ibaPDA v7.1.0

New Features

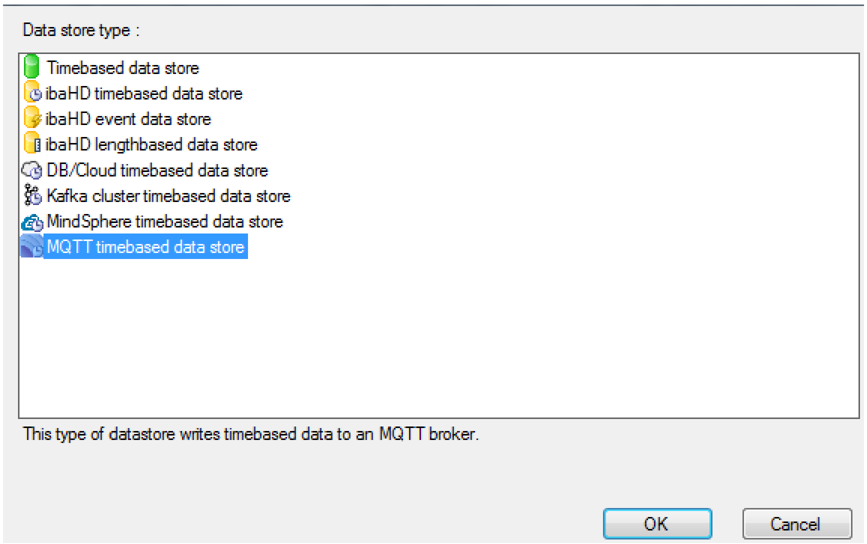
12/11/2019
iba AG

iba AG • Königswarterstr. 44 • D-90762 Fürth • Germany • www.iba-ag.com

Table of contents

1 MQTT timebased data store

The MQTT timebased data store allows you to write timebased signal data to a MQTT broker. The broker distributes those values to registered clients which subscribed to these signals.

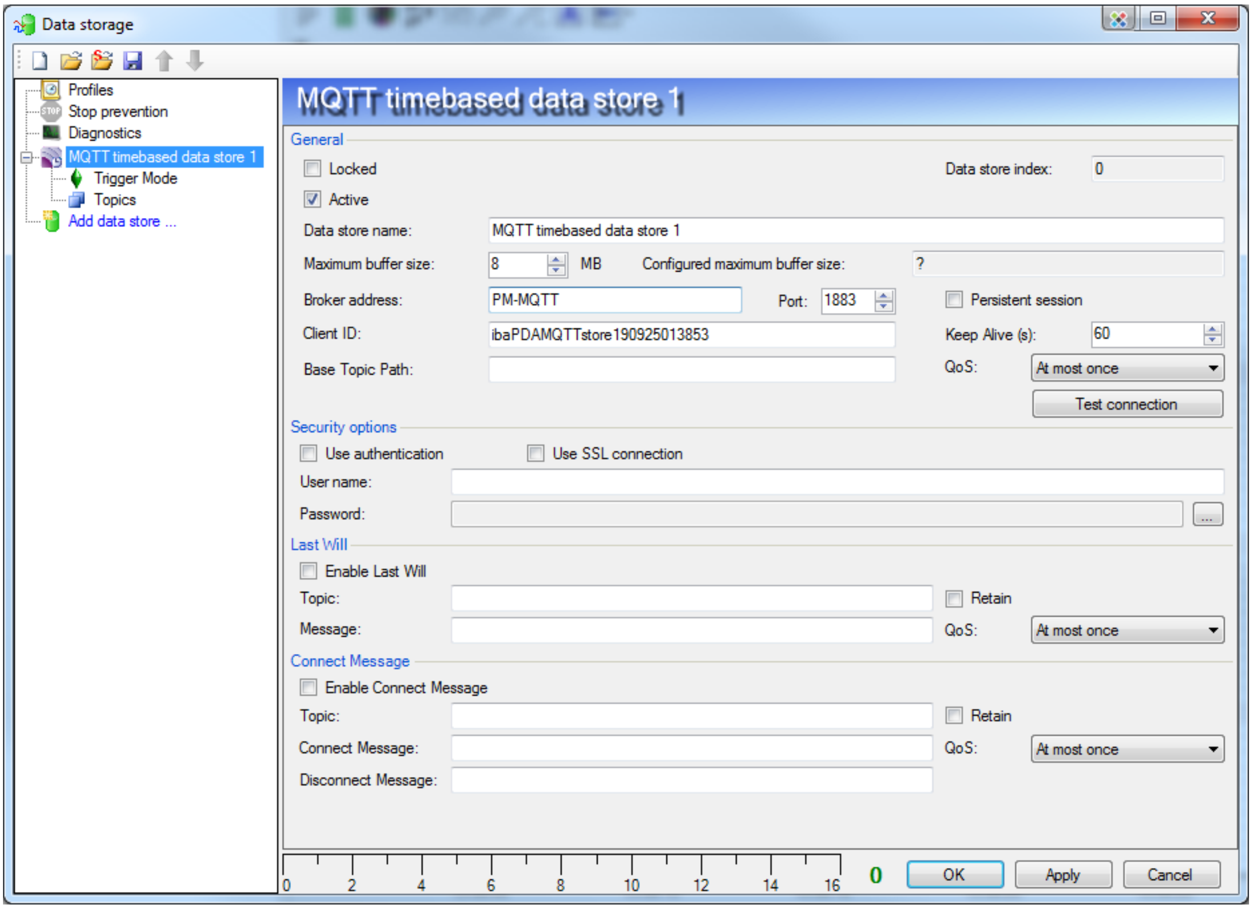


Data store type :

- Timebased data store
- ibaHD timebased data store
- ibaHD event data store
- ibaHD lengthbased data store
- DB/Cloud timebased data store
- Kafka cluster timebased data store
- MindSphere timebased data store
- MQTT timebased data store

This type of datastore writes timebased data to an MQTT broker.

Click the “*Add data store …*” node. The dialog will show the MQTT timebased data store if you have a license for a MQTT data store.



MQTT timebased data store 1

General

□ Locked

Data store index: 0

☑ Active

Data store name: MQTT timebased data store 1

Maximum buffer size: 8 MB

Configured maximum buffer size: ?

Broker address: PM-MQTT

Port: 1883

□ Persistent session

Client ID: ibaPDAMQTTstore190925013853

Keep Alive (s): 60

Base Topic Path:

QoS: At most once

Security options

□ Use authentication □ Use SSL connection

User name:

Password:

Last Will

□ Enable Last Will

Topic:

□ Retain

Message:

QoS: At most once

Connect Message

□ Enable Connect Message

Topic:

□ Retain

Connect Message:

QoS: At most once

Disconnect Message:

In the MQTT data store you first have to configure to which broker you want to connect. You have to enter the broker address and port number. Also each client connected to the broker must have an unique Client ID. ibaPDA will pre-fill that field. Be careful if you change it, duplicate client IDs will not be registered as an error, but will lead to severe problems and signal dropouts.

The parameters relevant to the connection to the MQTT broker have to be configured:

- **Broker address**: The IP address of the broker, can be a resolveable name or in number format.
- **Port**: Port to use for the connection. Normal MQTT port is 1883, or 8883 when using SSL.
- **Client ID**: When connecting to a broker, each client must choose an unique name that is used only once for this broker.
- **Base Topic Path**: <optional> This path is added in front of each registered topic. By this, the topic configuration can be made easier if all topics reside within a common path. While starting the path with “/” is valid, this adds an empty path entry to the topic, which should be avoided, as well as having two separators next to each other (“//”)
- **Persistent session**: If the client is disconnected, the connection interface itself saves the last values not sent to the broker, and delivers them when reconnecting with the same Client ID as before.
- **Keep Alive**: The time for sending a Keep Alive telegram to the broker to make sure the connection still is online.
- **QoS**: The Quality of Service used when registering to a topic on the broker. Possible values are: *At most once* (messages can be dropped), *At least once* (messages are repeated if an acknowledgement is not sent within a certain time), *Exactly once* (a secured handshake protocol for each message sent).

Following security options for the connection are available

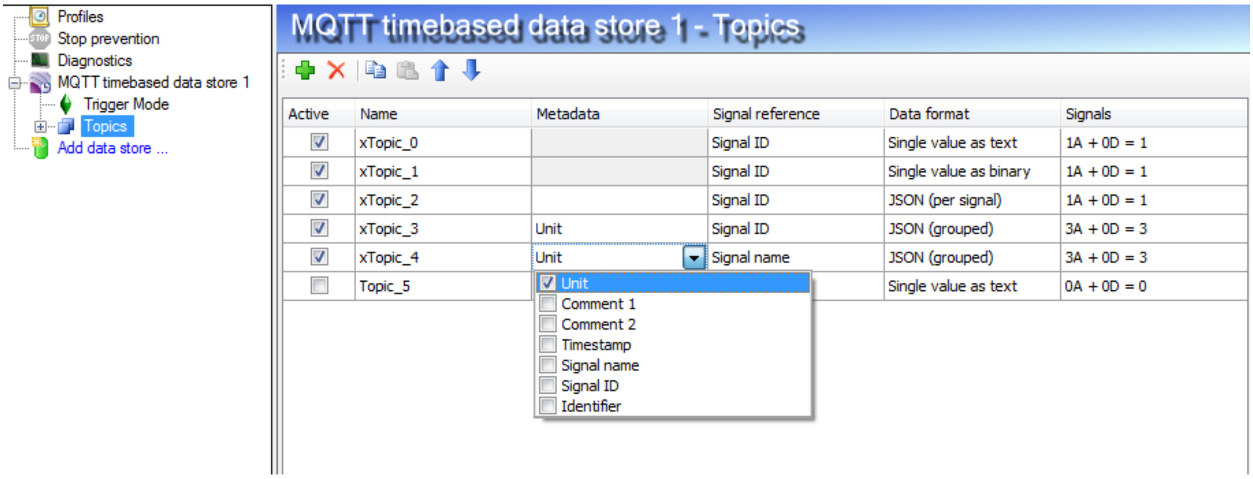
- **Use authentication**: send **user name** and **password** as login when connecting to the broker. This option has to be configured accordingly on the broker, otherwise the connection will fail.
- **Use SSL**: Use SSL instead of plain Tcp connection. Normally for this the port has to be set to 8883. This option has to be configured accordingly on the broker, otherwise the connection will fail.

Last Will is available to announce absence of this client to other clients, if the connection breaks without regular disconnect

- **Topic**: The topic used for the last will message. The base topic path is **not** applied to this topic.
- **Message**: The message sent.
- **Retain**: If this is set, the message in this topic is stored on the broker until it is overwritten.
- **QoS**: The quality of Service used for sending the last will message to other clients.

Connect Message has the same options as the **Last Will**, but these messages are sent on a regular connect resp. disconnect.

When clicking the **Test Connection** button, ibaPDA will try to establish a connection to the MQTT broker, using the selected security settings.



MQTT timebased data store 1 - Topics

Active	Name	Metadata	Signal reference	Data format	Signals
☑	xTopic_0		Signal ID	Single value as text	1A + 0D = 1
☑	xTopic_1		Signal ID	Single value as binary	1A + 0D = 1
☑	xTopic_2		Signal ID	JSON (per signal)	1A + 0D = 1
☑	xTopic_3	Unit	Signal ID	JSON (grouped)	3A + 0D = 3
☑	xTopic_4	Unit	Signal name	JSON (grouped)	3A + 0D = 3
□	Topic_5			Single value as text	0A + 0D = 0

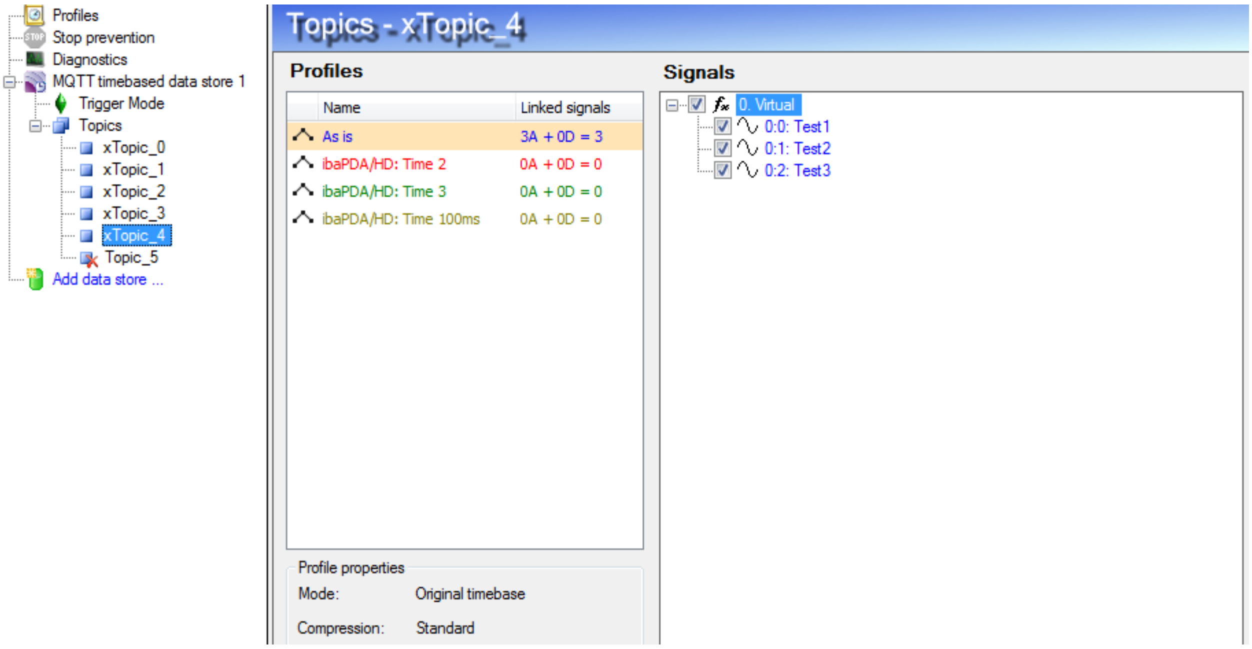
☑ Unit
□ Comment 1
□ Comment 2
□ Timestamp
□ Signal name
□ Signal ID
□ Identifier

You have to define now the topics which should be written to the broker. There are several different formats available, depending on the information needed for the subscribers:

- **Single value as text:** A single signal is written into the broker under that topic, formatting it in plain text mode.
- **Single value as binary:** A single signal is written into the broker under that topic, storing it in binary format.
- **JSON (per signal):** A single signal is written into the broker under that topic as a JSON-formatted line, adding optional metadata. E.g. {"Signal":"[0:2]","Value":"-18.9", “Unit”:”m/s”}
- **JSON (grouped):** Multiple signals are written into the broker under that topic as a JSON-formatted line, adding optional metadata. E.g. {"Test1":"-7.289686","Test1.Unit":"m/s","Test2":"45.88773","Test2.Unit":"mm","Test3":"-18.9","Test3.Unit":"s"}

For the JSON formatted signals, the contained signal reference can be either the signal ID (here: “[0:2]”) or the signal name (here: “Test1”).

