New Features in ibaPDA v6.38.0

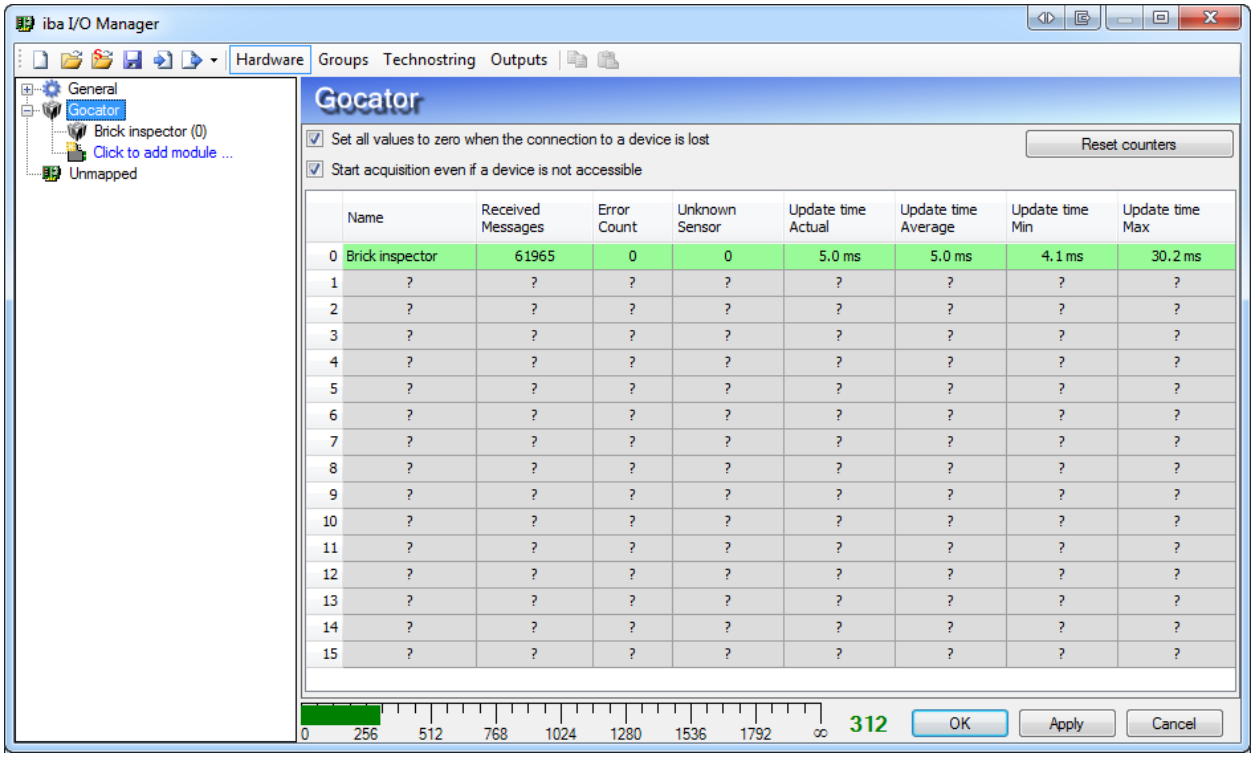
Author: iba Gent

Date: 08/06/2017

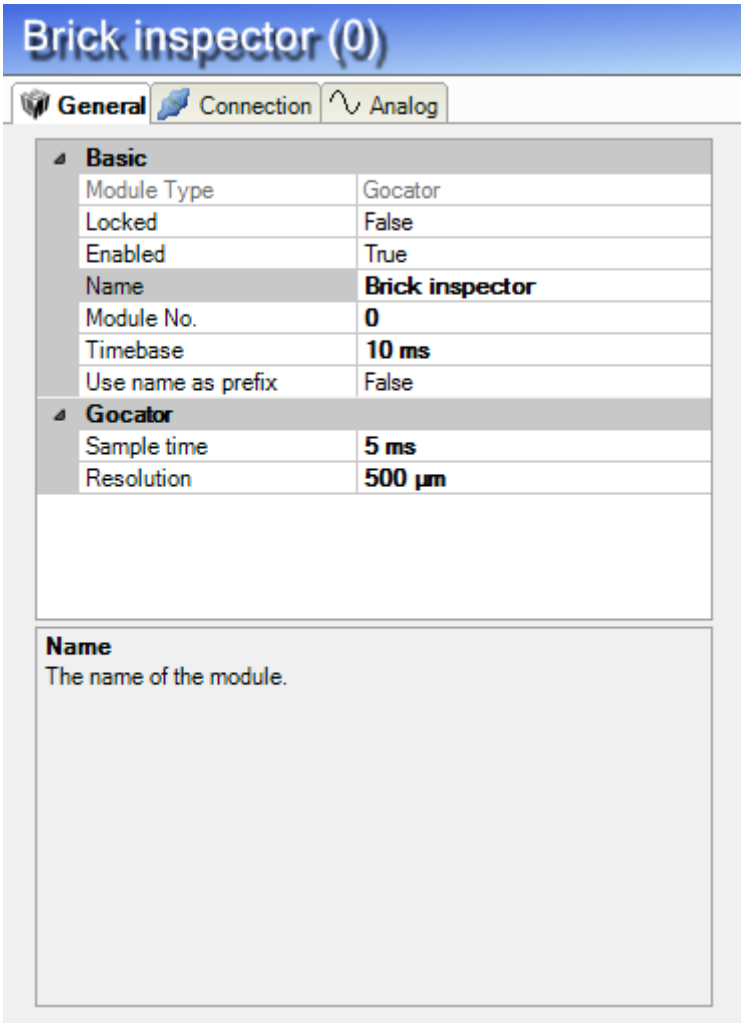
Table of contents

1 LMI Gocator Interface

The Gocator interface in ibaPDA is used to measure profile data from LMI Gocator sensors. Data from several, adjacent sensors can be collected and merged into a single profile.



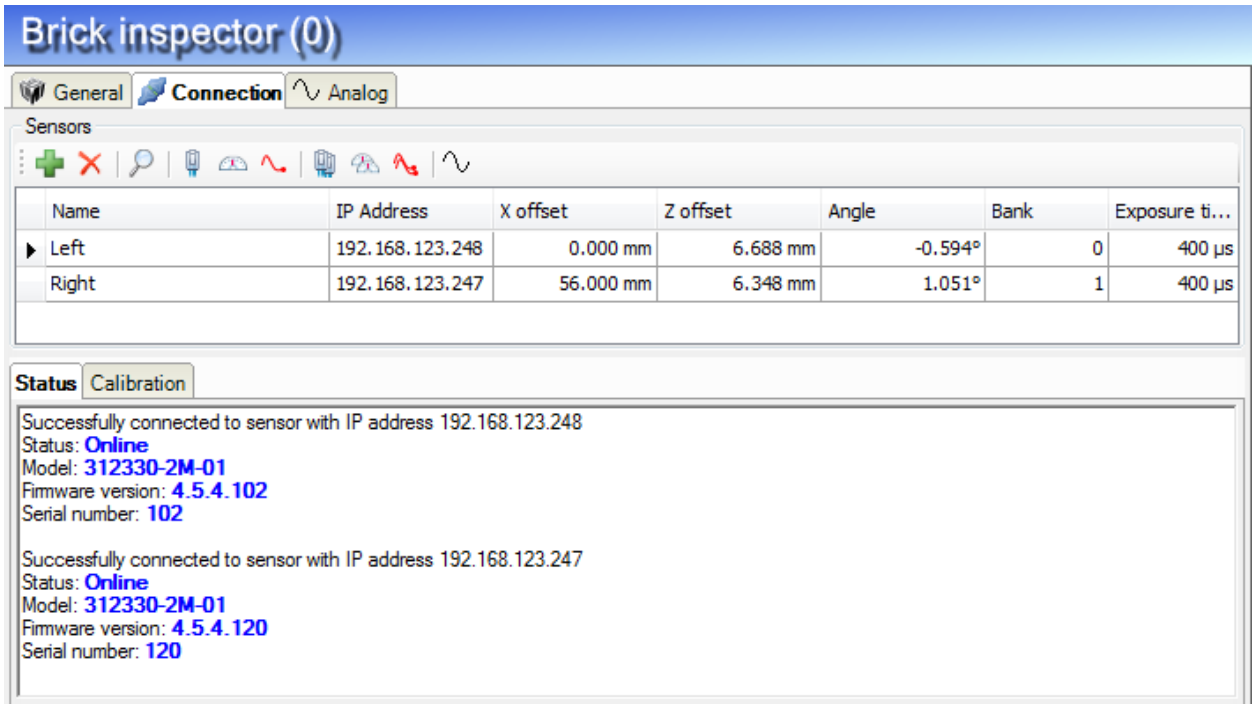
The Gocator interface shows a table of the configured Gocator modules. Each line in the table corresponds to a configured Gocator module where the background color of a properly functioning module is green; in case of a connection error, the background color will change to red.



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

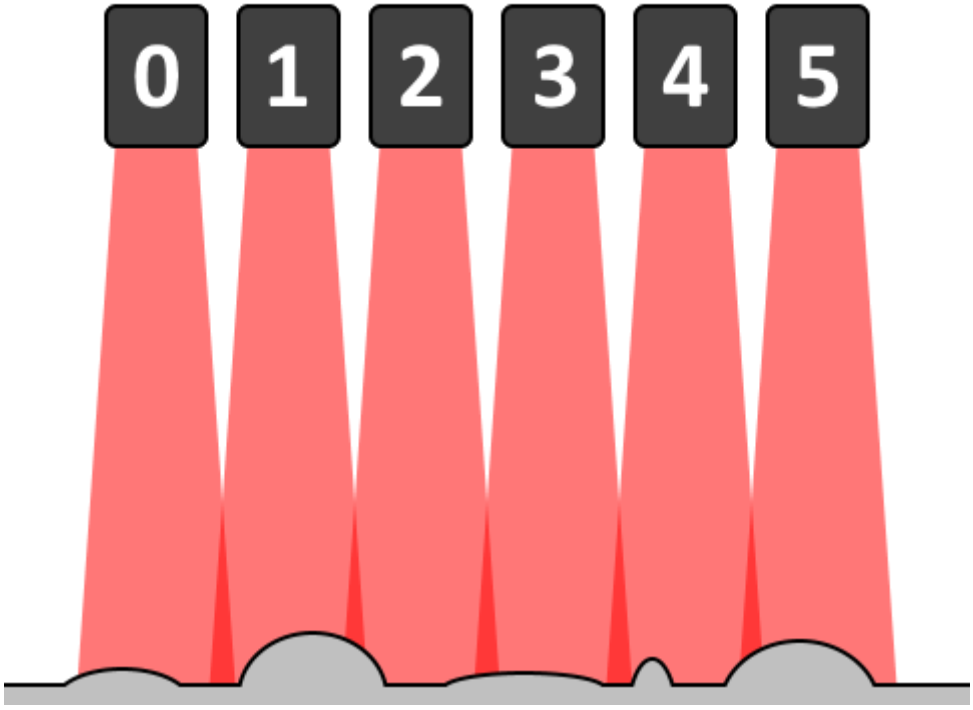
- **Sample time**: this determines how fast profile data is generated by the Gocator sensors. Although in theory, it is possible to set the sample time as low as 1 ms, in practice it might be limited by the sensors
- **Resolution**: this determines the distance in microns between two neighbouring profile 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.

To configure the sensors associated with a module, click the Connection tab.



Sensors can be added and removed manually by using the button and button respectively, though it is typically easier to use the discovery functionality ( ). This will list all available sensors in the network. If you wish not to use one or more of the discovered sensors, simply remove them from the table. 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.
- **X offset**: the offset of a sensor in mm along the axis of the sensor's laser line. This value is retrieved from and stored in the sensor. This value 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 that this 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. Typically this value is obtained by using the calibration 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. Typically this value is obtained by using the calibration function of the sensor (see below).
- **Bank**: a bank is a subgroup of sensors that can generate the laser line (and measure the profile) simultaneously without interfering each other. The following figure illustrates an array of 6 sensors measuring a slab:



In the ideal case where we could precisely position the sensors next to one another, there would be no overlap of laser lines. However, for high features in the slab, the projected width of the laser line of one sensor is smaller than it is for low features. This implies that, in case we would like to cover an entire slab where relatively high and low features can appear, we must position the sensors so that there will be 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 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. Suppose we would like to obtain a full profile every millisecond; this means that we can divide a continuous measurement in time slots of 1 ms. Suppose now that 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 to 400 µs and sensor 1 could do so from 400 to 800 µs which still leaves us with a 200 µs margin. There no longer is 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 sensors 0, 2 and 4 can generate the laser line at the same time (e.g. in the subslot from 0 to 400 µs) and 1, 3 and 5 can generate the laser line at another time (e.g. in the subslot from 400 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 this sensor. This value is retrieved from and stored in the sensor. Typically it is obtained by configuring the sensor in its web interface (by checking the live image).

