It is recommended to activate the "Device Exposure Multiplexing" option, as the banks can be set here, see ↗ *Configuration of sensors*, page 11.
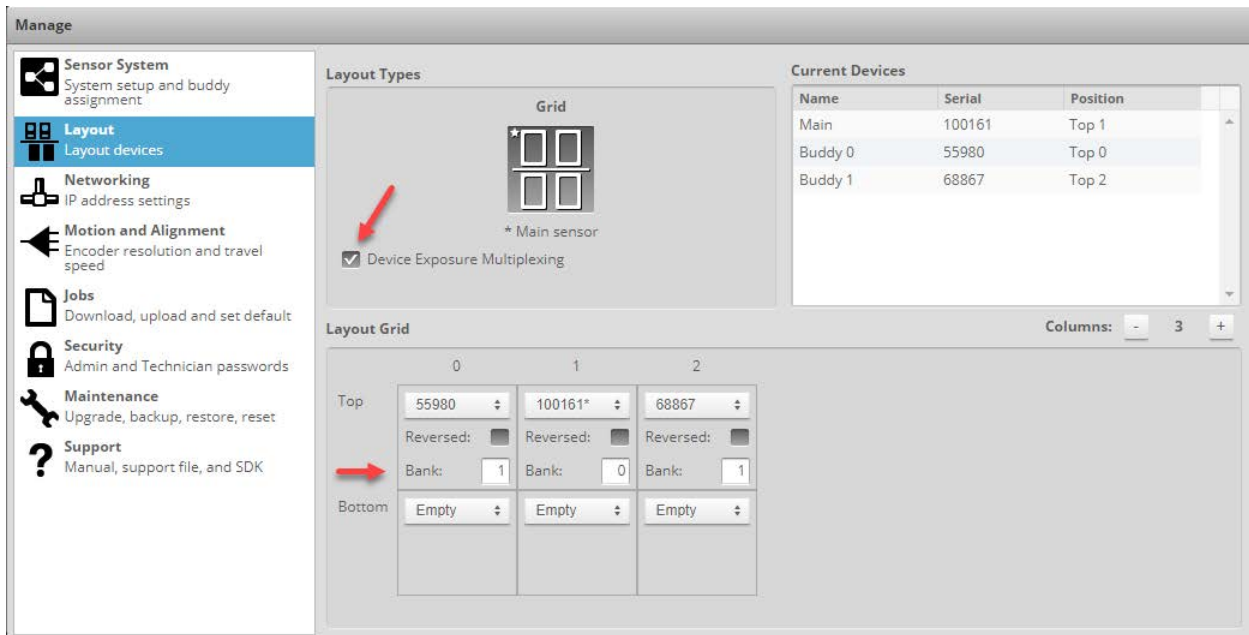
No profile data can be obtained

Make sure that the Top profile is being sent by the (main) sensor.
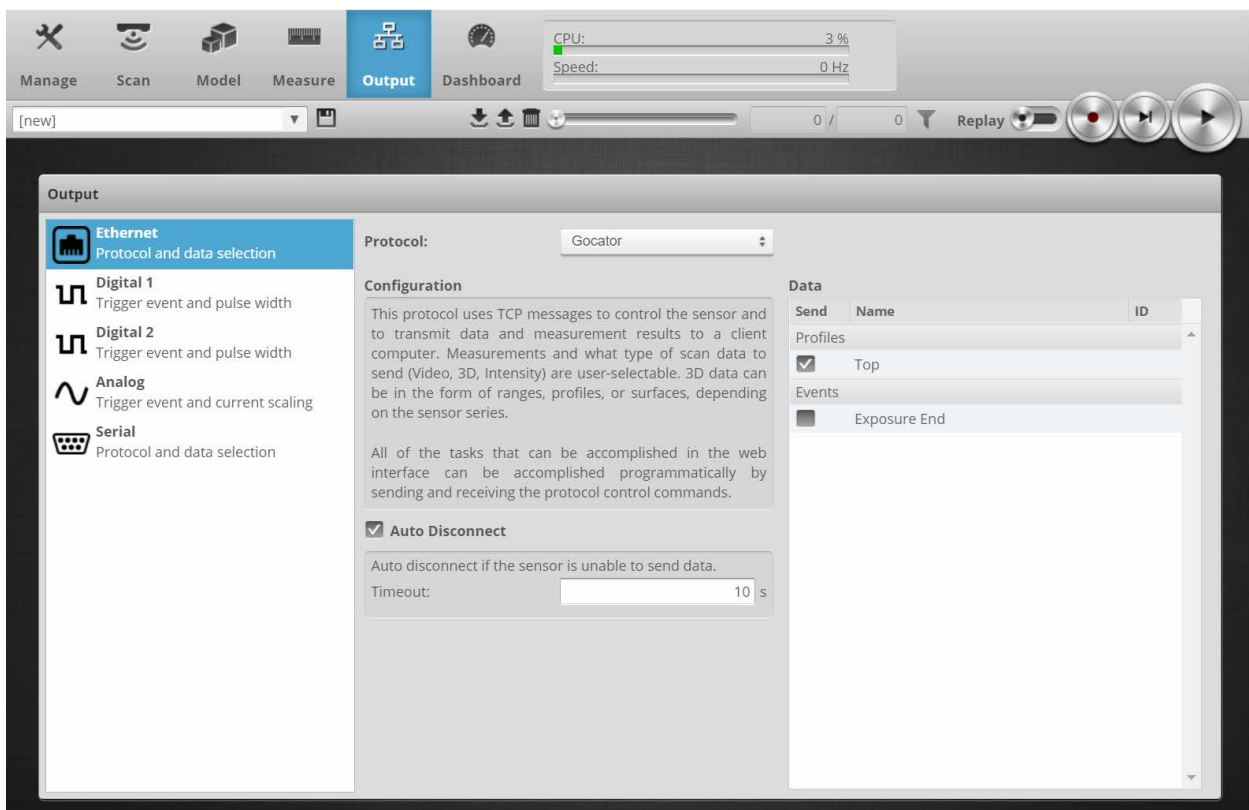
5   Support and contact

Support

Phone: +49 911 97282-14

Fax: +49 911 97282-33

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

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