In the optional metadata, the timestamp is included because due to the nature of the MQTT transport, any other time information is lost by writing into the broker.



Topics - xTopic_4

Profiles

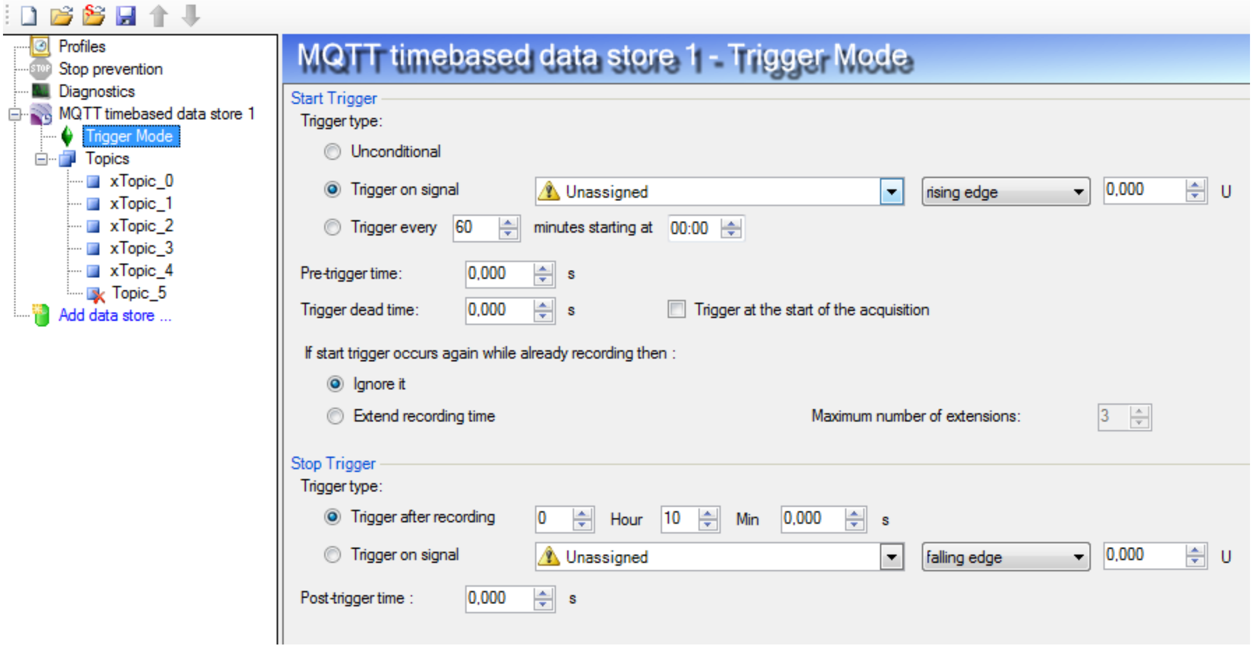
Name	Linked signals
As is	3A + 0D = 3
ibaPDA/HD: Time 2	0A + 0D = 0
ibaPDA/HD: Time 3	0A + 0D = 0
ibaPDA/HD: Time 100ms	0A + 0D = 0

Profile properties
Mode: Original timebase
Compression: Standard

Signals
☑ 0. Virtual
☑ 0:0: Test1
☑ 0:1: Test2
☑ 0:2: Test3

The signals can be selected from the standard time profiles. For the single-value formats, if more than one signal is selected, only the first one is used for writing, all others are simply ignored.

The final thing that needs to be configured in the MQTT data store is the trigger mode. Here you determine when data will be written into the database. It looks similar to the trigger mode of the normal dat file data stores but there are some differences.



MQTT timebased data store 1 - Trigger Mode

Start Trigger
Trigger type:
○ Unconditional
● Trigger on signal Unassigned rising edge 0,000 U
○ Trigger every 60 minutes starting at 00:00
Pre-trigger time: 0,000 s
Trigger dead time: 0,000 s ☐ Trigger at the start of the acquisition
If start trigger occurs again while already recording then :
● Ignore it
○ Extend recording time Maximum number of extensions: 3

Stop Trigger
Trigger type:
● Trigger after recording 0 Hour 10 Min 0,000 s
○ Trigger on signal Unassigned falling edge 0,000 U
Post-trigger time : 0,000 s

First of all you can select whether you want to write constantly to the database or only triggered. Select *unconditional* as start trigger type when you just want to write all data to the database. In this case all other settings are irrelevant.

When you want to write triggered data then select the start trigger. You can select between:

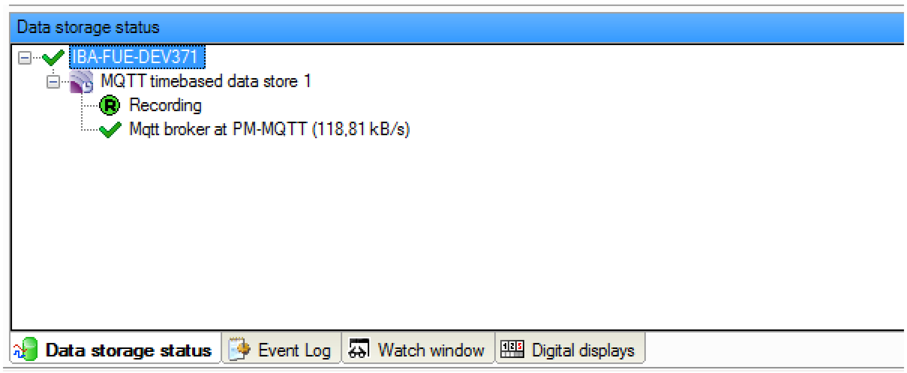
- Trigger on an edge of a signal.
- Trigger periodically on a specific timestamp.

You can configure a pre-trigger time to record some extra time before the trigger event occurred. The trigger dead time determines how much time needs to pass before a new start trigger event is considered. You can also configure what should happen when a new start trigger occurs while a recording is already running. You can either ignore this new start trigger or extend the recording time.

There are 2 options for the stop trigger:

- Trigger after recording a configureable amount of the time after the start trigger occurred.
- Trigger on an edge of a signal.

Finally there is also a post-trigger time to record some additional time after the stop trigger occurred.



Data storage status

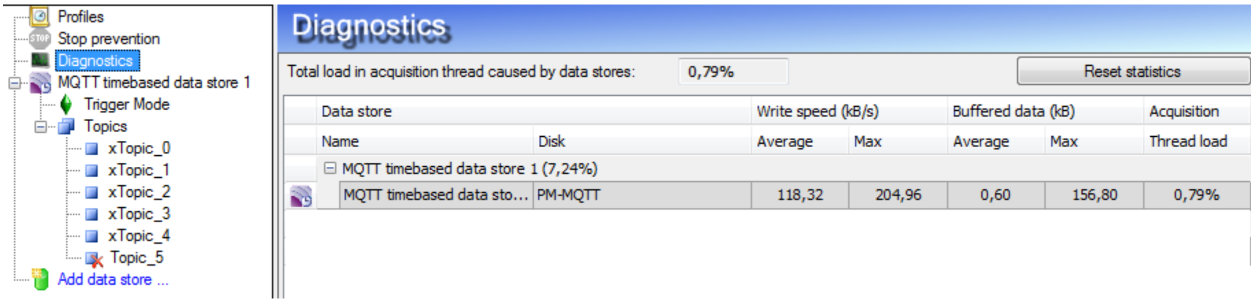
- IBA-FUE-DEV371
  - MQTT timebased data store 1
    - Ⓡ Recording
    - ✓ Mqtt broker at PM-MQTT (118,81 kB/s)

The current status of the DB/Cloud data store can be seen in the *data storage status* window. There are 2 nodes shown per data store. The first one shows the name of the database table and the recording time in brackets. The icon depicts the current state of the recording. There are 3 possible icons:

- Ⓣ Waiting for start trigger
- Ⓡ Recording
- Ⓟ Recording post trigger

By right clicking on this node you can also send a manual start or stop trigger.

The second node shows the status of the connection to the database. If the connection is ok then the write speed is shown. If the connection is not ok then the last error is shown.



Total load in acquisition thread caused by data stores: 0,79%

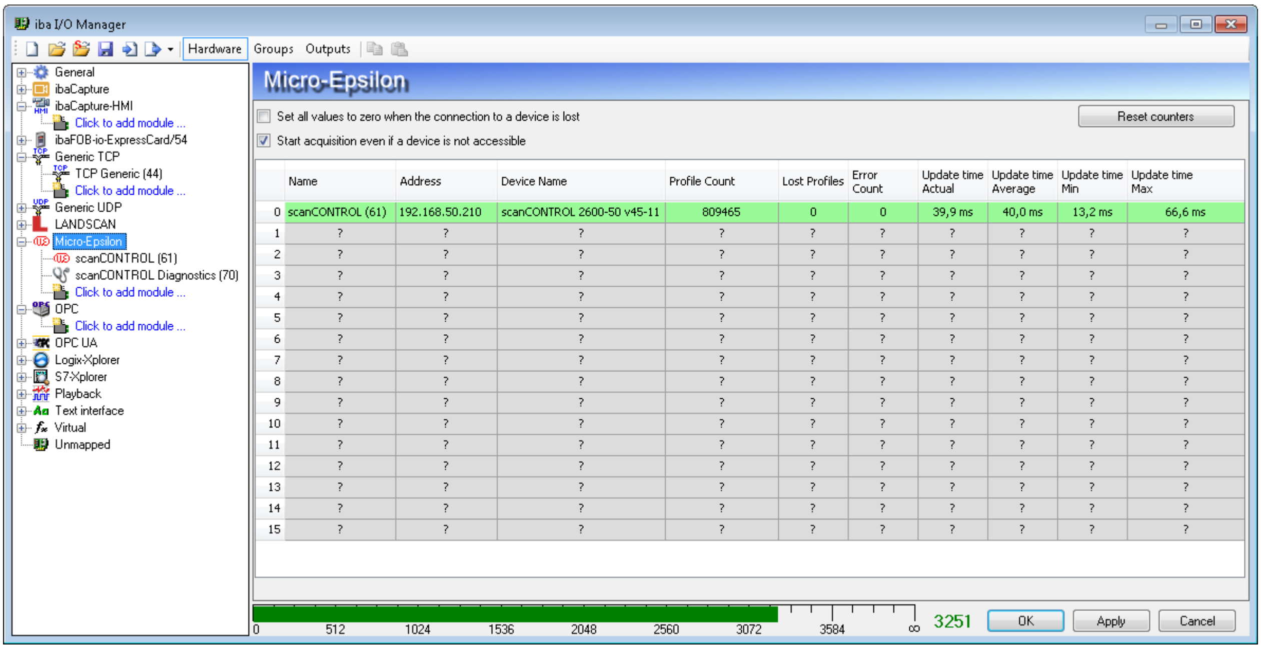
Data store		Write speed (kB/s)		Buffered data (kB)		Acquisition
Name	Disk	Average	Max	Average	Max	Thread load
MQTT timebased data store 1 (7,24%)						
MQTT timebased data sto...	PM-MQTT	118,32	204,96	0,60	156,80	0,79%

The diagnostics grid in the data storage manager has been extended with the MQTT data store as well. The group shows the load of the writer thread that writes the data to the broker. The disk column shows the address of the database. The write speed shows how fast data is written to the database. The buffered data shows how much data is buffered in ibaPDA. The buffered data will grow when there is no connection to the database. If the size goes over the configured buffer size then the oldest data will be discarded from the buffer. The acquisition thread load column shows the load generated by evaluating the triggers and creating the row data.

2 Micro-Epsilon interface for scanCONTROL devices

The Micro-Epsilon interface in ibaPDA is used to acquire data from a scanCONTROL laser profile scanner from Micro-Epsilon.

While it is possible to use a Micro-Epsilon scanCONTROL scanner connected to a network switch, it is highly recommended to use a 1:1 connection from the measurement PC to the device, using a gigabit connection. While the device is running, it can generate a high network load, depending on the configuration.