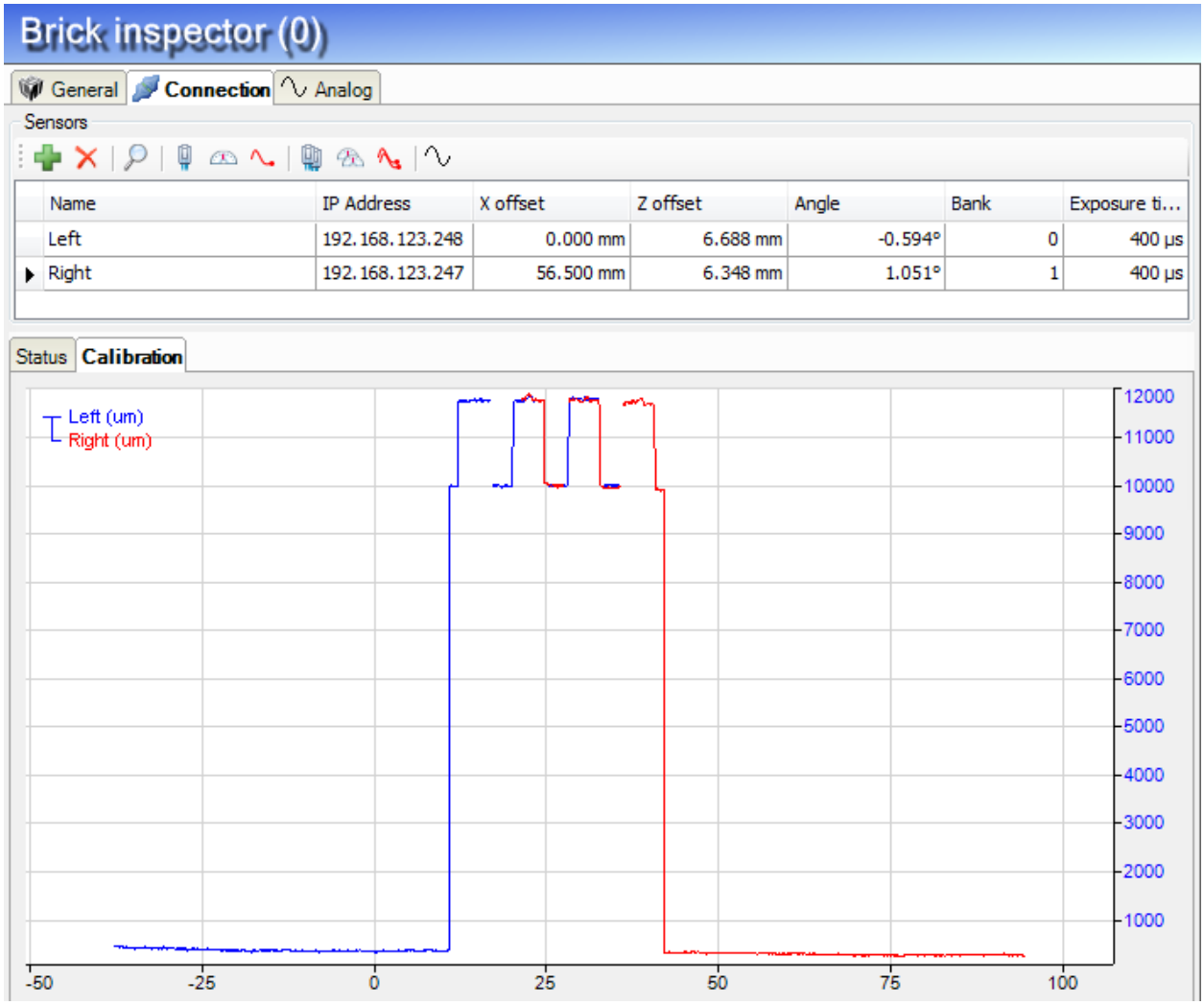
To check whether a connection to a sensor can be created or to obtain some basic diagnostic information, click the button to test the connection to one sensor or the button to test the connection to all listed sensors.

Testing the connection will display the sensor's current status, model, firmware version and serial number in the Status tab.

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 calibrate the selected sensor or the button to calibrate all sensors. The Z offset and Angle will be updated automatically.

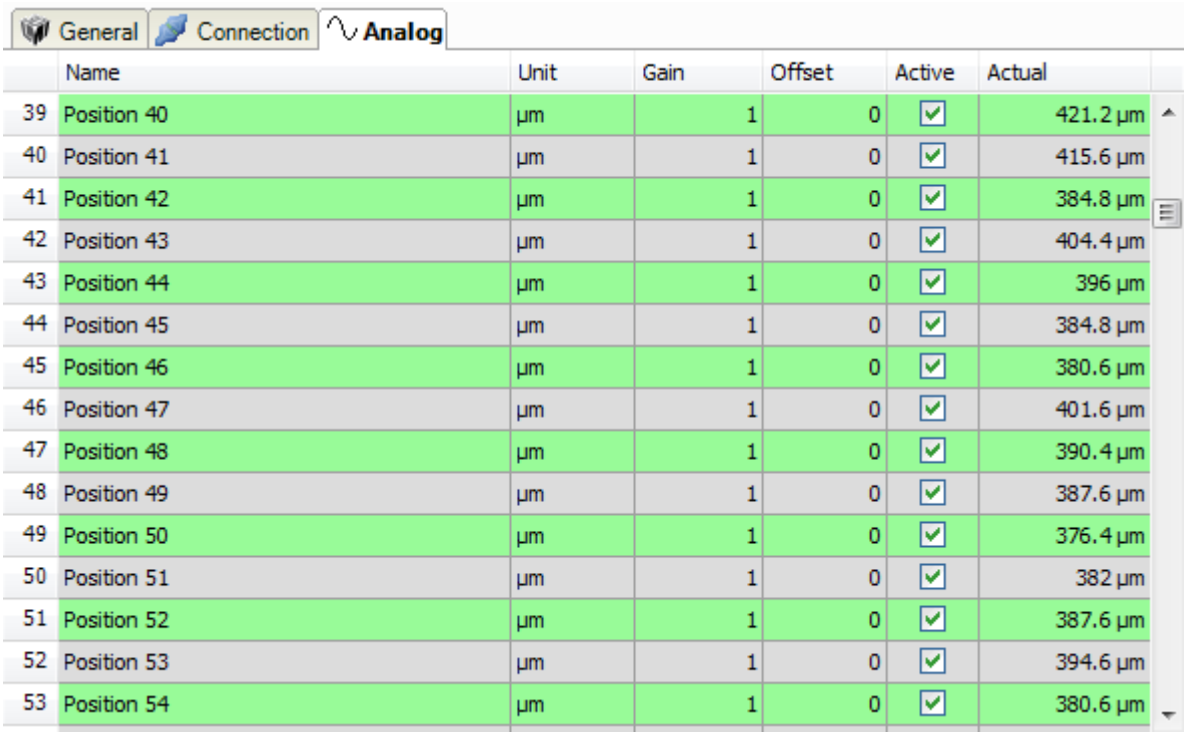
Though the X offset will also be set by calibrating the sensors, this will not be the correct value: the X offset is the distance of one sensor to another reference sensor. Since sensors are unaware of each other, there is no way to determine this automatically. However, the following method can make it easier to get a relatively precise value for this X offset: by clicking the button (for a single sensor) or the button (for all sensors), the current profile is retrieved and displayed in the Calibration tab.

In the graph above, the profile for two calibrated 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 we can measure the distance between these two regions and use that value as the X offset of one of the sensors. When we enter this value in the sensor table and acquire the profiles again we get the following result:



Now we see that the profiles overlap as they should so we can conclude that we have found a correct value for the X offset.

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, where each signal corresponds to a single data point of the entire profile.

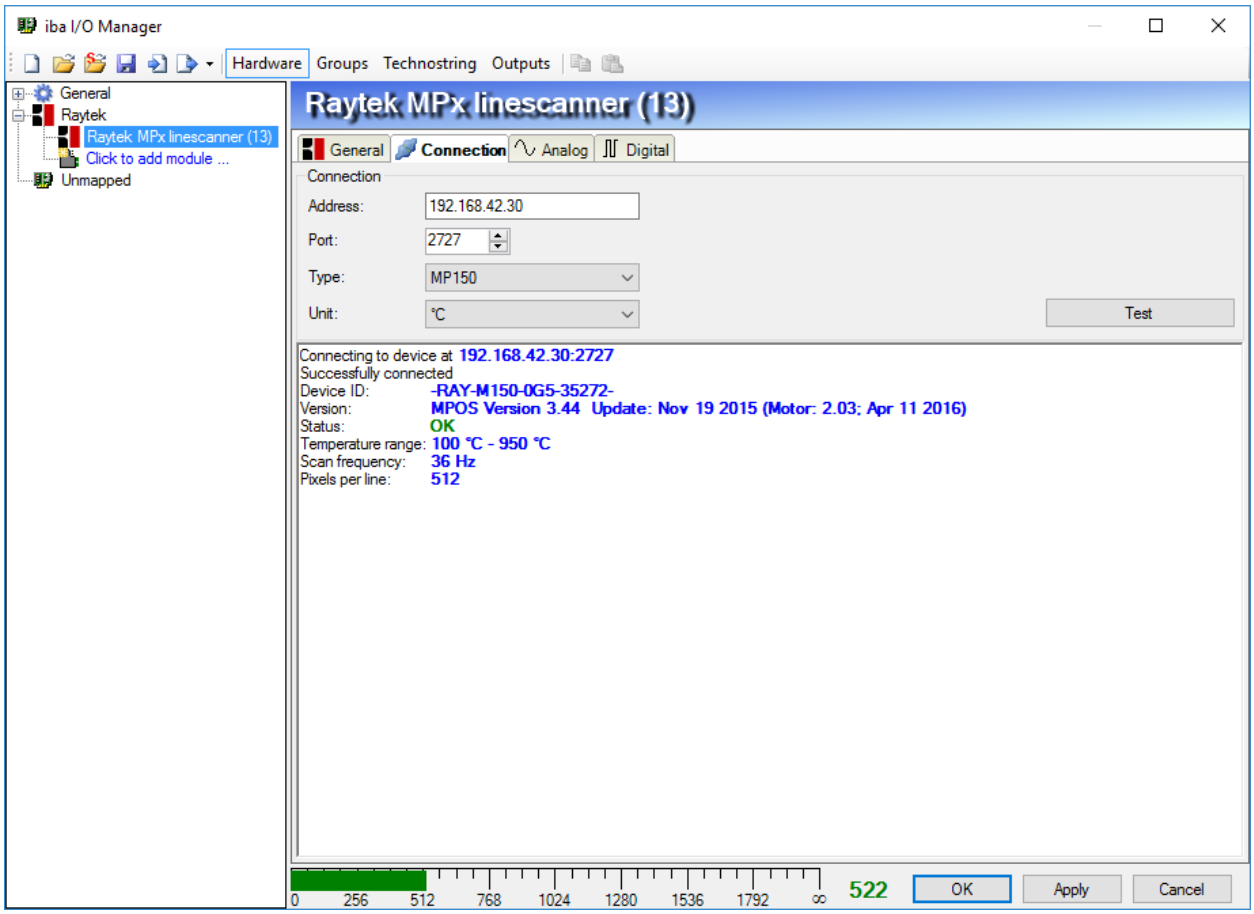


	Name	Unit	Gain	Offset	Active	Actual
39	Position 40	µm	1	0	☑	421.2 µm
40	Position 41	µm	1	0	☑	415.6 µm
41	Position 42	µm	1	0	☑	384.8 µm
42	Position 43	µm	1	0	☑	404.4 µm
43	Position 44	µm	1	0	☑	396 µm
44	Position 45	µm	1	0	☑	384.8 µm
45	Position 46	µm	1	0	☑	380.6 µm
46	Position 47	µm	1	0	☑	401.6 µm
47	Position 48	µm	1	0	☑	390.4 µm
48	Position 49	µm	1	0	☑	387.6 µm
49	Position 50	µm	1	0	☑	376.4 µm
50	Position 51	µm	1	0	☑	382 µm
51	Position 52	µm	1	0	☑	387.6 µm
52	Position 53	µm	1	0	☑	394.6 µm
53	Position 54	µm	1	0	☑	380.6 µm

In the Analog, no further changes need to be made (apart from chaning the signal names, gain and offset if desired).

2 Raytek interface

The Raytek interface can be used to measure data from Raytek (Fluke Process Instruments) temperature line scanners MP40, MP50 and MP150. The scanners can generate up to 1024 pixels per line and they can scan at up to 150 lines per second. The scanners send their data via a TCP connection to ibaPDA. The scanners support only a single connection.