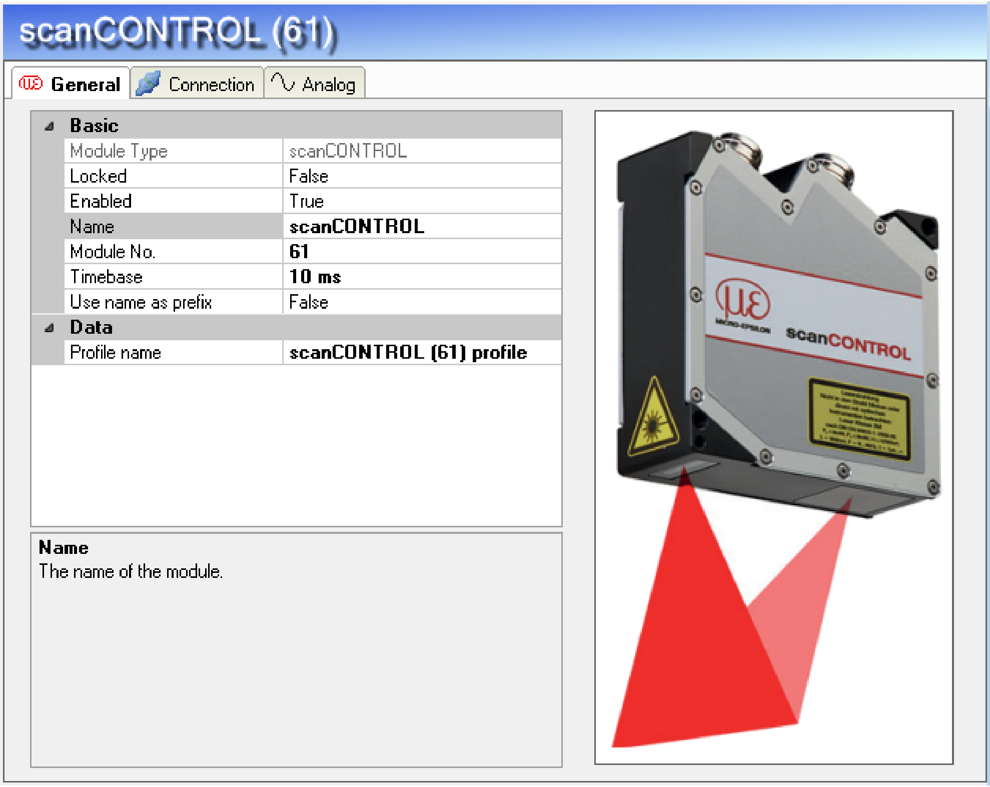
☐ Set all values to zero when the connection to a device is lost

☑ Start acquisition even if a device is not accessible

	Name	Address	Device Name	Profile Count	Lost Profiles	Error Count	Update time Actual	Update time Average	Update time Min	Update time Max
0	scanCONTROL (61)	192.168.50.210	scanCONTROL 2600-50 v45-11	809465	0	0	39,9 ms	40,0 ms	13,2 ms	66,6 ms
1	?	?	?	?	?	?	?	?	?	?
2	?	?	?	?	?	?	?	?	?	?
3	?	?	?	?	?	?	?	?	?	?
4	?	?	?	?	?	?	?	?	?	?
5	?	?	?	?	?	?	?	?	?	?
6	?	?	?	?	?	?	?	?	?	?
7	?	?	?	?	?	?	?	?	?	?
8	?	?	?	?	?	?	?	?	?	?
9	?	?	?	?	?	?	?	?	?	?
10	?	?	?	?	?	?	?	?	?	?
11	?	?	?	?	?	?	?	?	?	?
12	?	?	?	?	?	?	?	?	?	?
13	?	?	?	?	?	?	?	?	?	?
14	?	?	?	?	?	?	?	?	?	?
15	?	?	?	?	?	?	?	?	?	?

The Micro-Epsilon interface shows a table of the configured connections. Each Micro-Epsilon license allows access to 2 scanCONTROL devices. Each connection corresponds to a row in the table. The row is green when the connection is ok and data is being received. The row is red when the connection could not be established. The row is grey when there is no connection configured. The Profile Count increases with each update sent from the Device, which contains an internal profile number. The Lost Profiles reflects lost updates according to the internal profile number. The error count updates with communication errors that result in a connection loss.The profile update timing is reflected in the update time fields

On the interface node you can also configure if the acquisition is started even if the connection to a configured scanCONTROL device could not be established, and if the values should be reset to zero if a connection loss is detected. If the acquisition is started without establishing a connection ibaPDA tries to reestablish itcyclically during the running acquisition.

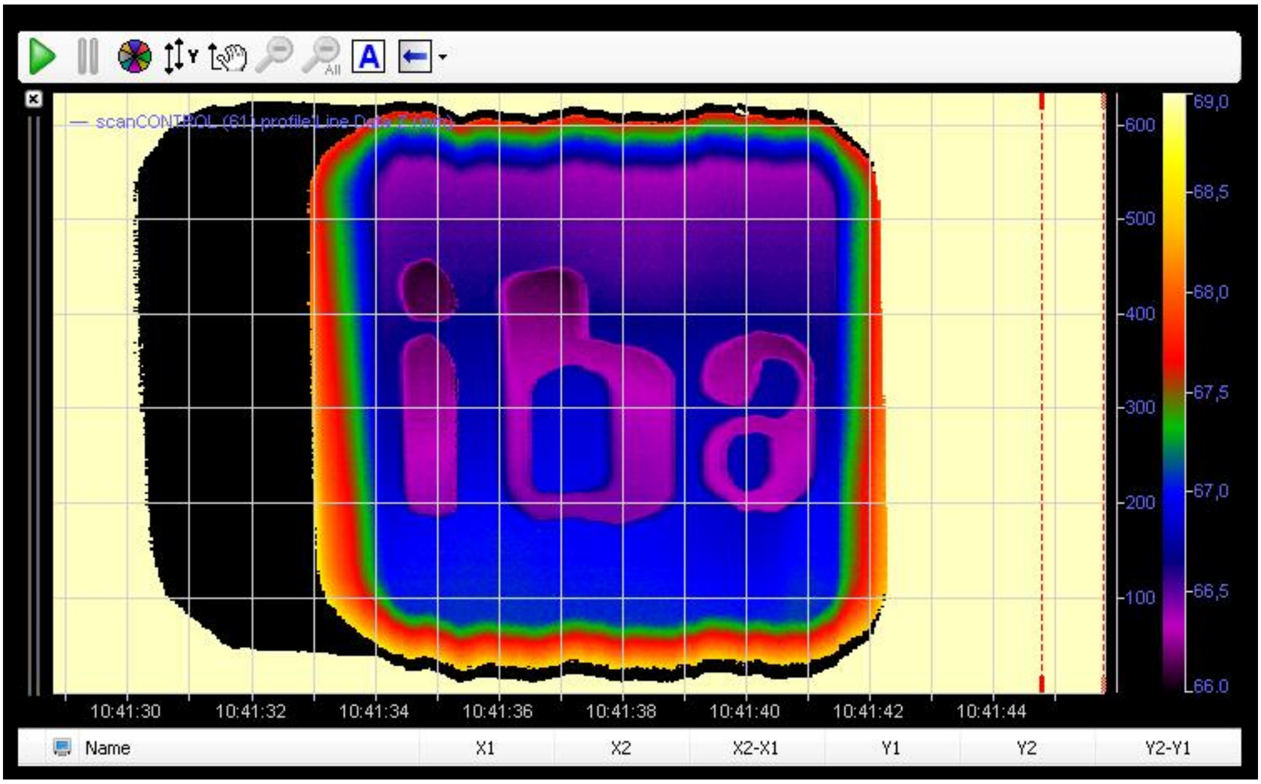


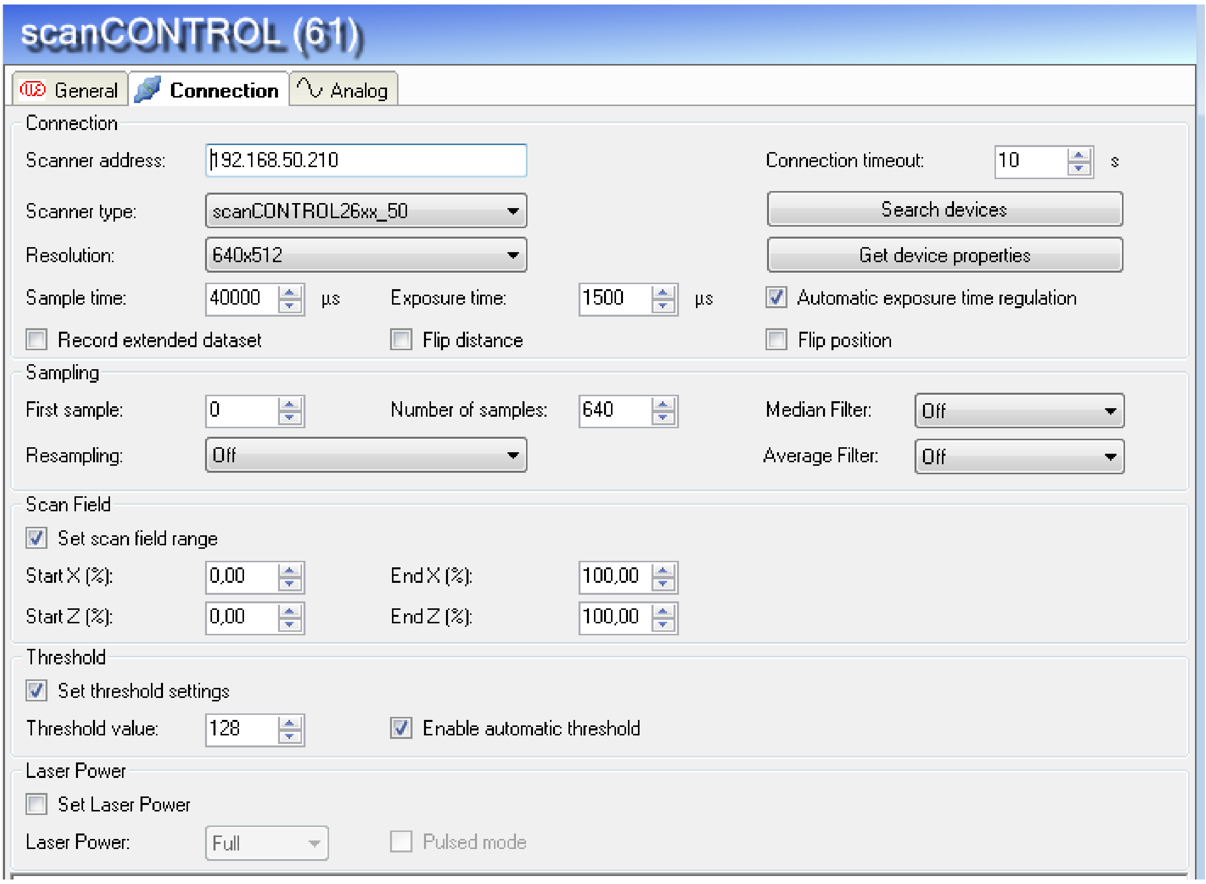
Basic	
Module Type	scanCONTROL
Locked	False
Enabled	True
Name	**scanCONTROL**
Module No.	**61**
Timebase	**10 ms**
Use name as prefix	False
Data	
Profile name	**scanCONTROL (61) profile**

Name
The name of the module.

In the General Tab of a scanCONTROL module, apart from the standard options, the profile name can be configured, which is used as name for the automatically generated signal vectors.

Vectors can be visualized very easily as a 2D top view in a trend graph in ibaQPanel (requires an additional license).





scanCONTROL (61)

Connection

Scanner address: 192.168.50.210

Connection timeout: 10 s

Scanner type: scanCONTROL26xx_50

Resolution: 640x512

Sample time: 40000 µs

Exposure time: 1500 µs

☑ Automatic exposure time regulation

□ Record extended dataset

□ Flip distance

□ Flip position

Sampling

First sample: 0

Number of samples: 640

Median Filter: Off

Resampling: Off

Average Filter: Off

Scan Field

☑ Set scan field range

Start X (%): 0,00

End X (%): 100,00

Start Z (%): 0,00

End Z (%): 100,00

Threshold

☑ Set threshold settings

Threshold value: 128

☑ Enable automatic threshold

Laser Power

□ Set Laser Power

Laser Power: Full

□ Pulsed mode

In the **Connection** tab, all parameters relevant to the connection to, and data acquisition from the scanCONTROL device have to be configured. It is split into several sections which can be independently activated or deactivated. These optional parameters can also be configured by using the Micro-Epsilon scanCONTROL configuration tool, and if they are not actively selected in ibaPDA, the pre-configured options are used. When clicking on the “Search devices” button, a scan for attached devices is performed, and on selecting a found device, its current properties are filled into the appropriate configuration fields. When clicking on the “Get device properties” buttons, the properties of the currently configured device are read from the device, and the appropriate configuration fields are updated.

Section “Connection”:

- **Scanner address**: The IP address of the scanner.
- **Scanner type**: The type of the connected scanner.
- **Resolution**: The used resolution of the detector. The list of available resolutions changes with the selected scanner type.
- **Connection timeout**: The maximum time between two profile updates before the connection is considered to be broken.
- **Sample time**: The time between two profile acquisitions by the scanner.
- **Exposure time**: The time for the shutter to open. Depends on laser power and scanned material. Please refer to device manual for further details. This must be smaller than the selected sample time.

- **Automatic exposure time regulation**: After starting with the configured exposure time, the scanner automatically adjusts this time for optimal results.
- **Record extended dataset**: For normal operation, acquisition of the X and Z vectors is sufficient. For diagnostic reasons, additional vectors containing the used threshold, reflection width and reflection maximum can be recorded, too.
- **Flip distance**: The Z vector contains the distance between the scanned material and the scanner. These values are normally reverse to the material height. To use the values it might be easier to flip the distance value.
- **Flip position**: Due to the scanner position, it might be that the value alignment from left to right from scanner view does match the material postion. This can be fixed with this option. Caution! Without resampling, this only affects the values in the X vector. The order of the values remains unchanged.

Section "Sampling":

These values always have to be configured in ibaPDA, as they are defined per acquisition session for the device, and affect the vector size available for further processing.

- **First sample**: The first sample in the data transferred to ibaPDA. Also changes the first sample number in the resulting vectors
- **Number of samples**: The number of samples in the data transferred to ibaPDA. This changes the number of values in the resulting vectors. Caution! The sum of first sample and the number of samples must be smaller than the selected resolution.
- **Resampling**: Resample vector to equidistant values. Important for further processing in ibaAnalyzer, as it supports only equidistant values. Please refer to device manual for further details.
- **Median Filter / Average Filter**: Apply filter to Z values, especially important for resampling. Please refer to device manual for further details.

Section "Scan field":

The scan field defines the region of the device sensor that is used for data acquisition. This is important for acquiring data with short sample times. The scan field can be set independent of the "First sample" / "Number of samples" values, but values outside the scan field are set to 0. Please refer to device manual for further details.

- **Set scan field range**: Use scan field settings as defined in ibaPDA
- **Start X / End X / Start Z / End Z**: Define region of the device sensor that is used for data acquisition, in percent of the full sensor.

Section "Threshold":

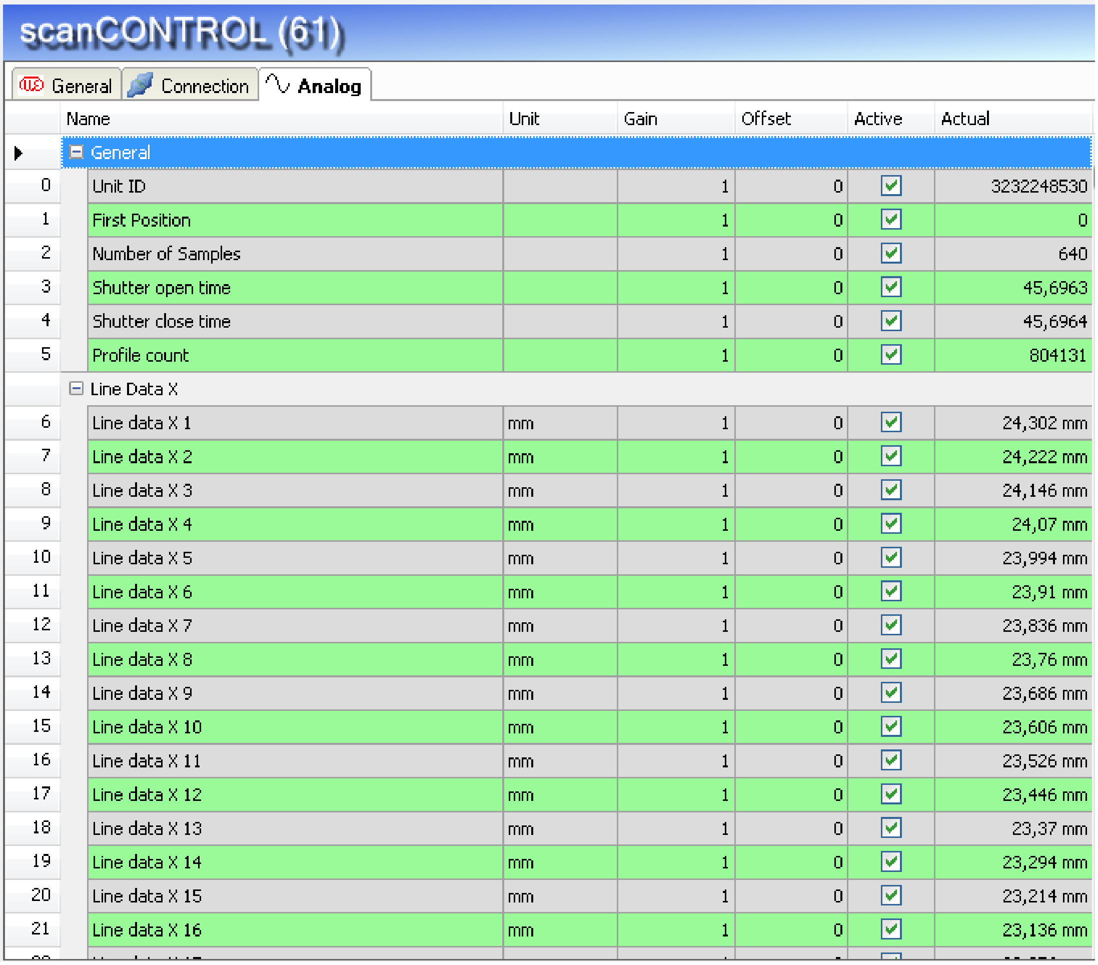
The threshold is used to remove noise resulting from external light reflections, or high scattering due to surface textures. Please refer to device manual for further details.

- **Set threshold settings**: Use threshold settings as defined in ibaPDA
- **Threshold value**: The threshold for considering a measured response to be valid
- **Enable automatic threshold**: The threshold is calculated by the sensor to adjust better to different materials. The current threshold value can be checked by recording the extended dataset.

Section “Laser Power”:

Controls the laser power used for measurement. Please refer to device manual for further details.

- **Set Laser Power**: Use the laser power settings as defined in ibaPDA
- **Laser Power**: Can be set to Full/Reduced/Off
- **Pulsed mode**: Pulse laser according to sample time

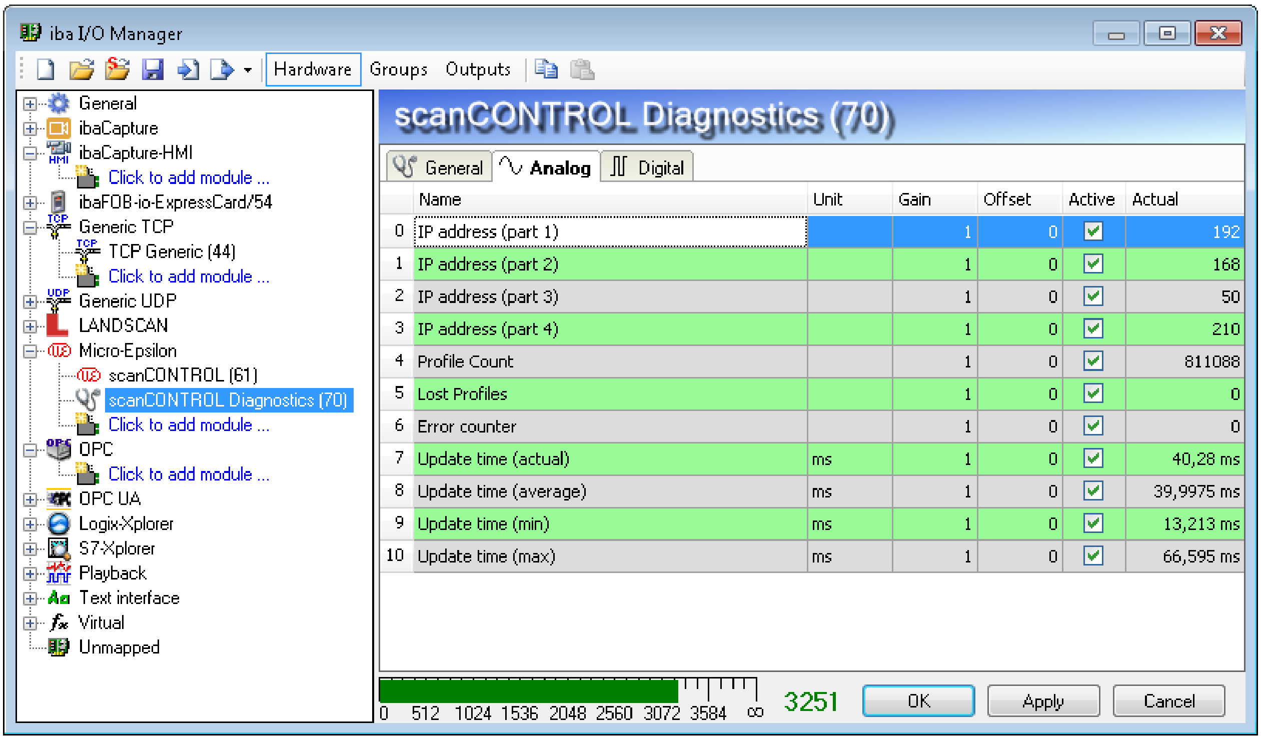


scanCONTROL (61)

	Name	Unit	Gain	Offset	Active	Actual
	⊟ General					
0	Unit ID		1	0	☑	3232248530
1	First Position		1	0	☑	0
2	Number of Samples		1	0	☑	640
3	Shutter open time		1	0	☑	45,6963
4	Shutter close time		1	0	☑	45,6964
5	Profile count		1	0	☑	804131
	⊟ Line Data X					
6	Line data X 1	mm	1	0	☑	24,302 mm
7	Line data X 2	mm	1	0	☑	24,222 mm
8	Line data X 3	mm	1	0	☑	24,146 mm
9	Line data X 4	mm	1	0	☑	24,07 mm
10	Line data X 5	mm	1	0	☑	23,994 mm
11	Line data X 6	mm	1	0	☑	23,91 mm
12	Line data X 7	mm	1	0	☑	23,836 mm
13	Line data X 8	mm	1	0	☑	23,76 mm
14	Line data X 9	mm	1	0	☑	23,686 mm
15	Line data X 10	mm	1	0	☑	23,606 mm
16	Line data X 11	mm	1	0	☑	23,526 mm
17	Line data X 12	mm	1	0	☑	23,446 mm
18	Line data X 13	mm	1	0	☑	23,37 mm
19	Line data X 14	mm	1	0	☑	23,294 mm
20	Line data X 15	mm	1	0	☑	23,214 mm
21	Line data X 16	mm	1	0	☑	23,136 mm

The **Analog** tab of a scanCONTROL module shows the available values. When the acquisition is running, the “Actual” column shows the current value of the available data.

Remark: The number of line data signals is always smaller than configured as in ‘Number of samples’. By definition in the scanCONTROL interface the last data fields are always overwritten with time data information which is excluded from recording in ibaPDA. Please refer to device manual for further details.

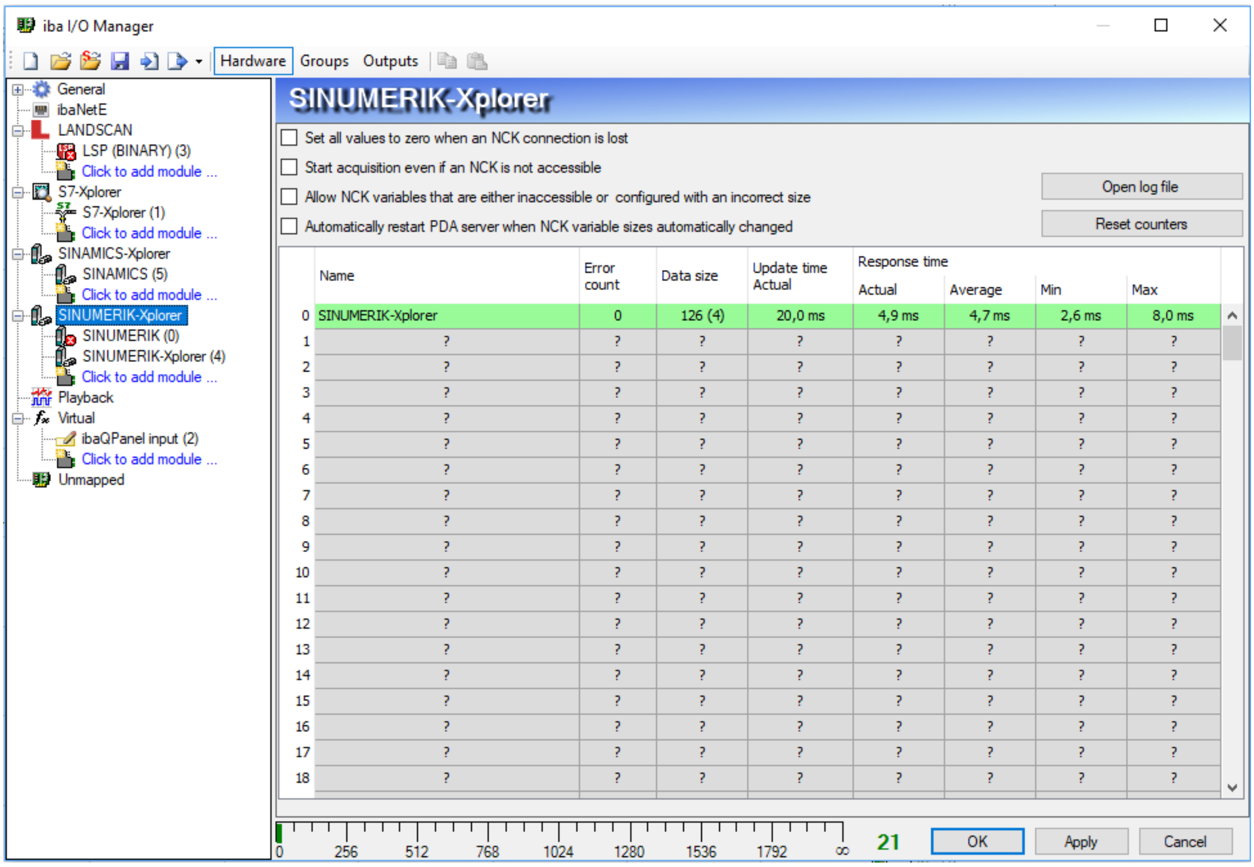


	Name	Unit	Gain	Offset	Active	Actual
0	IP address (part 1)		1	0	☑	192
1	IP address (part 2)		1	0	☑	168
2	IP address (part 3)		1	0	☑	50
3	IP address (part 4)		1	0	☑	210
4	Profile Count		1	0	☑	811088
5	Lost Profiles		1	0	☑	0
6	Error counter		1	0	☑	0
7	Update time (actual)	ms	1	0	☑	40,28 ms
8	Update time (average)	ms	1	0	☑	39,9975 ms
9	Update time (min)	ms	1	0	☑	13,213 ms
10	Update time (max)	ms	1	0	☑	66,595 ms

For the Micro-Epsilon devices, a diagnostic module can be configured.It can record the values that are available on the interface table.