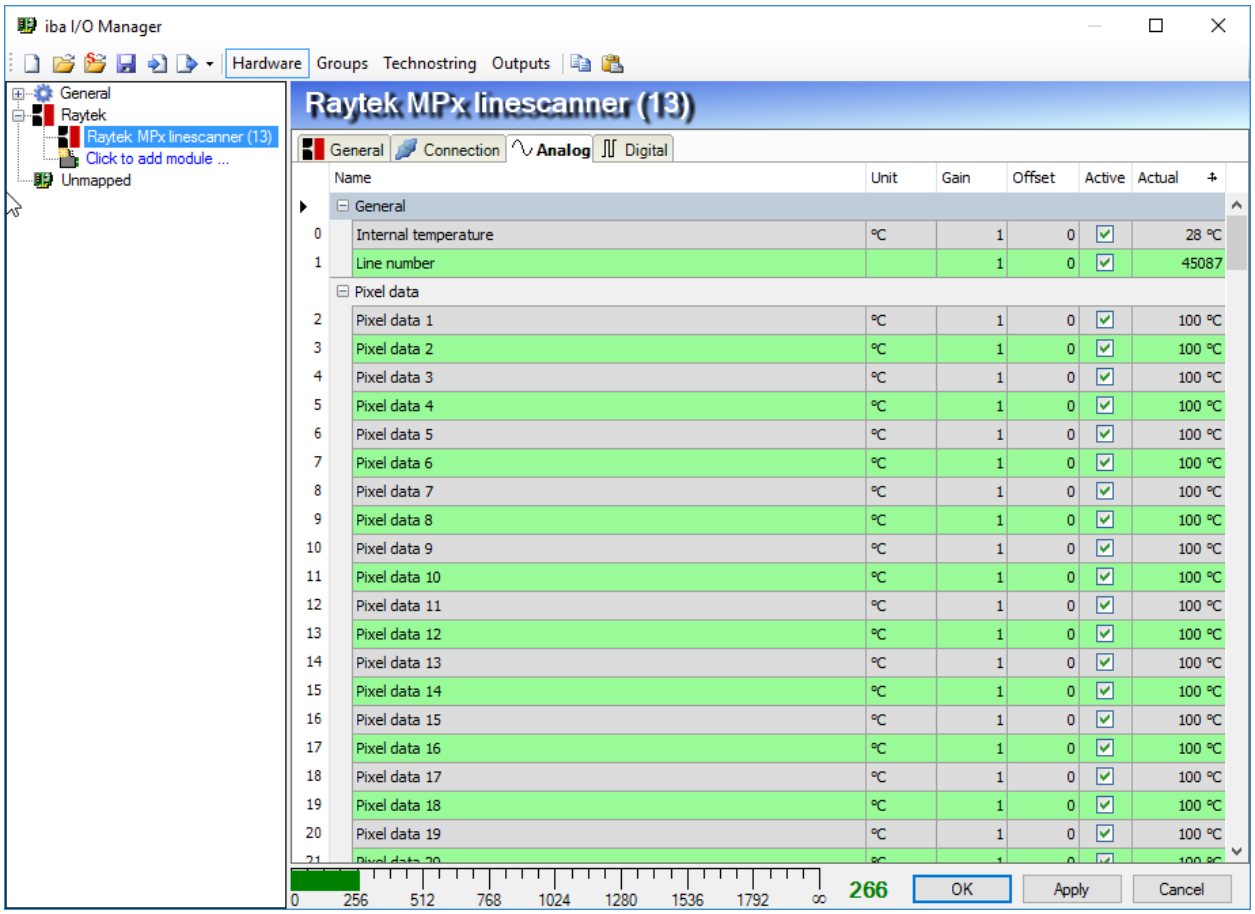


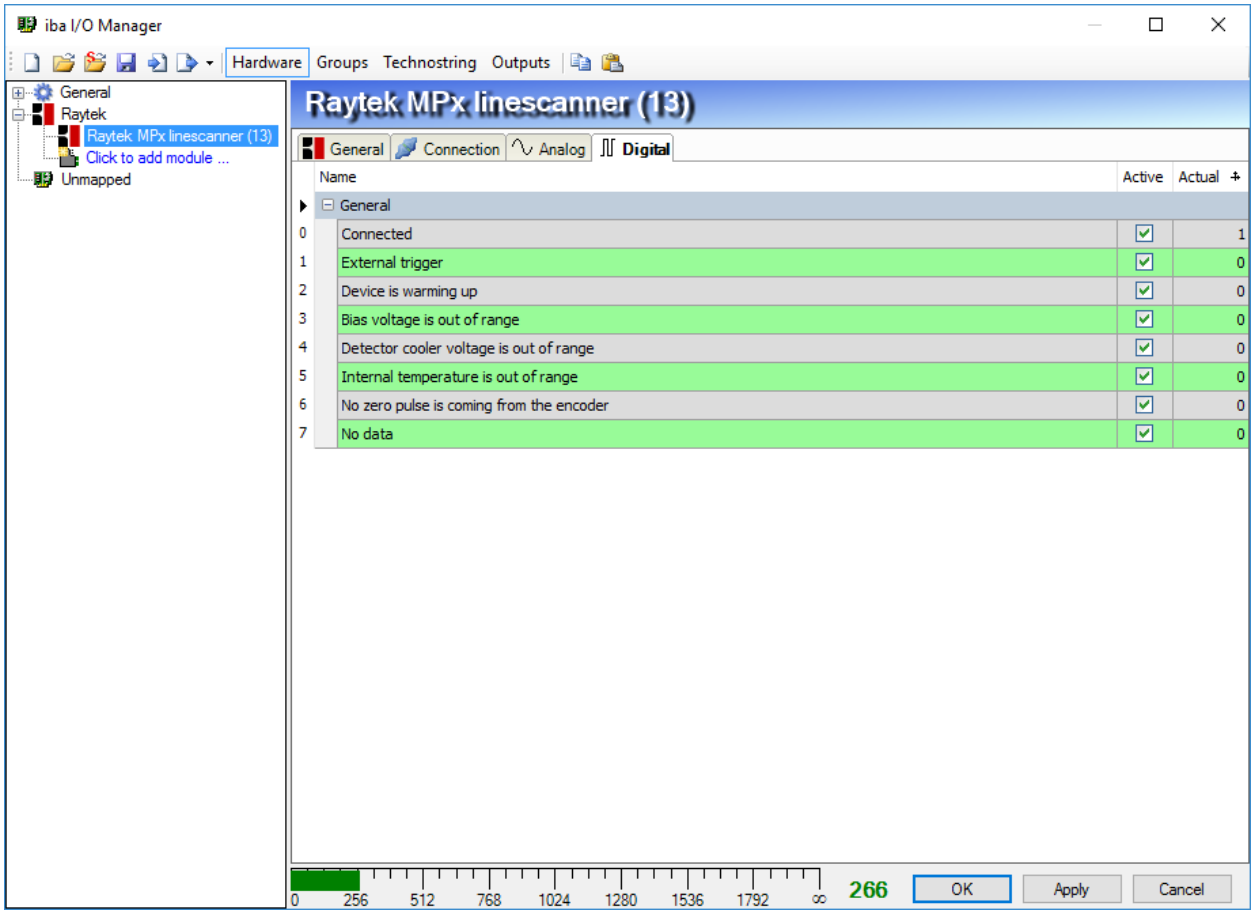
When you have the correct license a Raytek interface will appear in the I/O manager. Under this interface you can add a Raytek module. On the connection tab you must enter the IP address of the scanner and the port number. Normally you don't need to enter the type of the scanner since this will be automatically detected by ibaPDA when you click the *Test* button. You can choose between °C and °F as temperature unit. ibaPDA will use the correct scaling factors for the temperature.

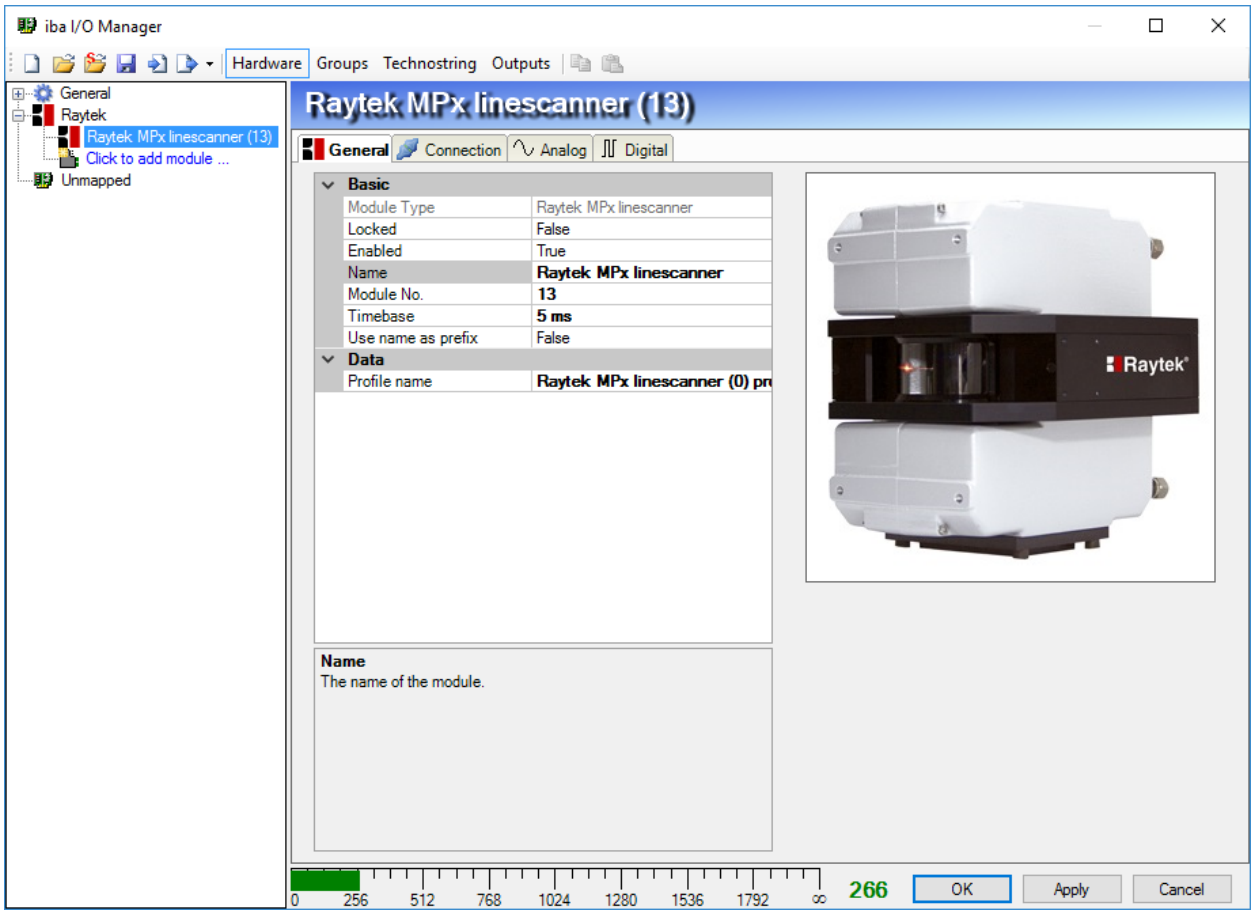
When you click the *Test* button ibaPDA will try to connect to the scanner and read out some information like the device ID, its version, the current status, temperature range scan frequency and the number of pixels per line.

This information is used to adjust the number of analog signals. You have to use the Raytek software if you want to change the number of pixels and the scan frequency.

The module contains all analog and digital signals that the scanner sends. They are grouped by functionality in the signal grids. When the acquisition is running you also see the actual values coming from the scanner.

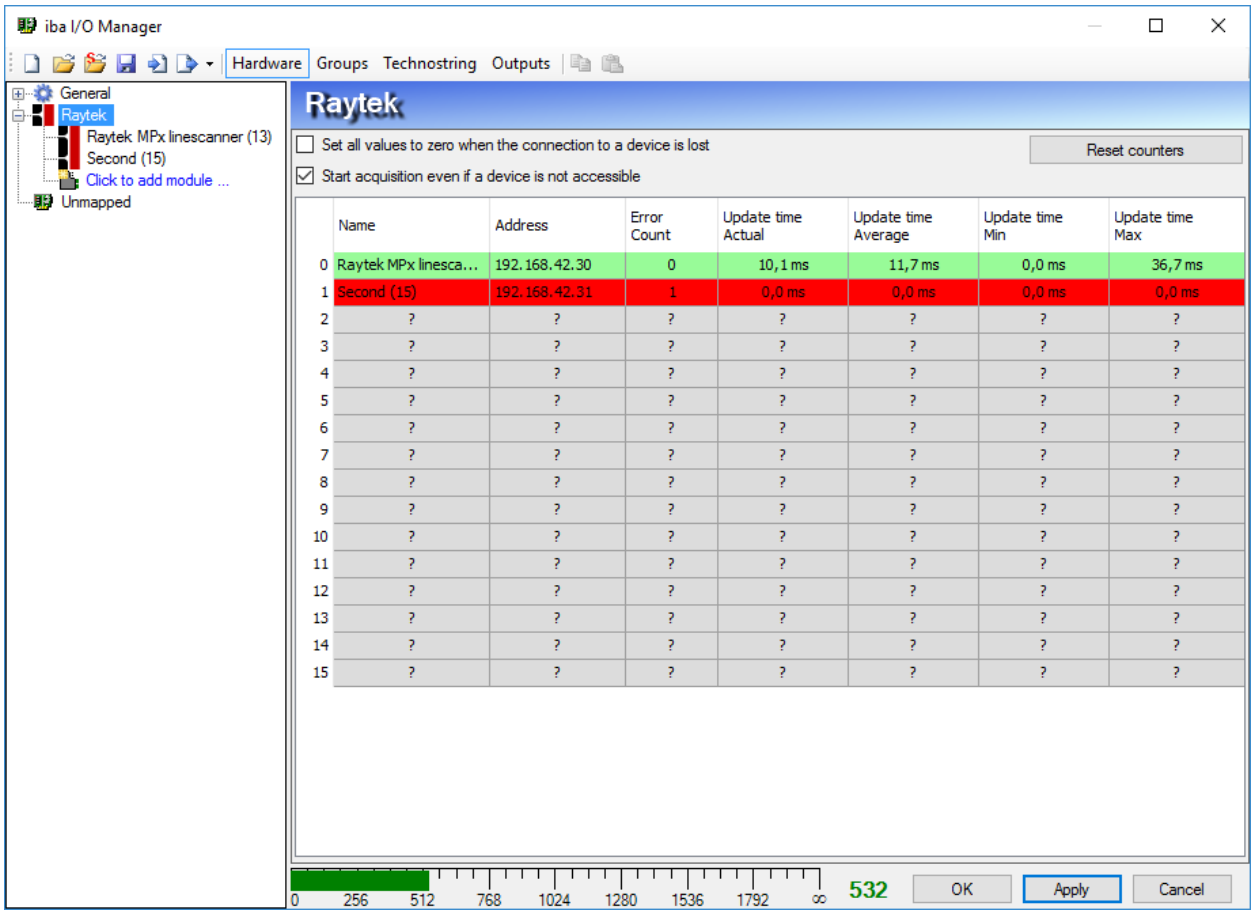

iba I/O Manager
Hardware
Groups
Technostring
Outputs
General
Raytek
Raytek MPx linescanner (13)
Click to add module ...
Unmapped
Raytek MPx linescanner (13)
General
Connection
Analog
Digital
Name
Unit
Gain
Offset
Active
Actual
General
Internal temperature
Line number
Pixel data
Pixel data 1
Pixel data 2
Pixel data 3
Pixel data 4
Pixel data 5
Pixel data 6
Pixel data 7
Pixel data 8
Pixel data 9
Pixel data 10
Pixel data 11
Pixel data 12
Pixel data 13
Pixel data 14
Pixel data 15
Pixel data 16
Pixel data 17
Pixel data 18
Pixel data 19
0 256 512 768 1024 1280 1536 1792 ∞
266
OK
Apply
Cancel


iba I/O Manager
Hardware
Groups
Technostring
Outputs
General
Raytek
Raytek MPx linescanner (13)
Click to add module ...
Unmapped
Raytek MPx linescanner (13)
General
Connection
Analog
Digital
Name
Active
Actual
General
Connected
External trigger
Device is warming up
Bias voltage is out of range
Detector cooler voltage is out of range
Internal temperature is out of range
No zero pulse is coming from the encoder
No data
0 256 512 768 1024 1280 1536 1792 ∞
266
OK
Apply
Cancel



There is also a vector generated for the pixel data. You can determine the name of this profile vector by setting the *Profile name* property on the general tab. You can use the '\' character to put the profile vector in a subgroup.