3 SINUMERIK-Xplorer

The SINUMERIK-Xplorer interface can be used to measure data from the NCK part of Siemens Sinumerik NCUs. To access the PLC you can use S7-Xplorer. For each SINUMERIK-Xplorer license you can connect up to 16 Sinumerik NCUs. You can purchase up to 16 licenses, resulting in a maximum of 256 SINUMERIK-Xplorer connections.



☐ Set all values to zero when an NCK connection is lost

☐ Start acquisition even if an NCK is not accessible

☐ Allow NCK variables that are either inaccessible or configured with an incorrect size

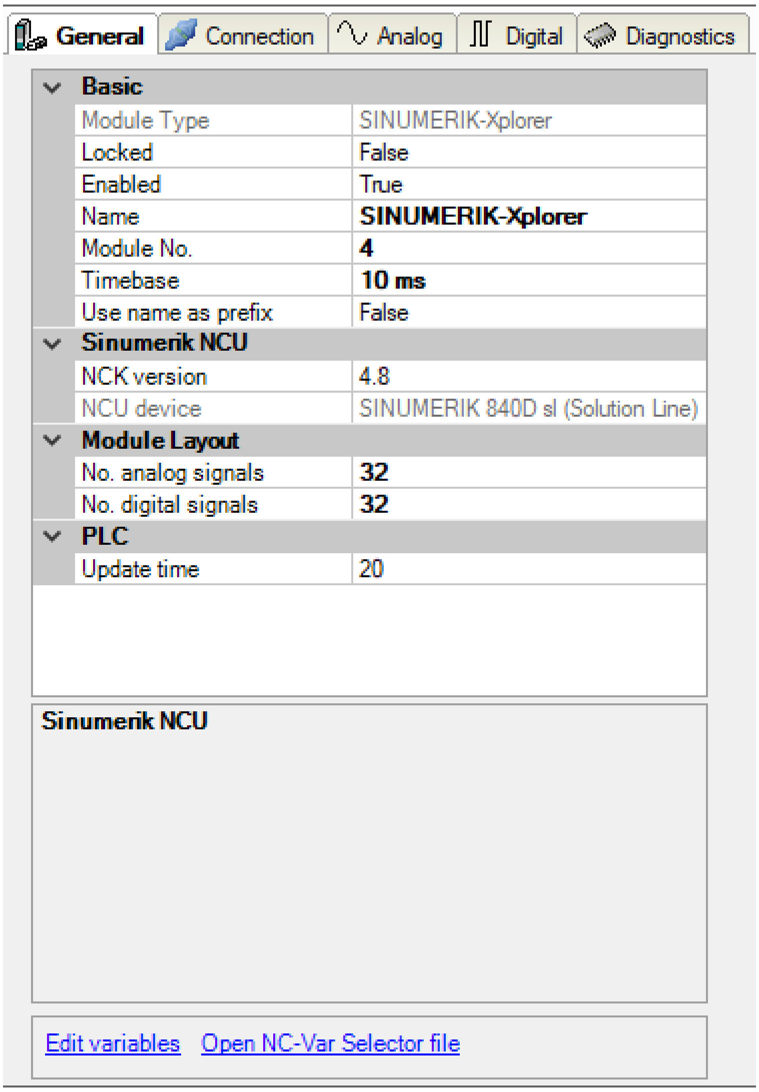
☐ Automatically restart PDA server when NCK variable sizes automatically changed

	Name	Error count	Data size	Update time Actual	Response time			
					Actual	Average	Min	Max
0	SINUMERIK-Xplorer	0	126 (4)	20,0 ms	4,9 ms	4,7 ms	2,6 ms	8,0 ms
1	?	?	?	?	?	?	?	?
2	?	?	?	?	?	?	?	?
3	?	?	?	?	?	?	?	?
4	?	?	?	?	?	?	?	?
5	?	?	?	?	?	?	?	?
6	?	?	?	?	?	?	?	?
7	?	?	?	?	?	?	?	?
8	?	?	?	?	?	?	?	?
9	?	?	?	?	?	?	?	?
10	?	?	?	?	?	?	?	?
11	?	?	?	?	?	?	?	?
12	?	?	?	?	?	?	?	?
13	?	?	?	?	?	?	?	?
14	?	?	?	?	?	?	?	?
15	?	?	?	?	?	?	?	?
16	?	?	?	?	?	?	?	?
17	?	?	?	?	?	?	?	?
18	?	?	?	?	?	?	?	?

The SINUMERIK-Xplorer interface node contains a table listing all configured and enabled SINUMERIK-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).

In case of a stable connection to the SINUMERIK NCK, 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.

3.1 General



Basic	
Module Type	SINUMERIK-Xplorer
Locked	False
Enabled	True
Name	**SINUMERIK-Xplorer**
Module No.	**4**
Timebase	**10 ms**
Use name as prefix	False
Sinumerik NCU	
NCK version	4.8
NCU device	SINUMERIK 840D sl (Solution Line)
Module Layout	
No. analog signals	**32**
No. digital signals	**32**
PLC	
Update time	20

Sinumerik NCU

In the General tab of a SINUMERIK-Xplorer module you can configure a couple of settings, apart from the general module settings. The NCK version can be chosen. That version defines which variables can be selected from a tree of documented NCK variables in the Lists Manuals available for that specific firmware version. The NCU device is automatically set when the connection is tested. It is read only and useful for support cases only.

There are multiple ways to enter the details about which variables must be fetched. The information can be entered directly in the analog or digital signal grids, the Sinumerik symbol browser can be used (see below) or a var file can be opened.

All the variables contained in a .var file can be added by clicking on the link “Open NC-Var Selector file”. Var files can be created with the NC-VAR-Selektor, a program that is bundled with the Siemens Sinumerik Toolbox for both Step 7 and TIA Portal.

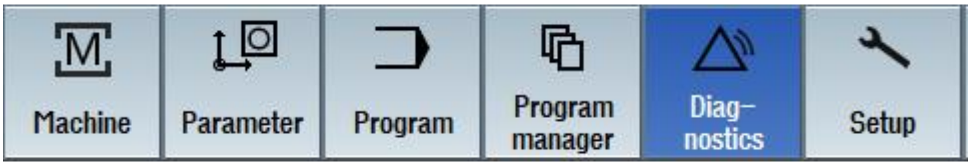
The “Edit variables” link opens the Sinumerik symbol browser (see below).

3.1.1 NCK version

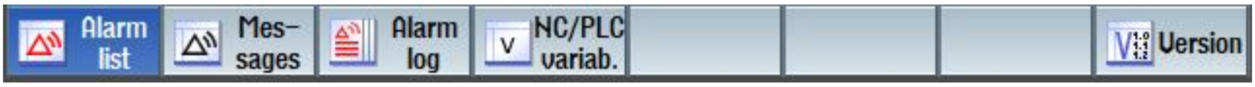
The NCK version is used in the symbol browser, to show only the symbols that are documented to work with the selected firmware. Entering the wrong firmware version has no further impact.

The NCK version is the version of the firmware that is installed on the flash card inserted in the Sinumerik controller. If the Sinumerik controller is switched off, you can pull out the flash card and read the firmware version, which is printed on its label. However, when using SINUMERIK-Xplorer, in most cases the controller will be powered. The NCK version can also be found by using the HMI.

In the HMI, open the Diagnostics tab by pressing the “ALARM” or “Diagnostics” key

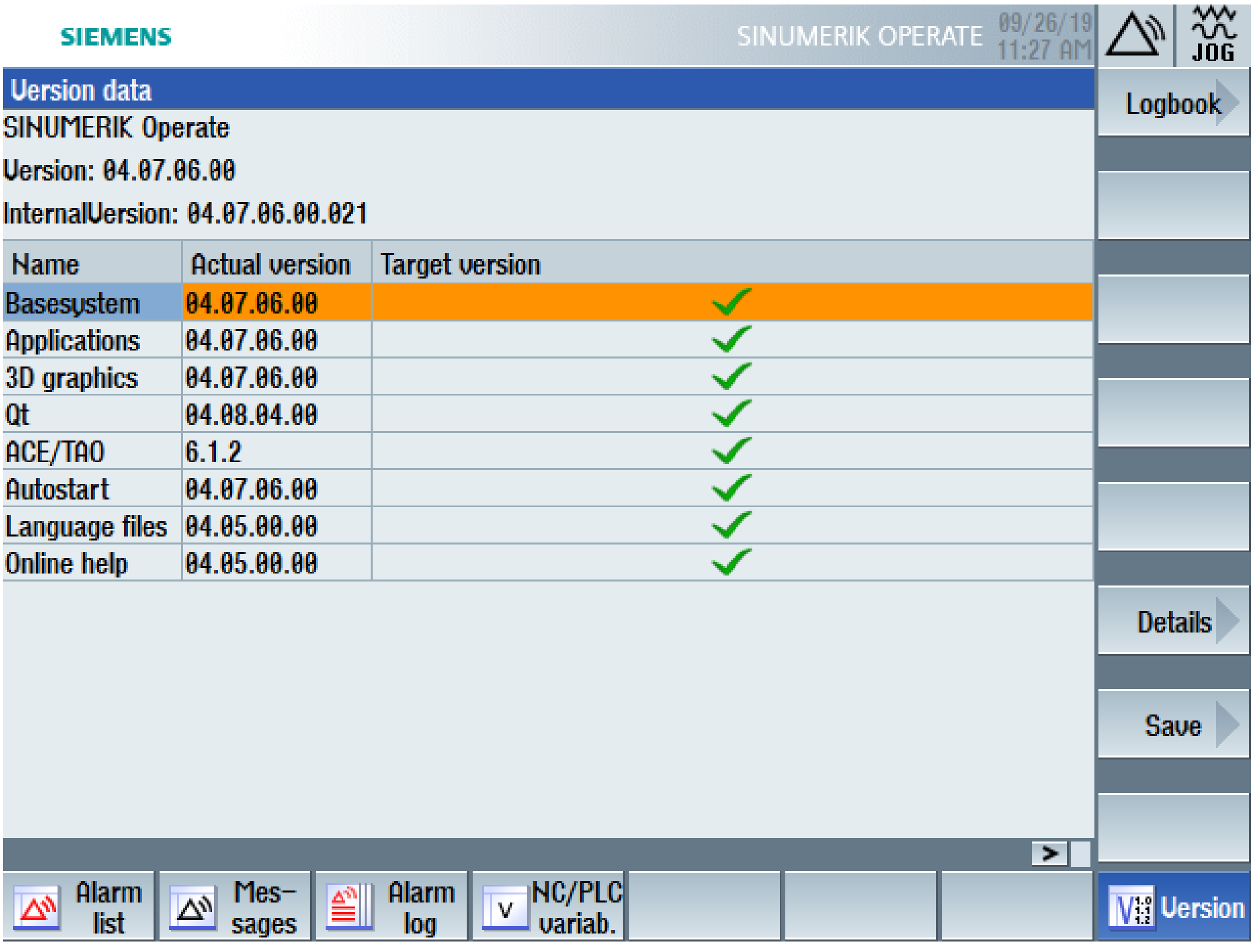


In the Diagnostics area, you can see the following softkeys at the bottom of the screen.



Click on the “Version button”

Now you get an overview of the versions of different components in the NC controller. In the image below, you can see that firmware version 4.7 is used.



Version data

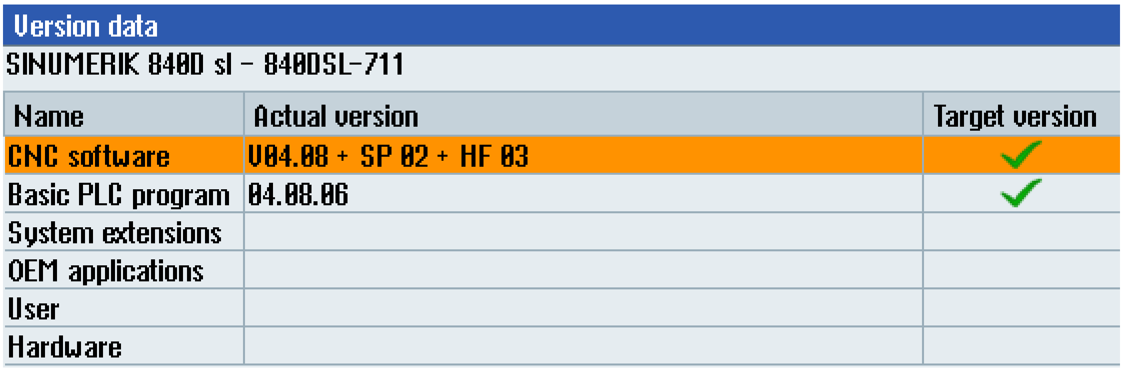
SINUMERIK Operate

Version: 04.07.06.00

InternalVersion: 04.07.06.00.021

Name	Actual version	Target version
Basesystem	04.07.06.00	✓
Applications	04.07.06.00	✓
3D graphics	04.07.06.00	✓
Qt	04.08.04.00	✓
ACE/TAO	6.1.2	✓
Autostart	04.07.06.00	✓
Language files	04.05.00.00	✓
Online help	04.05.00.00	✓

The next image shows an example of the version data of a different controller, with firmware version 4.8.



Version data

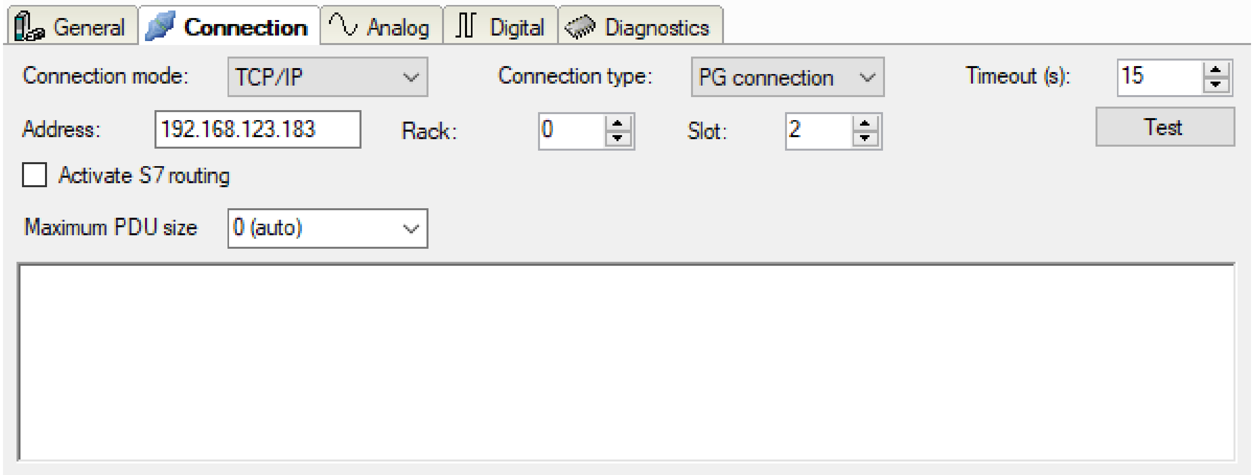
SINUMERIK 840D sl - 840DSL-711

Name	Actual version	Target version
CNC software	V04.08 + SP 02 + HF 03	✓
Basic PLC program	04.08.06	✓
System extensions		
OEM applications		
User		
Hardware		

3.2 Connection

With SINUMERIK-Xplorer, you can connect directly to the NCK part of the NCU. There is no need to change the PLC program to use the Iba system.

In the connection tab, you can make a connection to the NCK CPU of a Sinumerik NCU. Make sure that you connect to the NCK and not to the PLC. Both can have the same IP or MPI address, but it is not possible to read NCK variables using a connection to the PLC part of the NCU.



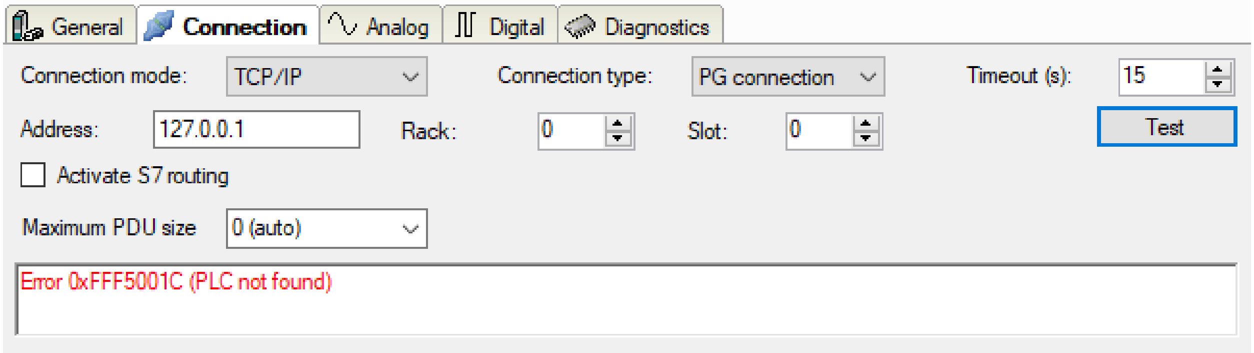
Connection mode: TCP/IP
Connection type: PG connection
Timeout (s): 15
Address: 192.168.123.183
Rack: 0
Slot: 2
□ Activate S7 routing
Maximum PDU size 0 (auto)

The connection mode can be TCP/IP or PC/CP.

When using an MPI/DP to Ethernet converter, like the Helmholz NetLink-PRO compact, you can connect using TCP/IP after the adapter is configured through its web interface. There is then no need to install Siemens SIMATIC NET. However, SIMATIC NET can also be used. Some converters work correctly when connecting to a PLC, but not when connecting to an NCK. Changing the Maximum PDU size to 240 can resolve that issue. In most cases however, the maximum PDU size should be left to have value 0. The PDU size is then automatically negotiated by the network protocol.

When testing a connection with the Test button, there are 3 possible outcomes:

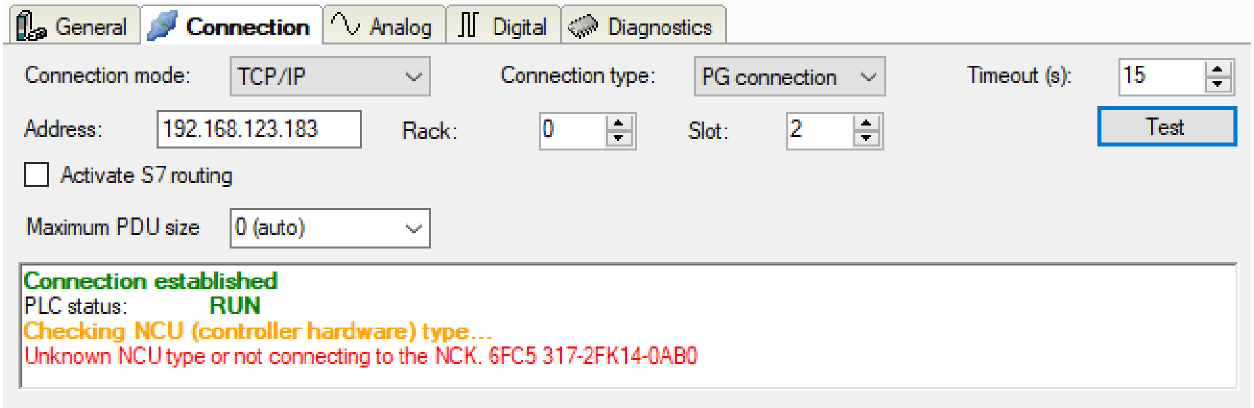
It is possible that no CPU is found:



Connection mode: TCP/IP
Connection type: PG connection
Timeout (s): 15
Address: 127.0.0.1
Rack: 0
Slot: 0
□ Activate S7 routing
Maximum PDU size 0 (auto)

```
Error 0xFFF5001C (PLC not found)
```

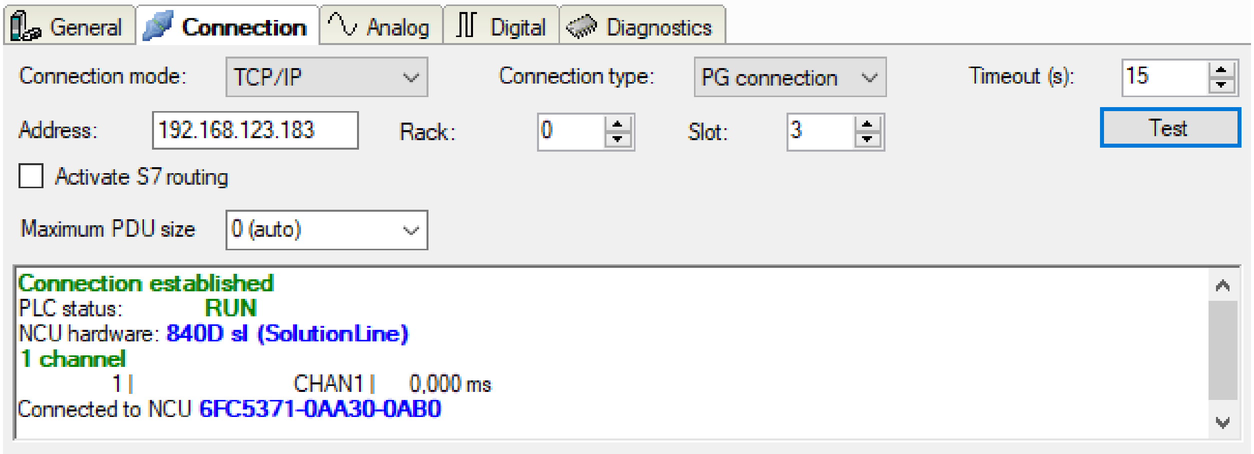
A CPU can be found, but it is not the NCK. In such a case, the MLFB number of the CPU that was found is displayed if possible:



Connection mode: TCP/IP
Connection type: PG connection
Timeout (s): 15
Address: 192.168.123.183
Rack: 0
Slot: 2
□ Activate S7 routing
Maximum PDU size 0 (auto)

```
Connection established
PLC status:         RUN
Checking NCU (controller hardware) type...
Unknown NCU type or not connecting to the NCK. 6FC5 317-2FK14-0AB0
```

Or the NCK is found, which is a successful result for SINUMERIK-Xplorer.



Connection mode: TCP/IP
Connection type: PG connection
Timeout (s): 15
Address: 192.168.123.183
Rack: 0
Slot: 3
□ Activate S7 routing
Maximum PDU size 0 (auto)

```
Connection established
PLC status:         RUN
NCU hardware: 840D sl (SolutionLine)
1 channel
        1 |              CHAN1 |    0,000 ms
Connected to NCU 6FC5371-0AA30-0AB0
```

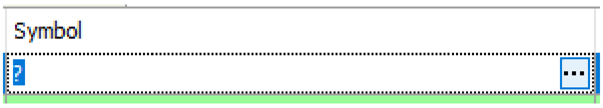
3.3 Adding NCK variables to a Sinumerik module

An NCK variable is determined by 5 parameters, which are shown in both the analog and the digital signal grids. Area, Area Number, Module, Column and Row. Each variable also has a data type, which is determined by the NCK.

3.3.1 Using the signal grids directly

Each of the determining parameters in the signal grids can be changed in the signal grids. In the analog tab, the data type can also be changed.

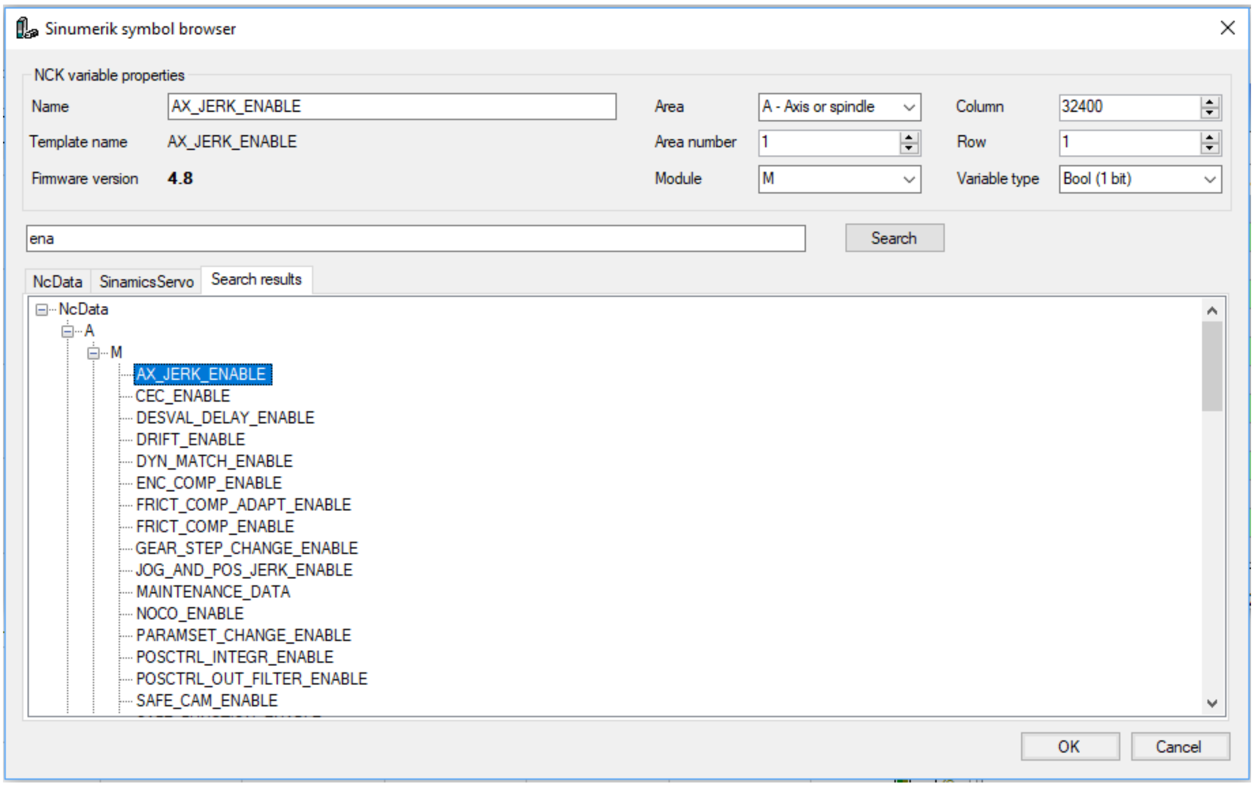
In the Symbol column, an ellipsis button is shown when a cell has focus.



Clicking on the ellipsis button opens the symbol browser for the current row.

3.3.2 Sinumerik symbol browser

The Sinumerik symbol browser can be used to change the content of the signal grids. When opened with the ellipsis button in the Symbol column, it is closed when ok is clicked. When opened with the link in the General tab, it remains open until it is actively closed or Cancel is clicked.


Sinumerik symbol browser

NCK variable properties

Name	AX_JERK_ENABLE	Area	A - Axis or spindle	Column	32400
Template name	AX_JERK_ENABLE	Area number	1	Row	1
Firmware version	**4.8**	Module	M	Variable type	Bool (1 bit)

ena

NcData
A
M
AX_JERK_ENABLE
CEC_ENABLE
DESVAL_DELAY_ENABLE
DRIFT_ENABLE
DYN_MATCH_ENABLE
ENC_COMP_ENABLE
FRICT_COMP_ADAPT_ENABLE
FRICT_COMP_ENABLE
GEAR_STEP_CHANGE_ENABLE
JOG_AND_POS_JERK_ENABLE
MAINTENANCE_DATA
NOCO_ENABLE
PARAMSET_CHANGE_ENABLE
POSCTRL_INTEGR_ENABLE
POSCTRL_OUT_FILTER_ENABLE
SAFE_CAM_ENABLE

Variables with Bool data type will be digital signals, others will be analog. If you want a digital signal to be shown in the analog tab, you can change its data type to char or byte.