On the general tab you can also configure the timebase that will be used to measure the scanner's data. The scanner itself sends the data each time a line is finished. ibaPDA just receives the data. It doesn't have to request the data each time.



The Raytek interface shows a connection grid. There is one row per connection to a scanner. Each Raytek license allows 2 scanners. A maximum of 16 scanners is supported.

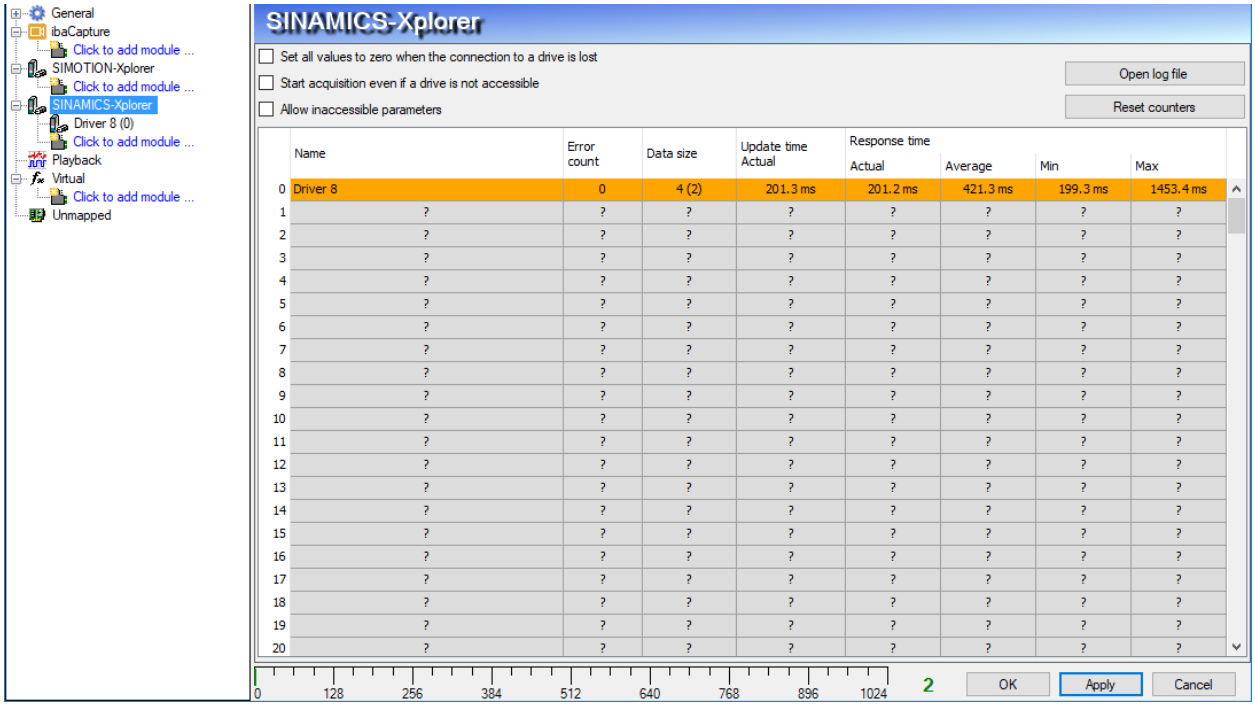
A row is green when the connection is ok and it is red when there is no connection.

These are the columns in the grid:

- Name: Name of the module.
- Address: IP address of the scanner.
- Error count: The number of communication errors that occurred.
- Update time actual, average, min, max: The update time is the time between consecutive data messages. It should correspond to the configured scan frequency in the scanner.

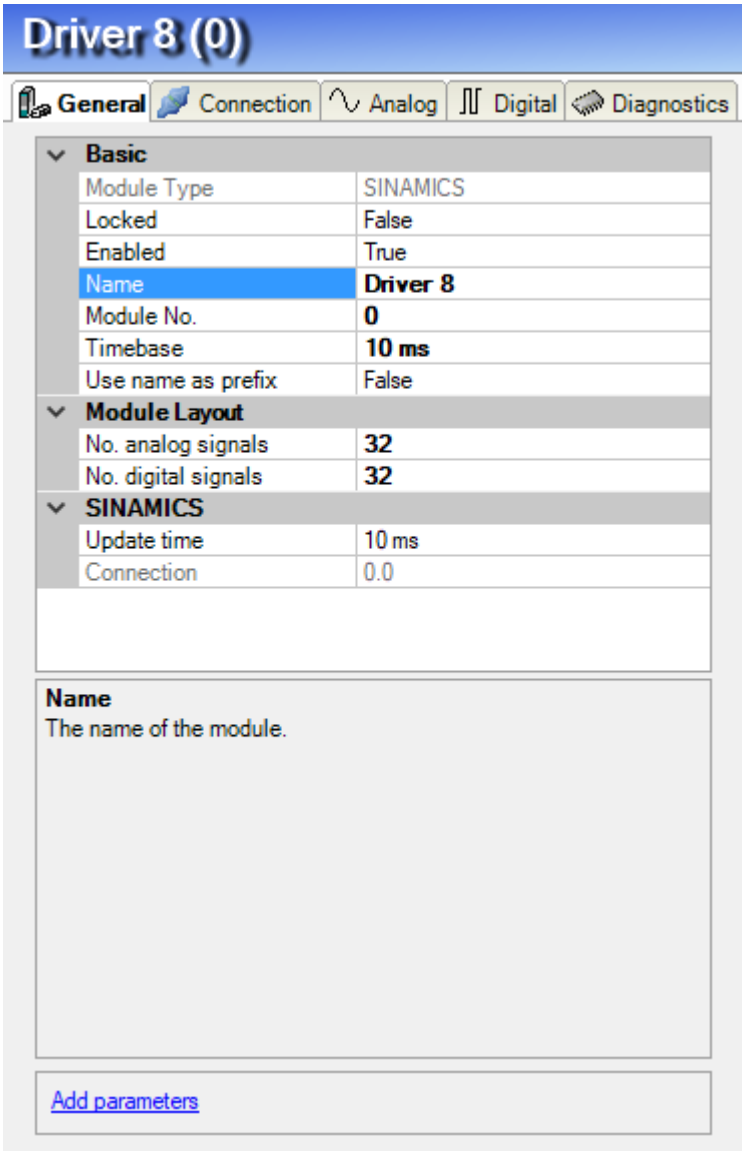
3 SINAMICS-Xplorer

The SINAMICS-Xplorer interface can be used to measure data from Siemens SINAMICS drives. For each SINAMICS-Xplorer license you can connect to up to 32 SINAMICS drives. You can purchase a total of 16 licenses which brings the total maximum amount of SINAMICS-Xplorer connections to 512.

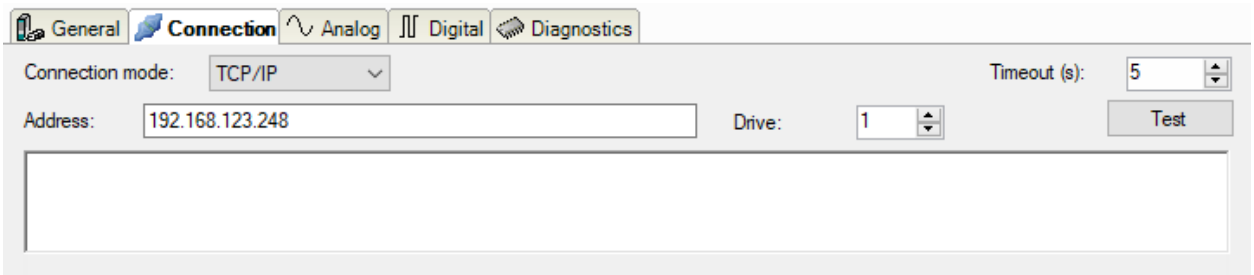


The SINAMICS-Xplorer interface node contains a table listing all configured SINAMICS-Xplorer connections along with some connection status information. The Data size column displays the amount of requested bytes as well as the number of request messages (value displayed between brackets; for support cases only).

In case of a stable connection to the SINAMICS drive, the corresponding entry in the table will have a green background. If the actual update time is higher than the configured one, the background will be orange indicating that the data is coming in fine but at a slower rate than expected. A red background color indicates a connection failure.

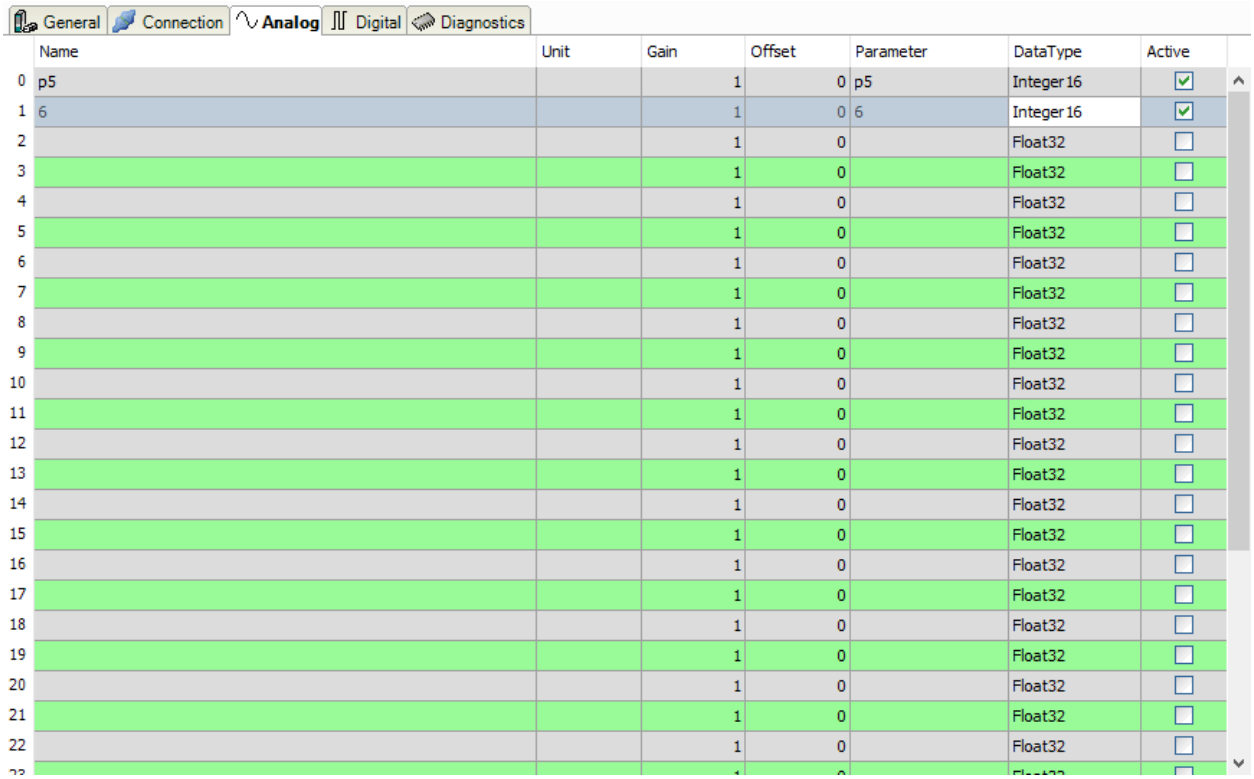


In the General tab of a SINAMICS-Xplorer module you can configure, apart from the general module settings, how fast ibaPDA should request new data from the SINAMICS drive. A read-only Connection ID is displayed as well and can be useful for support cases.



In the Connection tab, it is possible to choose a connection using TCP/IP or PC/CP. The IP address and drive number can be configured here as well. Both values can be obtained by accessing the running SINAMICS drive configuration using e.g. Siemens SCOUT.

By clicking the Test button, a connection is established to the SINAMICS drive. Note that the validity of the drive number is not checked at this point; this can only be verified at the start of the acquisition.



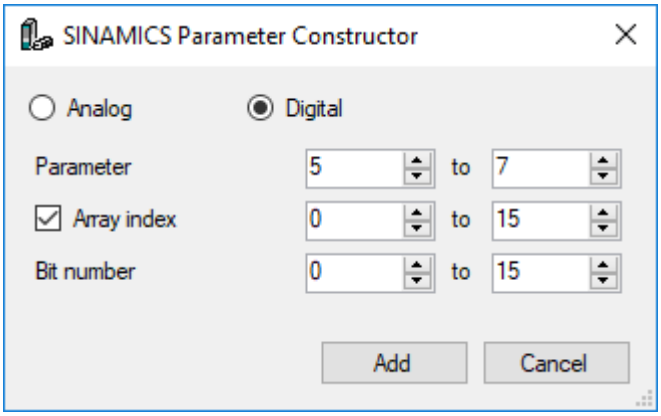
	Name	Unit	Gain	Offset	Parameter	DataType	Active
0	p5		1	0	p5	Integer16	☑
1	6		1	0	6	Integer16	☑
2			1	0		Float32	☐
3			1	0		Float32	☐
4			1	0		Float32	☐
5			1	0		Float32	☐
6			1	0		Float32	☐
7			1	0		Float32	☐
8			1	0		Float32	☐
9			1	0		Float32	☐
10			1	0		Float32	☐
11			1	0		Float32	☐
12			1	0		Float32	☐
13			1	0		Float32	☐
14			1	0		Float32	☐
15			1	0		Float32	☐
16			1	0		Float32	☐
17			1	0		Float32	☐
18			1	0		Float32	☐
19			1	0		Float32	☐
20			1	0		Float32	☐
21			1	0		Float32	☐
22			1	0		Float32	☐

In the Analog tab you can configure which parameters should be read from the SINAMICS drive. The index of the parameter should be entered in the Parameter column. As can be seen in the above figure, it is allowed to add the prefix 'p' or 'r' to the index since this is the commonly used notation for SINAMICS drive parameters. However, these prefixes are ignored by ibaPDA. Apart from the parameter index, you should also set the correct datatype. You can choose one of the following types:

- Unsigned8
- Integer16
- Unsigned16
- Integer32
- Unsigned32
- Float32

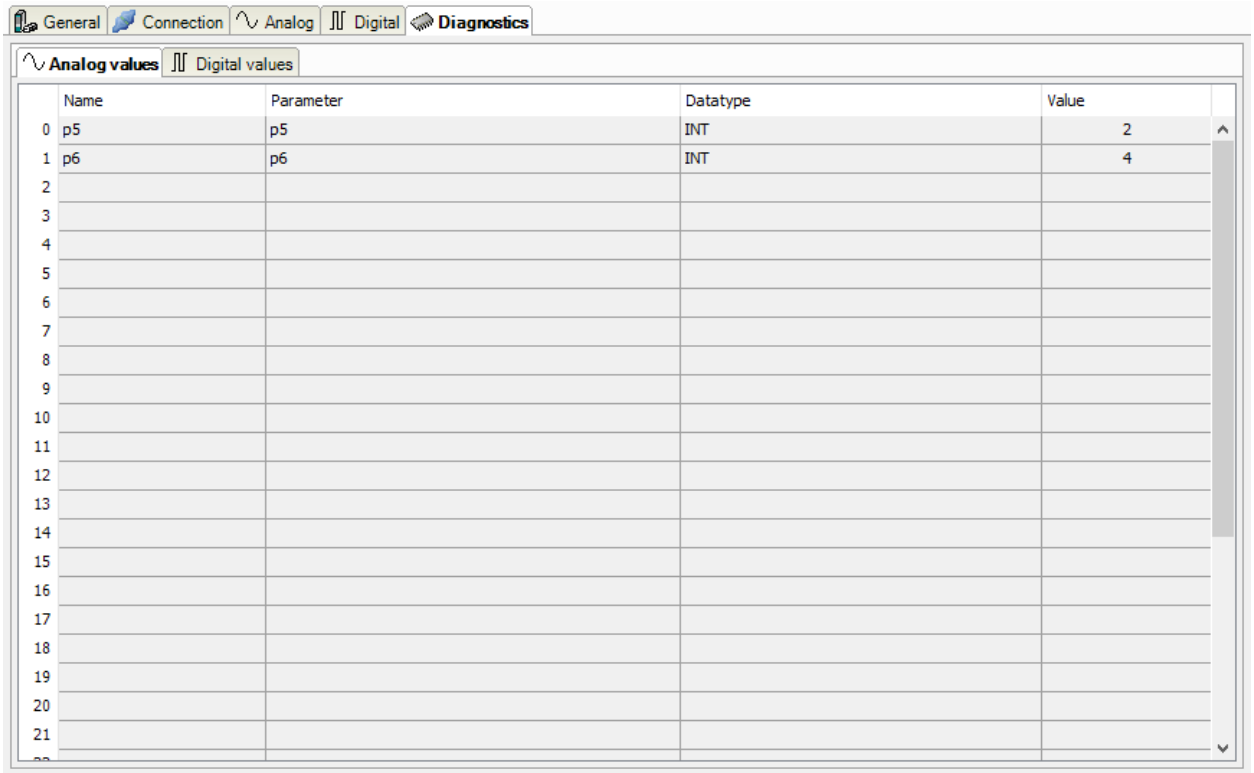
To access an array member, add the array index between square brackets to the parameter index. For example: p5[0] will return the 0-th element of the array which is represented by parameter 5.

In a similar way, digital values can be configured in the Digital tab. Apart from supplying the parameter index, you should also indicate which bit position of the requested value you want to measure.



It is possible to add a range of parameters in the Analog or Digital signal tab by going to the General tab and clicking the link Add parameters at the bottom left.

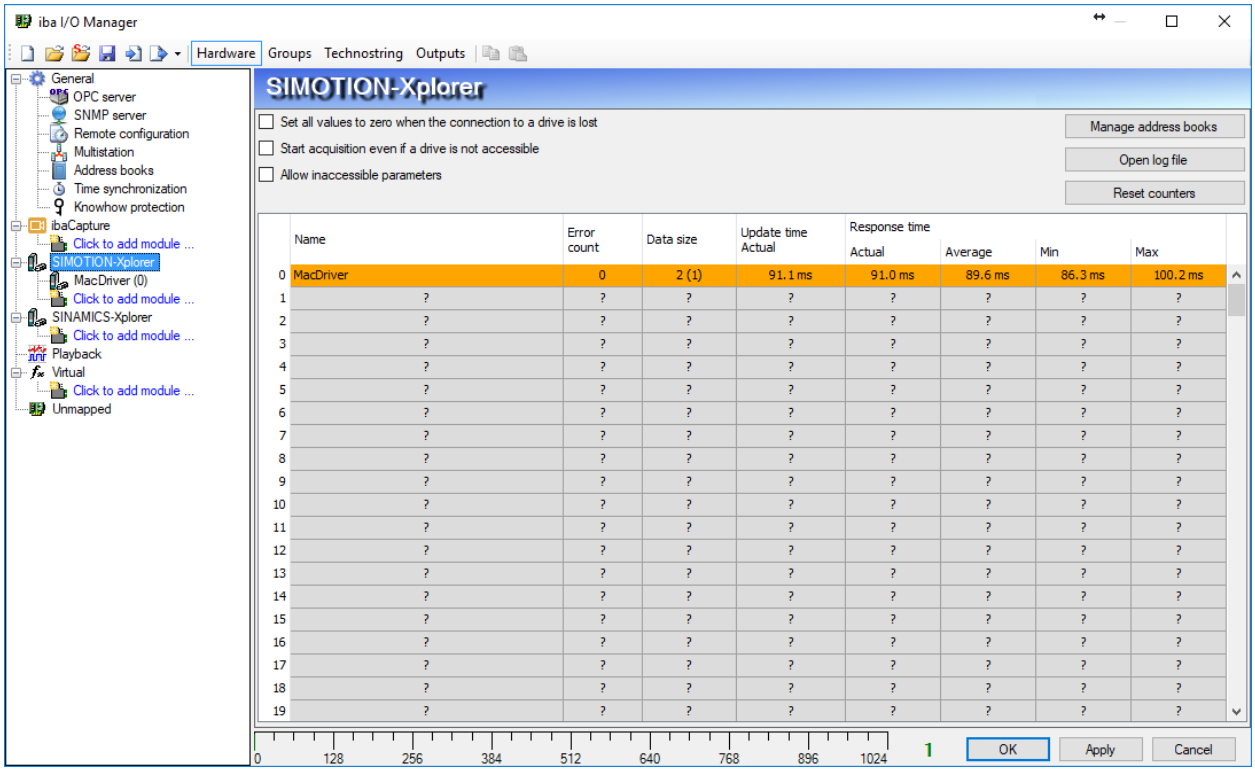
In the dialog that appears, first select whether the parameters should be added to the Analog or Digital tab. Next, enter the start and end of the parameter range you want to add. By checking the Array index box, you can also add (sub)arrays. Finally, in case of digital signals, you can also add a range of bits. In the above example, the following signals would be added: bits 0 to 15 of array members 0 to 15 of parameters 5 to 7 resulting in a total of 768 digital signals.



In the Diagnostics tab, the requested analog and digital signals are displayed along with their current values.

4 SIMOTION-Xplorer

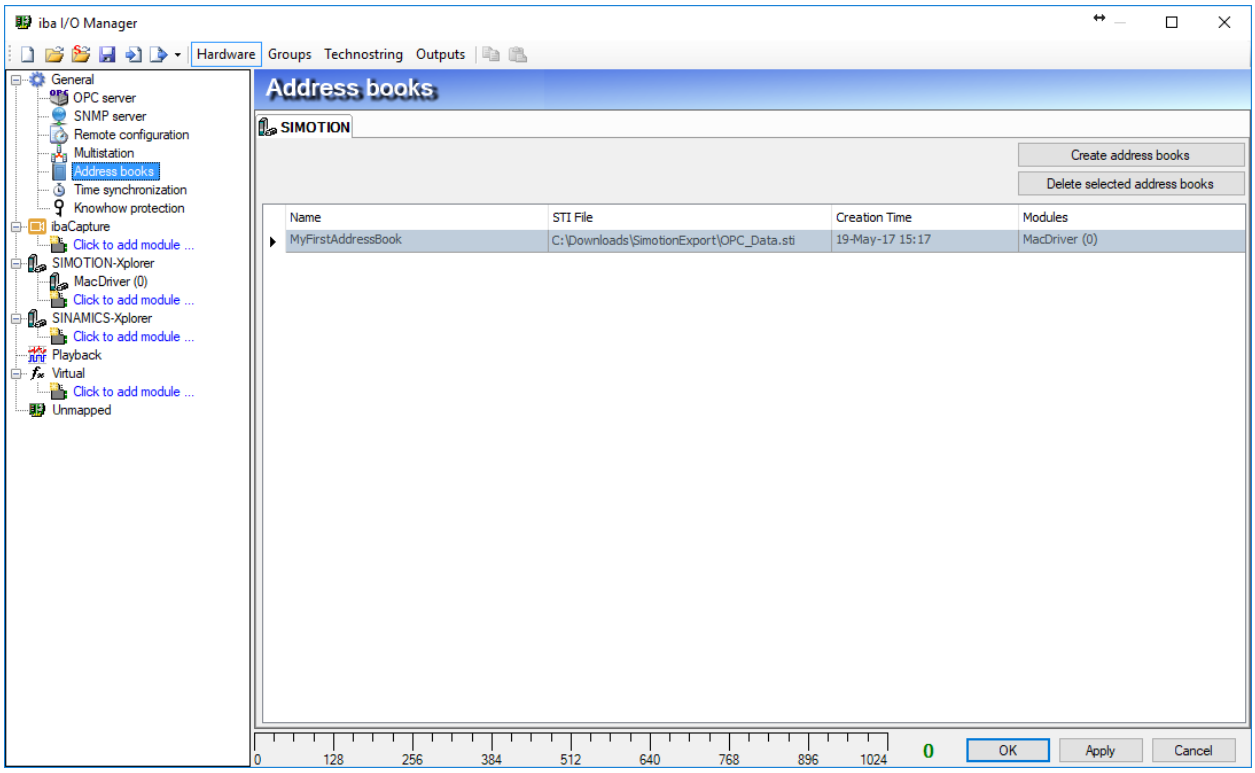
The SIMOTION-Xplorer interface can be used to measure data from Siemens SIMOTION drives. For each SIMOTION-Xplorer license you can connect to up to 32 SIMOTION drives. You can purchase a total of 16 licenses which brings the total maximum amount of SIMOTION-Xplorer connections to 512.



The SIMOTION-Xplorer interface node contains a table listing all configured SIMOTION-Xplorer connections along with some connection status information. The Data size column displays the amount of requested bytes as well as the number of request messages (value displayed between brackets; for support cases only).

In case of a stable connection to the SIMOTION drive, the corresponding entry in the table will have a green background. If the actual update time is higher than the configured one, the background will be orange indicating that the data is coming in fine but at a slower rate than expected. A red background color indicates a connection failure.

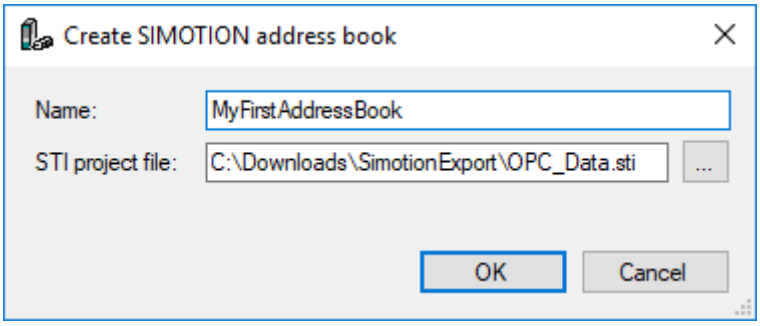
By clicking the Manage address books button, the SIMOTION tab of the Address books node (a subnode of the General node in the I/O manager) will be displayed.



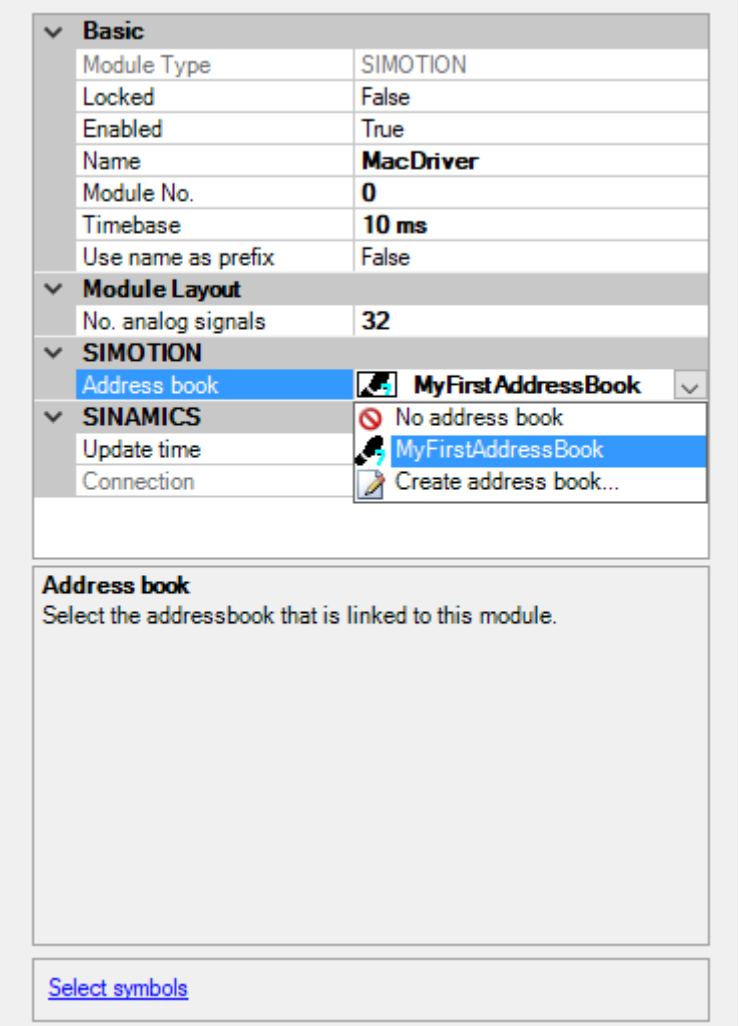
A list of all imported address books is shown here. For each address book entry a user-defined name, the path to the original STI file from which the address book is generated, the time the address book was imported in ibaPDA and the modules currently using the address book are displayed.