You can browse for variables or you can search for them by name.

3.3.3 Diagnostics

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

3.4 Measuring both NCK and Integrated Sinamics signals

The Sinumerik NCU is an advanced numerical control system that internally consists of multiple hardware parts. With SINUMERIK-Xplorer it is possible to gain access to 2 parts of an NCU:

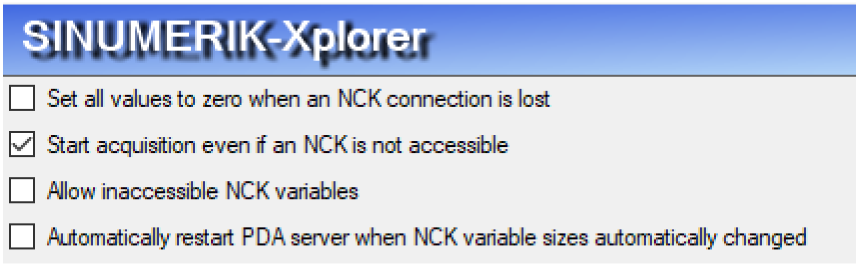
- The NCK, which is the kernel that controls all motion and control functions
- The integrated Sinamics, or anything except NcData in the Symbol Browser.

The NC Integrated Sinamics is not the same as other Sinamics modules. NCU Integrated Sinamics can be accessed with SINUMERIK-Xplorer, while separate Sinamics modules can be accessed with SINAMICS-Xplorer.

In many cases, you will want to access both the NCK and the Integrated Sinamics of the same NCU. However, it is not possible to start requesting NCK and Sinamics variables of the same NCU at exactly the same time.

In order to correctly configure PDA to acquire both NCK and Integrated Sinamics values, the following measures must be taken.

- Integrated Sinamics and NCK signals must be defined in separate modules.
- In the SINUMERIK-Xplorer interface control, “Start acquisition even if an NCK is not accessible” must be checked. That will allow to independently connect to the NCK and the Integrated Sinamics.



SINUMERIK-Xplorer

□ Set all values to zero when an NCK connection is lost

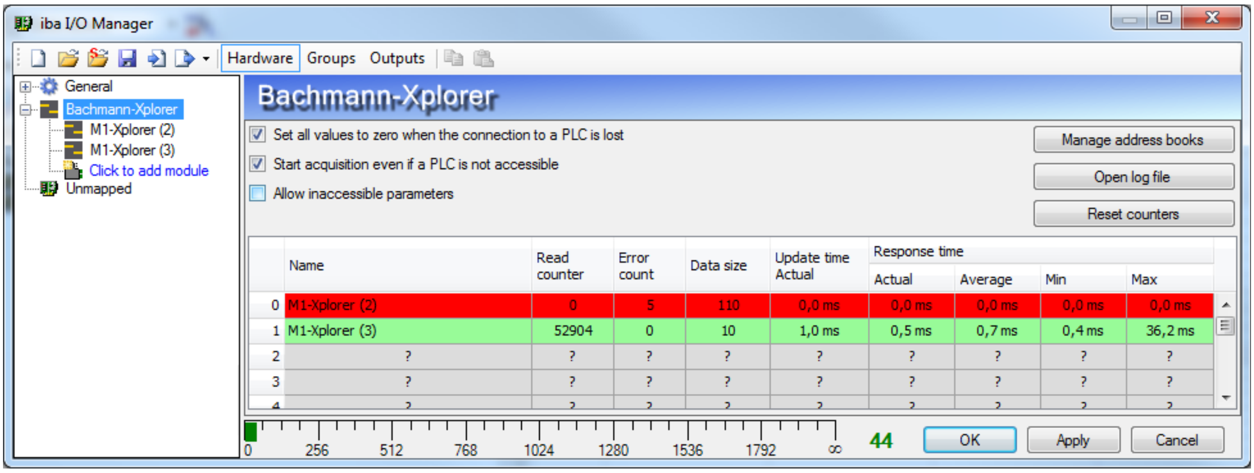
☑ Start acquisition even if an NCK is not accessible

□ Allow inaccessible NCK variables

□ Automatically restart PDA server when NCK variable sizes automatically changed

4 Bachmann-Xplorer Interface

The Bachmann M1 XPlorer interface in ibaPDA is used to measure data from Bachmann M1 PLCs. It is an Xplorer interface which means that the data is cyclically read by ibaPDA instead of being sent by the PLC.



☑ Set all values to zero when the connection to a PLC is lost

☑ Start acquisition even if a PLC is not accessible

□ Allow inaccessible parameters

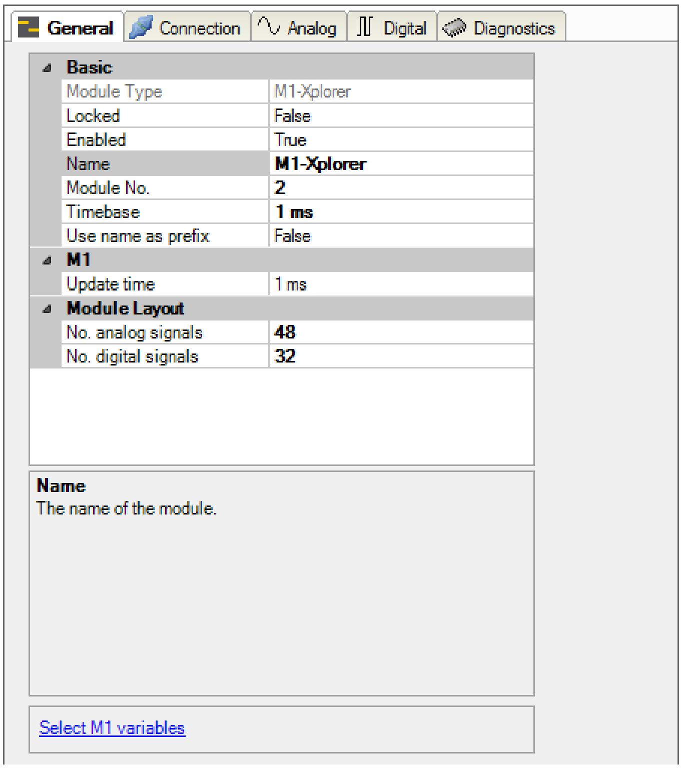
	Name	Read counter	Error count	Data size	Update time Actual	Response time			
						Actual	Average	Min	Max
0	M1-Xplorer (2)	0	5	110	0,0 ms	0,0 ms	0,0 ms	0,0 ms	0,0 ms
1	M1-Xplorer (3)	52904	0	10	1,0 ms	0,5 ms	0,7 ms	0,4 ms	36,2 ms
2	?	?	?	?	?	?	?	?	?
3	?	?	?	?	?	?	?	?	?
4	?	?	?	?	?	?	?	?	?

The Bachmann-Xplorer interface shows a table of the available connections. Per Bachmann-Xplorer license you get 16 connections .A maximum of 240 connections is allowed. This means that maximum 15 licenses can be used. Each connection corresponds to a row in the table. The row is green when the connection is ok and data is being read. The rows is orange when the connection is ok but the data is coming slower than the configured update time. The row is red when the connection could not be established. The row is grey when there is no connection configured. The response time is the time it takes to read the data for a connection. The table shows the actual, average, minimum and maximum values of the response time. The update time is the time between 2 read operations. The data size shows how much data is read per read operation; in between brackets is the number of commands used to request the data. You can use the “Reset counters” button to clear the counters for all connections. Clicking “Open log file” opens the most recent log file related to Bachmann-Xplorer connections.

On the interface you can also decide how to handle some error conditions:

- When the connection to a PLC is lost during the acquisition then you can choose if the values stay at the last read value or if they are set to zero.
- When a PLC is not accessible during the start of the acquisition then you can choose if the acquisition starts without this PLC or if the acquisition is not started. When the acquisition is started without the PLC then ibaPDA will periodically (every 10s) try to connect to the PLC during the acquisition. As long as the PLC is disconnected the values will remain at zero.
- When ibaPDA tries to access an operand that is not available when validating the I/O configuration, the PLC will return an error. If the option “Allow inaccessible operands” is enabled then ibaPDA will ignore this signal and start the acquisition without this signal. If the option is not enabled then the acquisition will not start.

The General tab of a M1-Xplorer module looks as follows:

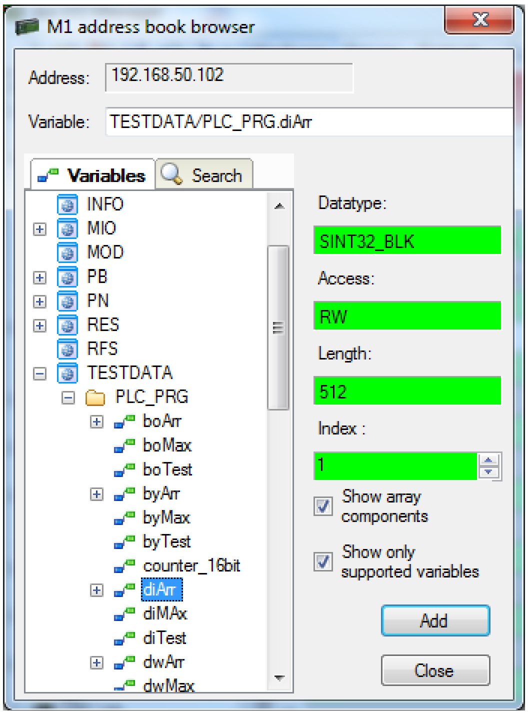


Basic	
Module Type	M1-Xplorer
Locked	False
Enabled	True
Name	**M1-Xplorer**
Module No.	**2**
Timebase	**1 ms**
Use name as prefix	False
M1	
Update time	1 ms
Module Layout	
No. analog signals	**48**
No. digital signals	**32**

Name
The name of the module.

Select M1 variables

Apart from the standard options that can be found on most other modules, the **Update time** has to be configured. This is the time in ms between two read operations.
New signals can be added by clicking **Select M1 variables** at the bottom.

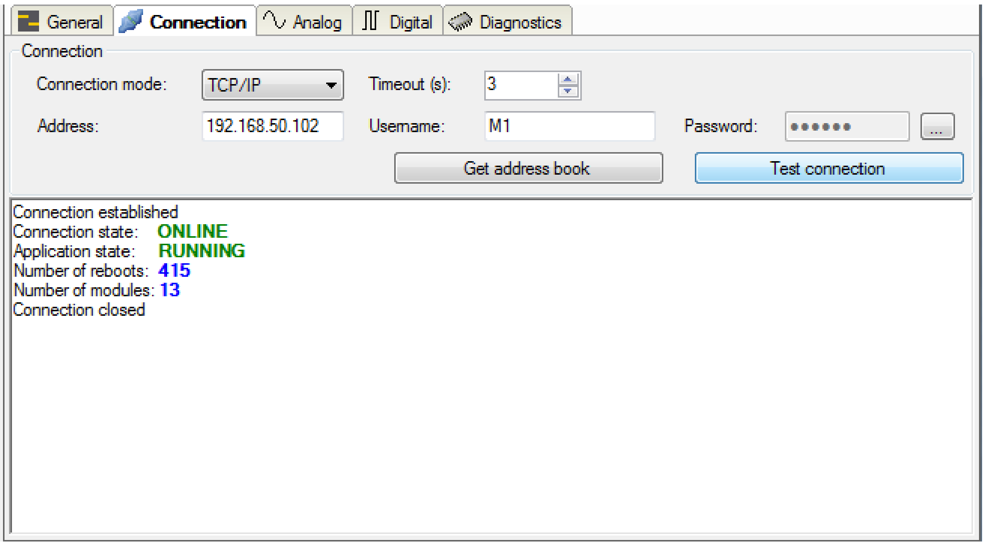


Address: 192.168.50.102
Variable: TESTDATA/PLC_PRG.diArr

Datatype: SINT32_BLK
Access: RW
Length: 512
Index : 1

☑ Show array components
☑ Show only supported variables

Using the M1 address book browser you can easily add analog or digital signals to the M1-Xplorer module, by double-clicking any variable, or selecting multiple variables and clicking on “Add”



Connection mode: TCP/IP

Timeout (s): 3

Address: 192.168.50.102

Username: M1

Password: ●●●●●●

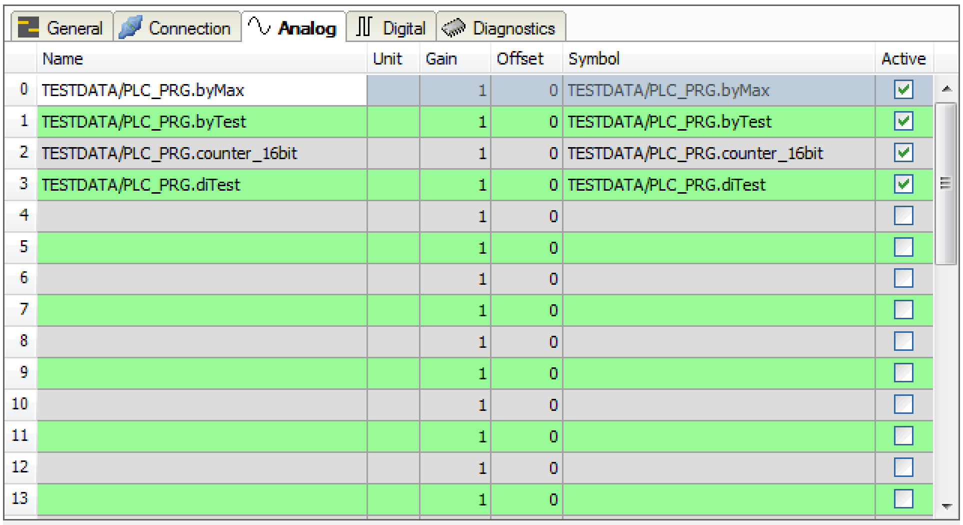
```
Connection established
Connection state:    ONLINE
Application state:   RUNNING
Number of reboots:  415
Number of modules: 13
Connection closed
```