When exporting the SIMOTION configuration to an STI file, make sure the option “Arrays with single elements” is enabled.

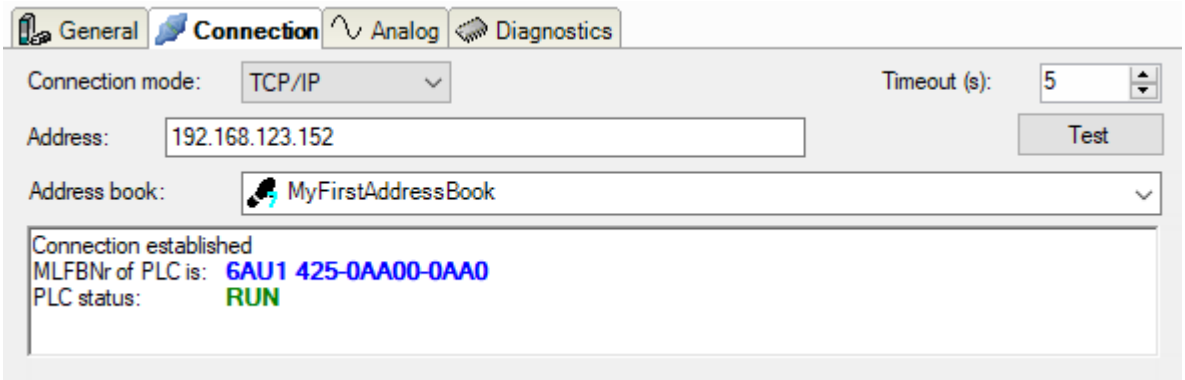
An address book can be added to the list by clicking the Create address books button.



SIMOTION address books are generated based on an STI file which is exported by SIMOTION configuration software like Siemens SCOUT. Enter the path to the exported STI file (or browse for it) and supply a name for the new address book. After clicking OK, ibaPDA will proceed to import the SIMOTION symbols from the STI project file.

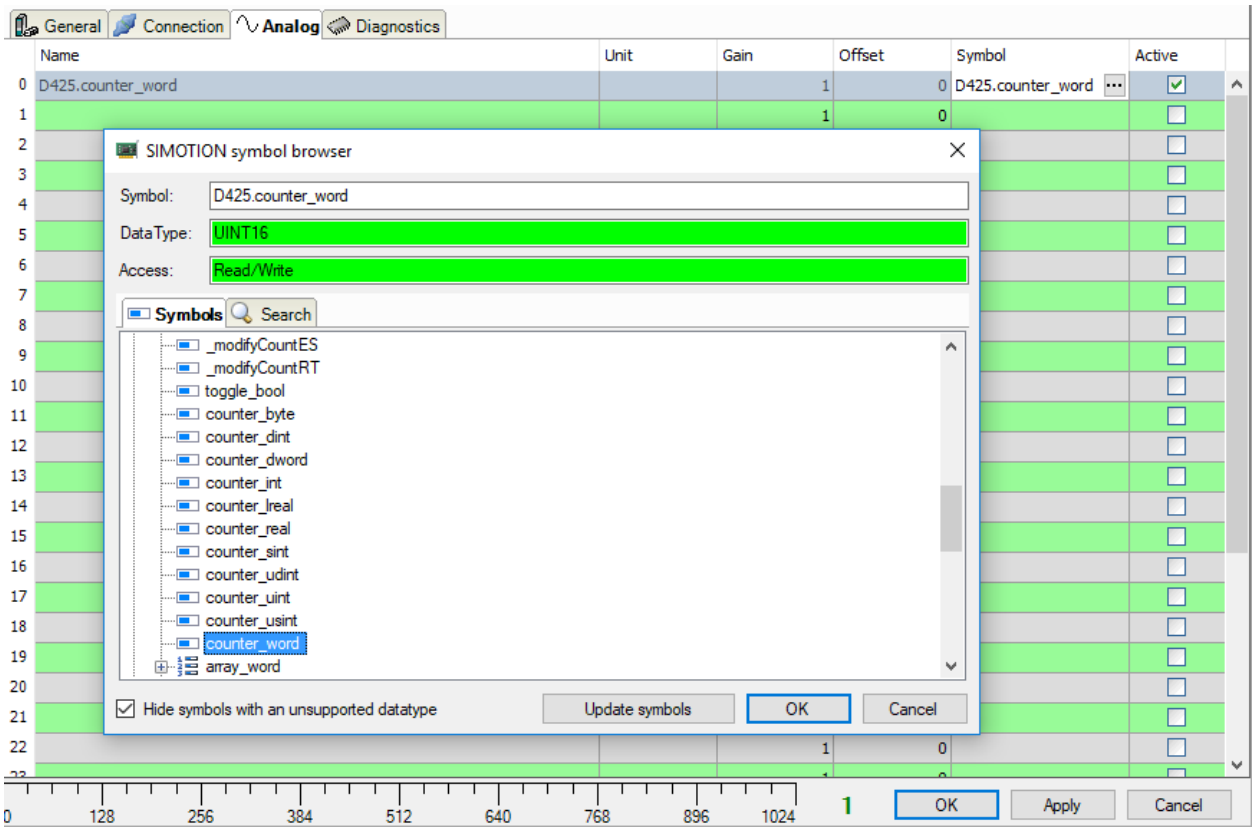


In the General tab of a SIMOTION-Xplorer module you can configure, apart from the general module settings, how fast ibaPDA should request new data from the SIMOTION drive and which SIMOTION address book should be used for this module. A read-only Connection ID is displayed as well and can be useful for support cases.

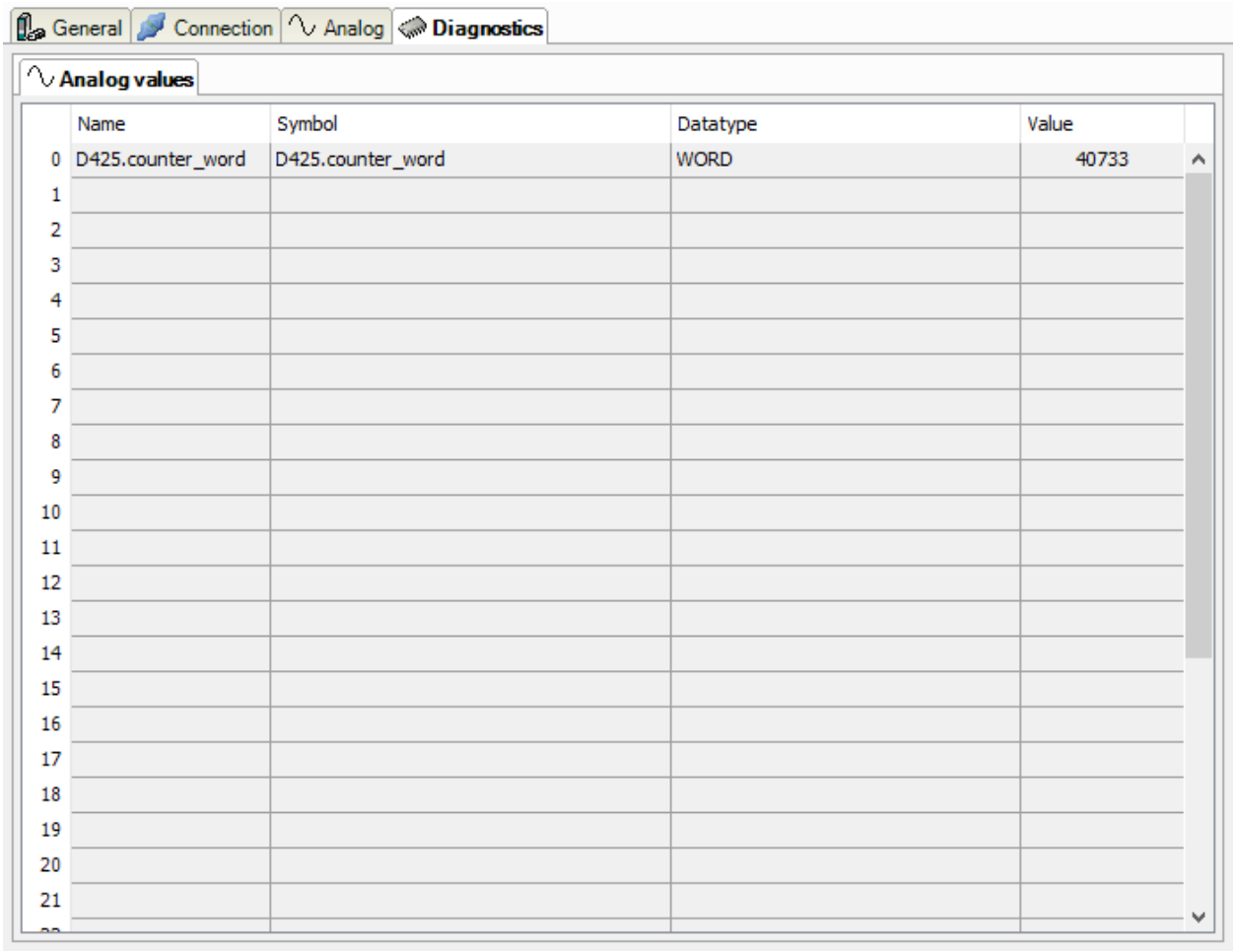


In the Connection tab, it is possible to choose a connection using TCP/IP or PC/CP and configure the IP address of the SIMOTION drive. The address book associated with this module can be selected here as well.

By clicking the Test button, a connection is established to the SIMOTION drive. Basic information such as the MLFB number and current drive state are displayed.



In the Analog tab you can configure which symbols should be read from the SIMOTION drive. By clicking the ... button in the Symbol column, the SIMOTION symbol browser will be opened. You can then select the SIMOTION symbol that should correspond to the selected analog signal. Note that the symbol browser can also be accessed by clicking the link Select symbols at the bottom left of the General tab. There you can add multiple symbols at once.



In the Diagnostics tab, the requested analog signals are displayed along with their current values.

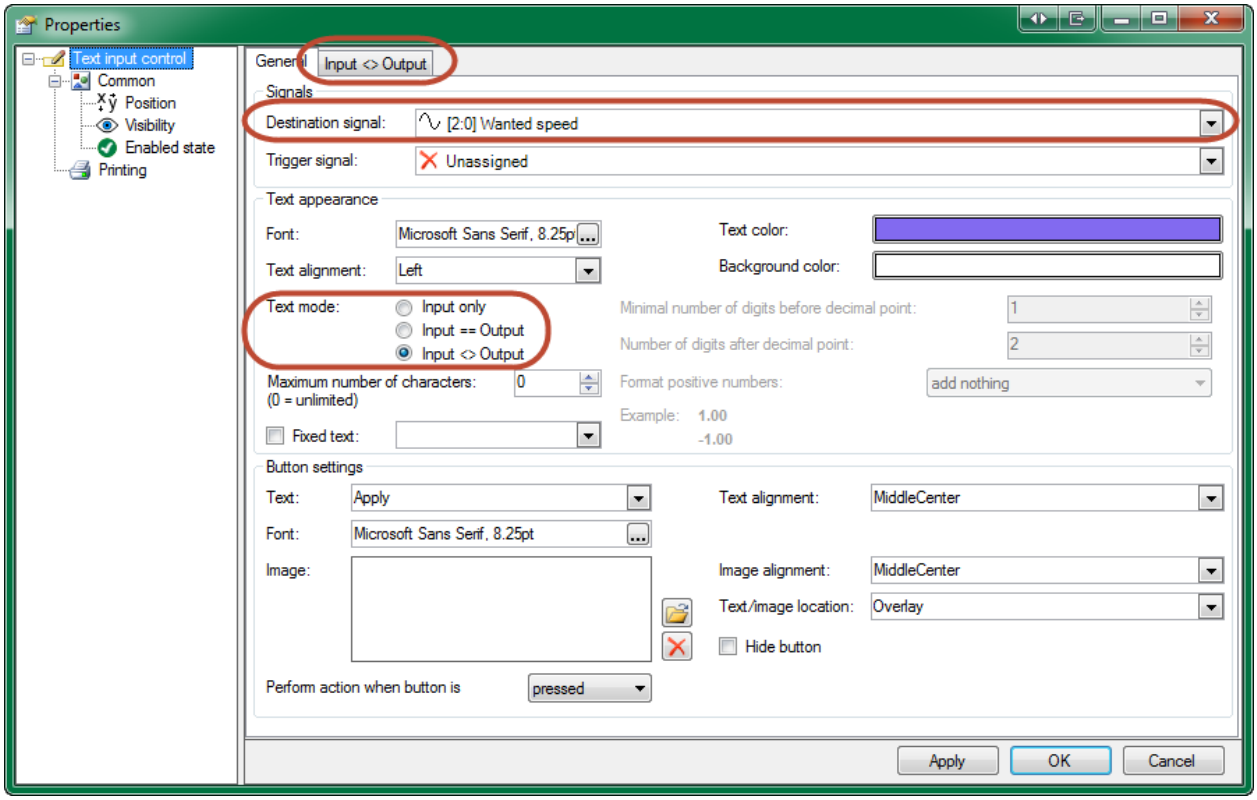
5 QPanel

5.1 Text input control

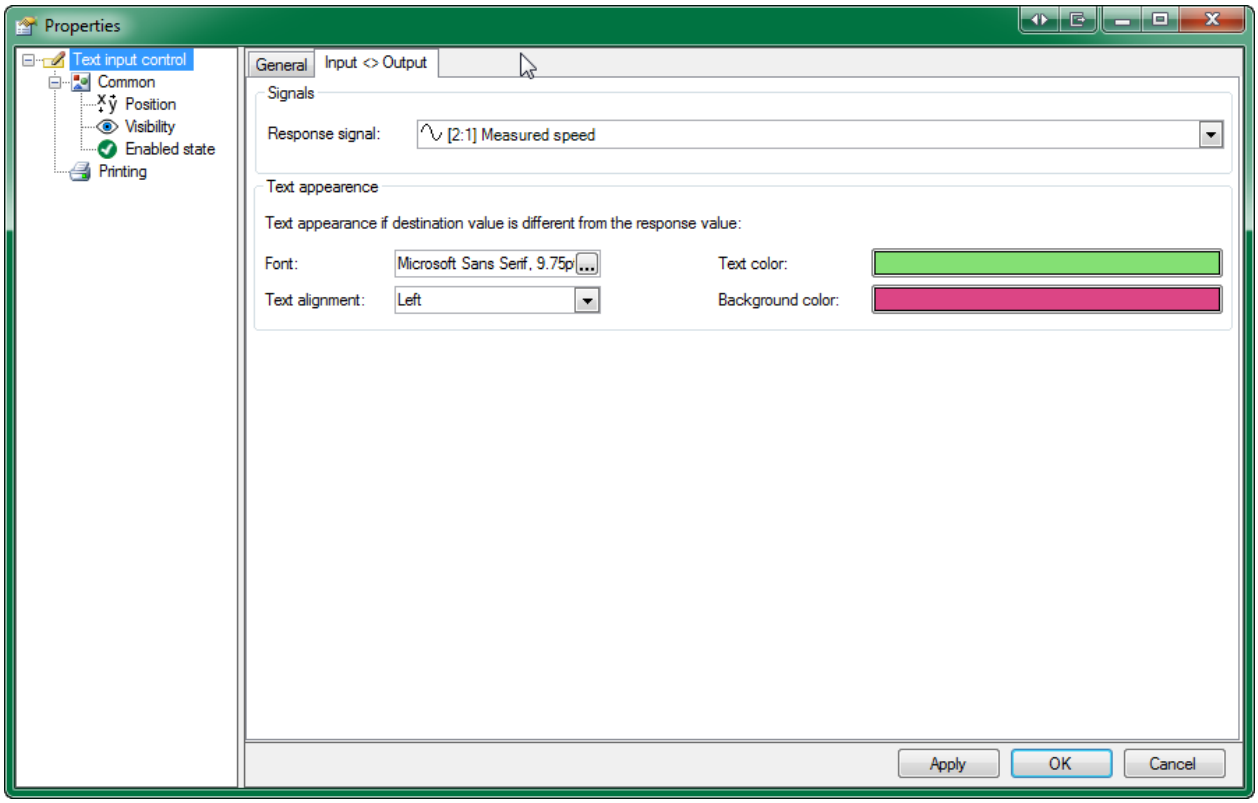
The text input control allows sending manually entered values to a *destination signal* in the ibaPDA server, using a QPanel input module.

When setting the input control to the "*Input == output*" mode, the text field is automatically updated when an other ibaPDA client changes the value.

Starting from version 6.38.0, the text input control was enhanced with a third text mode: "*Input <> Output*". This mode supports a *response signal* that differs from the *destination signal*. The *destination signal* could be a configuration value for setting a device arameter, for instance a speed. The *response signal* is the value that will be shown in the text field after leaving the field; it could be the actual speed of the device.

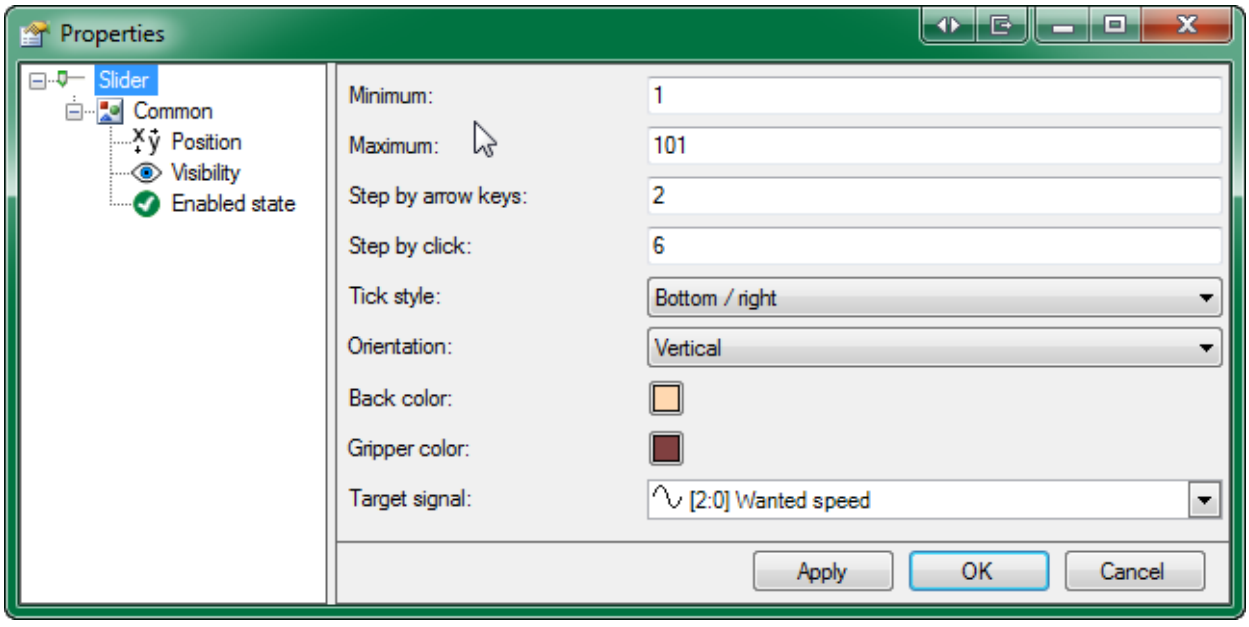


In the settings of the text input control, there is a new TAB to configure the response signal. One can configure some alternative appearance settings which are used when the destination value is different from the response value.

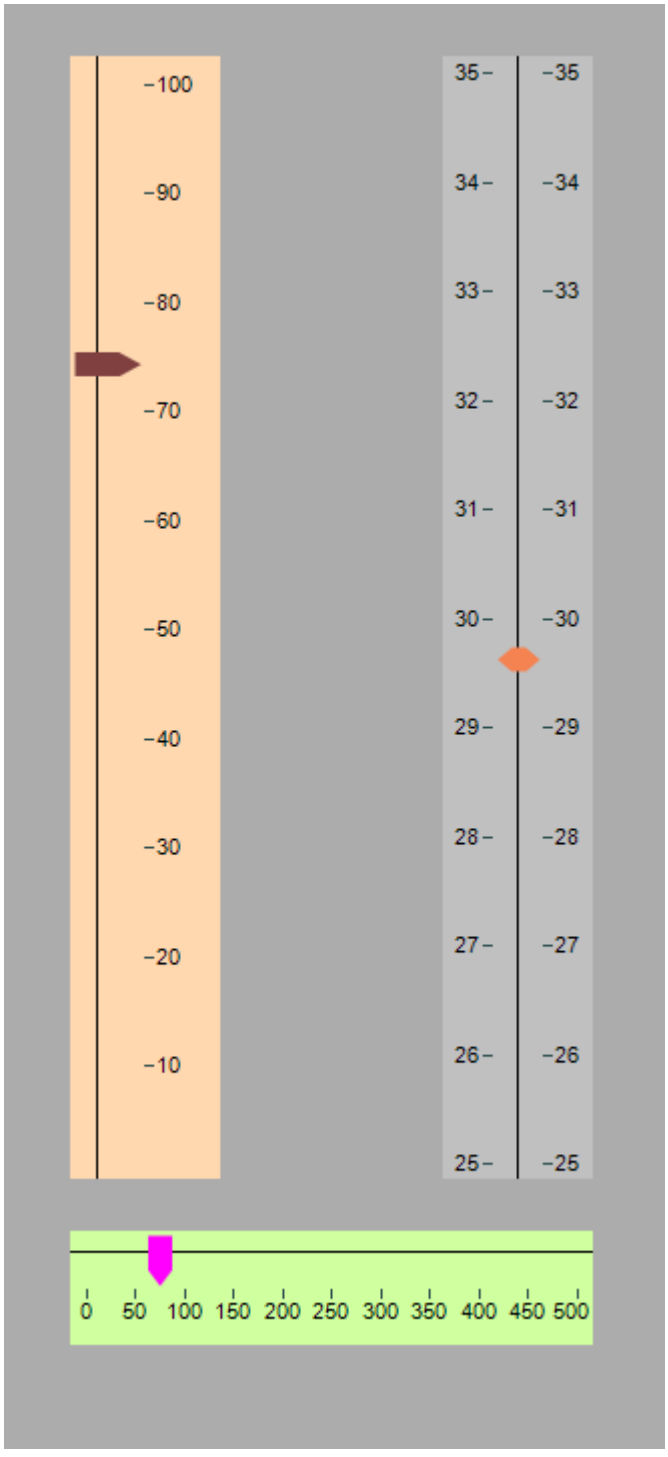


5.2 Slider view

With the new *slider view*, one can set the value of a QPanel input signal with a simple drag operation. The slider view can be fully configured with the settings in the picture below:



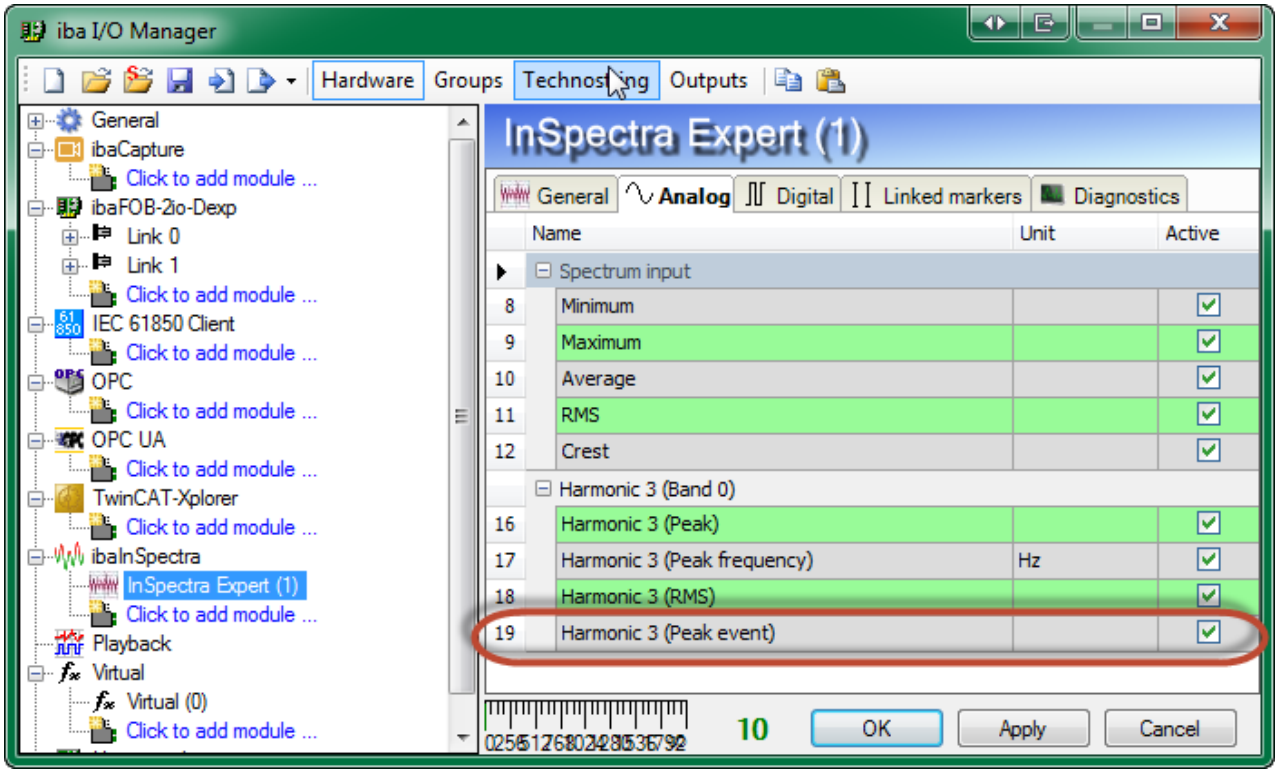
Some example slider views are displayed below. One can use the arrows (up/down or left/right) to change the value of the slider. The sliders can be resized in any direction.



6 Analog alarm/event signals for InSpectra Expert

InSpectra Expert modules use InSpectra profiles to define which frequency bands have to be monitored. For each band, one can set an alert and alarm for the peak and RMS value. For each alert/alarm, exactly one digital signal is available in the digital tab of the InSpectra Expert module.

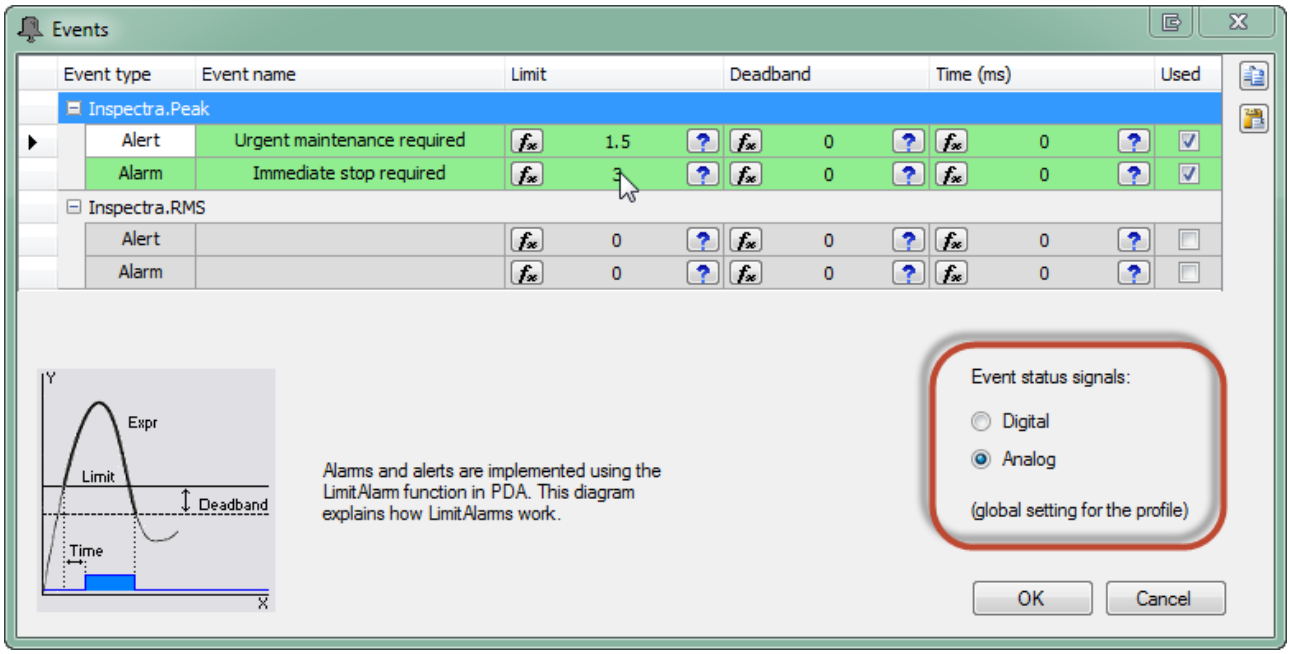
In this version, it is possible to use exactly one analog signal (instead of two digital signals) to indicate the status of the monitored value.



This analog signal can have the following integer values:

- 0: initial value, no information available yet
- 1: everything OK
- 2: alert
- 3: alarm

The choice between digital and analog event signals can be configured in the Events dialog, see picture below. This setting is shared across all bands of the profile. By default, the event signals are digital, like it used to be.

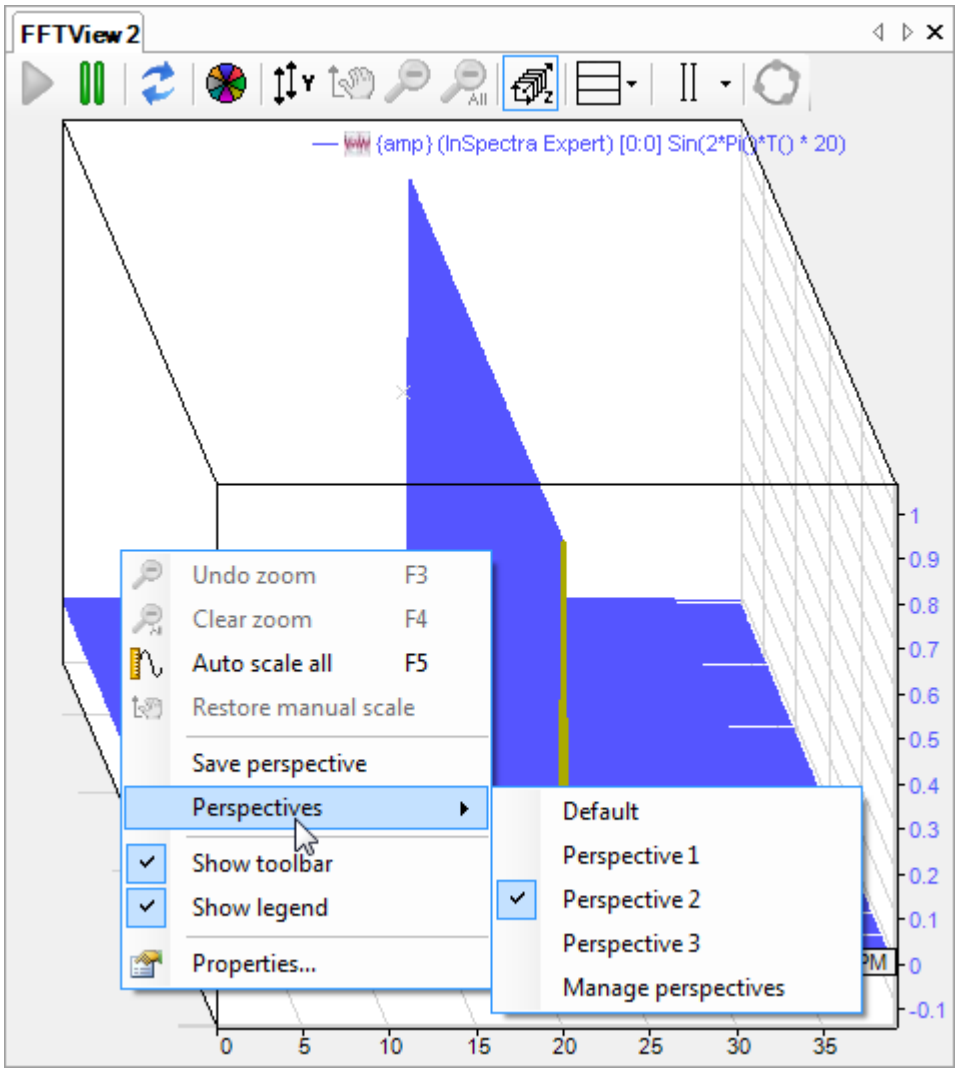


7 FFT view

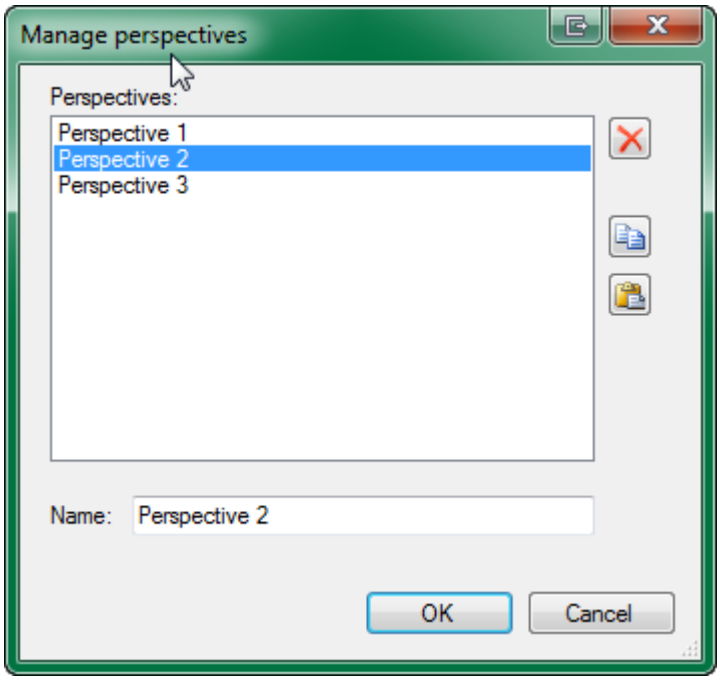
7.1 Waterfall perspectives

In the FFT waterfall view, one can see a history of spectra in an isometric perspective. This perspective can be changed by starting a drag operation while pressing CTRL.

From this version on, it is possible to save the current perspective and give it a name. In this way, one can define a pool of useful perspectives. Using the context menu, one can easily switch between the different perspectives.



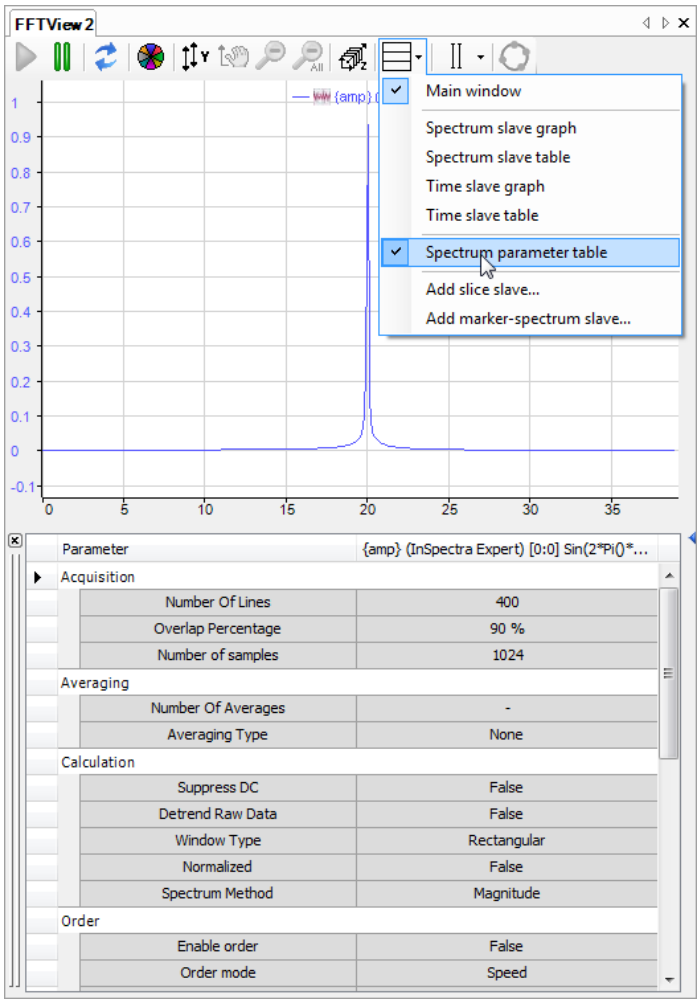
One can manage the perspectives in the dialog below. Perspectives can be renamed and the pool of perspectives can be copied to reuse them in another FFT view.

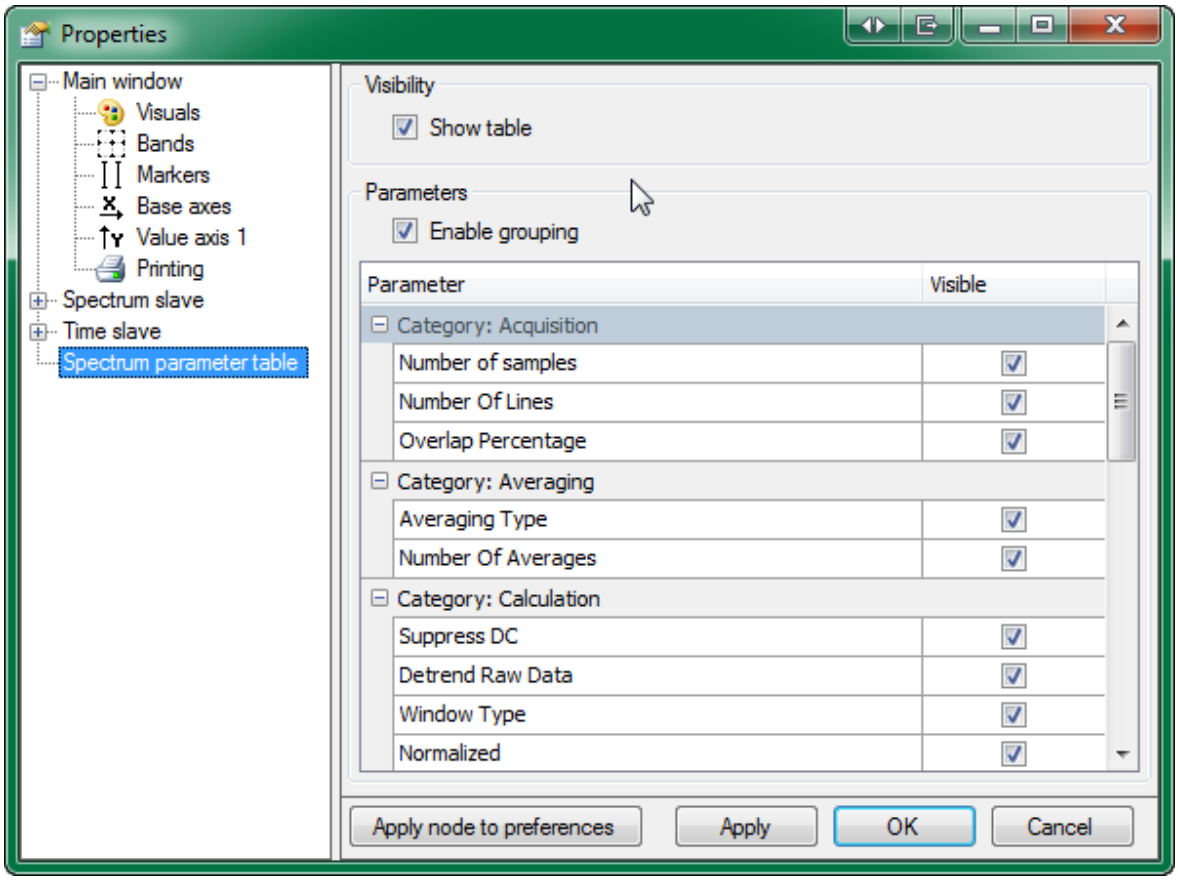


7.2 Live visualization of spectrum parameters

A new slave window was added to the FFT view to display the spectrum parameters used for calculating the spectrum.

Below you find a screenshot of the parameter table and the corresponding configuration dialog. The user can fully customize what parameters are visible.


FFTView 2
Main window
Spectrum slave graph
Spectrum slave table
Time slave graph
Time slave table
Spectrum parameter table
Add slice slave...
Add marker-spectrum slave...
Parameter
{amp} (InSpectra Expert) [0:0] Sin(2*PI()*...
Acquisition
Number Of Lines 400
Overlap Percentage 90 %
Number of samples 1024
Averaging
Number Of Averages -
Averaging Type None
Calculation
Suppress DC False
Detrend Raw Data False
Window Type Rectangular
Normalized False
Spectrum Method Magnitude
Order
Enable order False
Order mode Speed


Properties
Main window
Visuals
Bands
Markers
Base axes
Value axis 1
Printing
Spectrum slave
Time slave
Spectrum parameter table
Visibility
Show table
Parameters
Enable grouping
Parameter
Visible
Category: Acquisition
Number of samples
Number Of Lines
Overlap Percentage
Category: Averaging
Averaging Type
Number Of Averages
Category: Calculation
Suppress DC
Detrend Raw Data
Window Type
Normalized
Apply node to preferences
Apply
OK
Cancel