In the **Connection** tab, all parameters to establish a proper connection to the PLC have to be configured:

- **Protocol**: select either TCP/IP, QSOAP or SSL. Note that some PLC types only support TCP/IP; refer to your PLC's manual for more information.
- **Address**: The IP address of the PLC at which the network interface of the PLC is located.
- **Timeout**: the amount of time after which a connection attempt will be aborted.
- **Username**: The username needed to access the PLC, according the the PLC configuration.
- **Password**: The password corresponding to the selected username

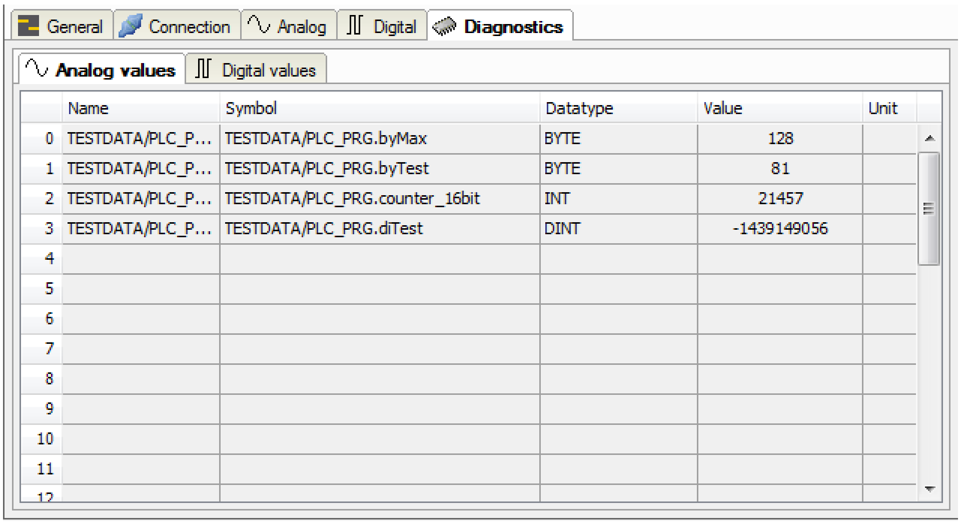
When clicking the **Test Connection** button, ibaPDA will try to establish a connection to the PLC.

When clicking the **Get address book** button, ibaPDA will try to establish a connection to the PLC and read the address book, overwriting any previous address book related to the selected IP address.



	Name	Unit	Gain	Offset	Symbol	Active
0	TESTDATA/PLC_PRG.byMax		1	0	TESTDATA/PLC_PRG.byMax	☑
1	TESTDATA/PLC_PRG.byTest		1	0	TESTDATA/PLC_PRG.byTest	☑
2	TESTDATA/PLC_PRG.counter_16bit		1	0	TESTDATA/PLC_PRG.counter_16bit	☑
3	TESTDATA/PLC_PRG.diTest		1	0	TESTDATA/PLC_PRG.diTest	☑
4			1	0		□
5			1	0		□
6			1	0		□
7			1	0		□
8			1	0		□
9			1	0		□
10			1	0		□
11			1	0		□
12			1	0		□
13			1	0		□

The above figure shows an example of the Analog tab of a M1-Xplorer module. The datatype is internally synchronized to the current address book.

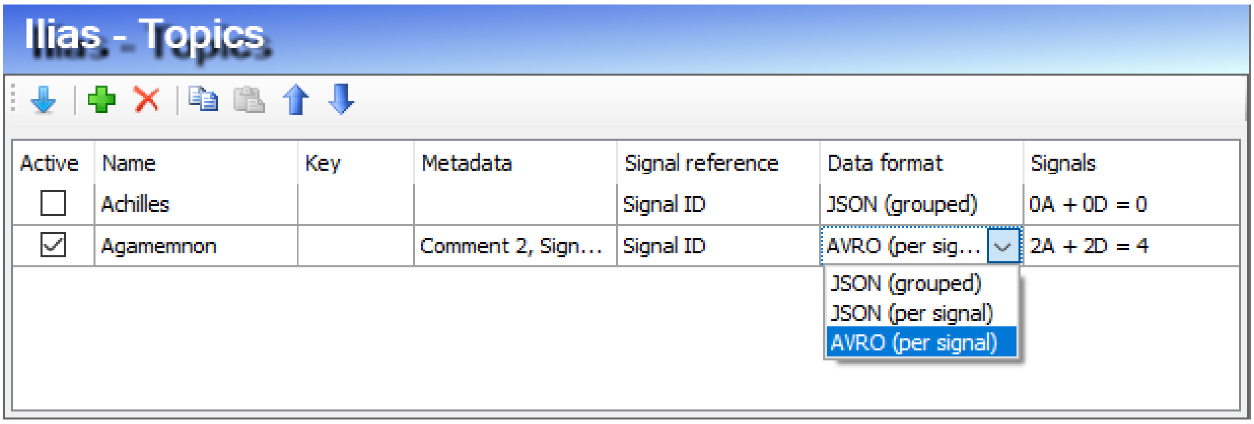


	Name	Symbol	Datatype	Value	Unit
0	TESTDATA/PLC_P...	TESTDATA/PLC_PRG.byMax	BYTE	128	
1	TESTDATA/PLC_P...	TESTDATA/PLC_PRG.byTest	BYTE	81	
2	TESTDATA/PLC_P...	TESTDATA/PLC_PRG.counter_16bit	INT	21457	
3	TESTDATA/PLC_P...	TESTDATA/PLC_PRG.diTest	DINT	-1439149056	
4					
5					
6					
7					
8					
9					
10					
11					
12					

The current values of the requested topics can be monitored in the **Diagnostics** tab. There a further distinction is made between **Analog values** and **Digital values**.

5 Kafka data store: AVRO encoding support

The Kafka data store now supports an additional encoding format: Apache AVRO. Compared to JSON, which stores data in human-readable format, AVRO uses binary encoding which reduces bandwidth and required storage space. For more information regarding this format we refer to https://avro.apache.org/.



Active	Name	Key	Metadata	Signal reference	Data format	Signals
☐	Achilles			Signal ID	JSON (grouped)	0A + 0D = 0
☑	Agamemnon		Comment 2, Sign...	Signal ID	AVRO (per sig...	2A + 2D = 4

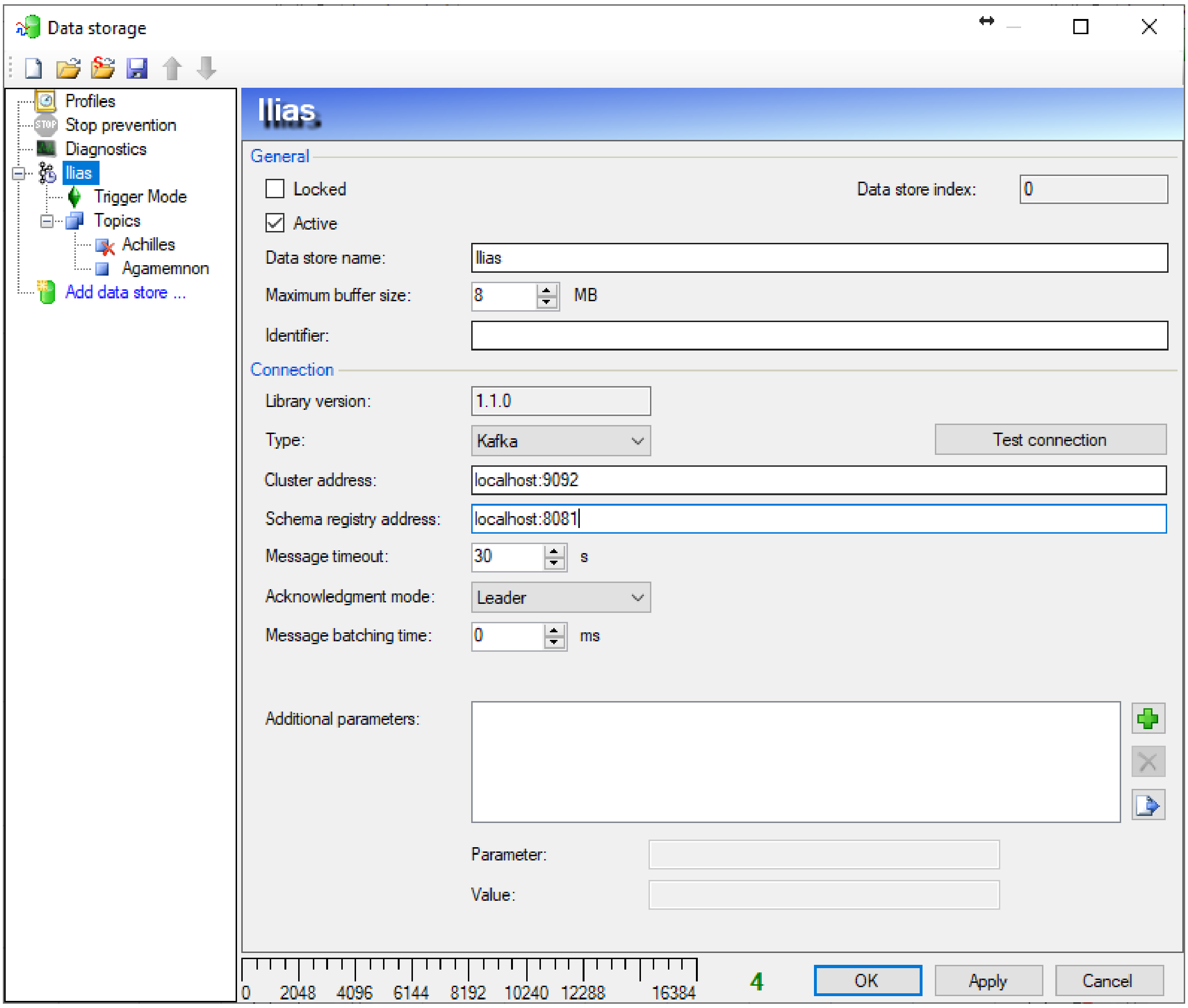
As shown in the picture above, currently encoding data in AVRO is only supported when writing a single record per signal per sample.

ibaPDA uses the following schema to serialize signal data:

```
{
   "namespace": "de.iba",
   "type": "record",
   "name": "PdaRecord",
   "fields": [
      {"name": "Signal", "type": "string"},
      {"name": "ID", "type": ["null", "string"]},
      {"name": "Name", "type": ["null", "string"]},
      {"name": "Unit", "type": ["null", "string"]},
      {"name": "Comment1", "type": ["null", "string"]},
      {"name": "Comment2", "type": ["null", "string"]},
      {"name": "Timestamp", "type": [
         "null",
         {"type": "long", "logicalType": "timestamp-micros"}
      ]},
      {"name": "Identifier", "type": ["null", "string"]},
      {"name": "ValueType", "type": {
         "type": "enum",
         "name": "ValueTypeEnum",
         "symbols": ["BOOLEAN", "BYTES", "DOUBLE", "FLOAT", "INT", "LONG", "STRING"]
      }},
      {"name": "BooleanValue", "type": ["null", "boolean"]},
      {"name": "BytesValue", "type": ["null", "bytes"]},
      {"name": "DoubleValue", "type": ["null", "double"]},
      {"name": "FloatValue", "type": ["null", "float"]},
      {"name": "IntValue", "type": ["null", "int"]},
      {"name": "LongValue", "type": ["null", "long"]},
      {"name": "StringValue", "type": ["null", "string"]}
   ]
}
```

ibaPDA also supports a connection to a Confluent Schema Registry cluster; this is however not required so this field can be left blank. When a Schema registry is configured, ibaPDA will register the schema used for encoding with the Schema registry. The returned ID is then attached to each record as follows:

Byte offset	Description
0	0x00 (Confluent AVRO Marker)
1	Schema ID (Big Endian)
5	Signal data



□ Locked

☑ Active

Data store index: 0

Data store name: Ilias

Maximum buffer size: 8 MB

Identifier:

Library version: 1.1.0

Type: Kafka

Cluster address: localhost:9092

Schema registry address: localhost:8081

Message timeout: 30 s

Acknowledgment mode: Leader

Message batching time: 0 ms

Additional parameters:

Parameter:

Value:

The address and port of the Schema registry can be entered in the main configuration tab of the Kafka data store under **Schema registry address**. When clicking **Test connection** ibaPDA will also try to establish a connection to the Schema registry (if a non-empty address is provided).

When a Schema registry is not used each record is encoded as follows:

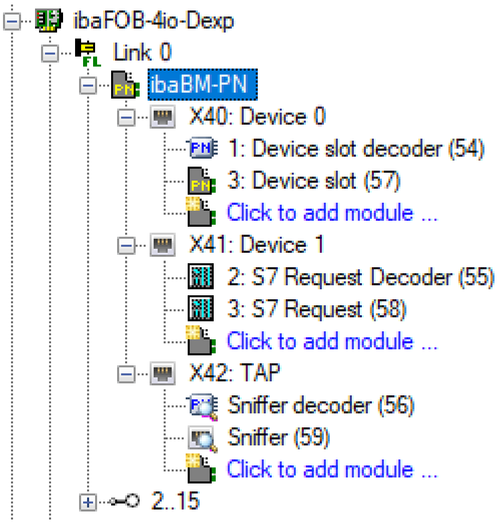
Byte offset	Description
0	0xC3 0x01 (Single-object encoding marker)
2	CRC-64-AVRO fingerprint of encoding schema
10	Signal data

6 ibaBM-PN decoder modules

There are 2 new module types available on the ibaBM-PN device:

- Device slot decoder
- Sniffer decoder

Decoder modules allow you to measure digital signals that are packed together in WORDs. This is similar to the ibaBM-DP decoder modules. In all decoder modules you can configure the number of decoders (= WORDs that need to be decoded in digital signals).



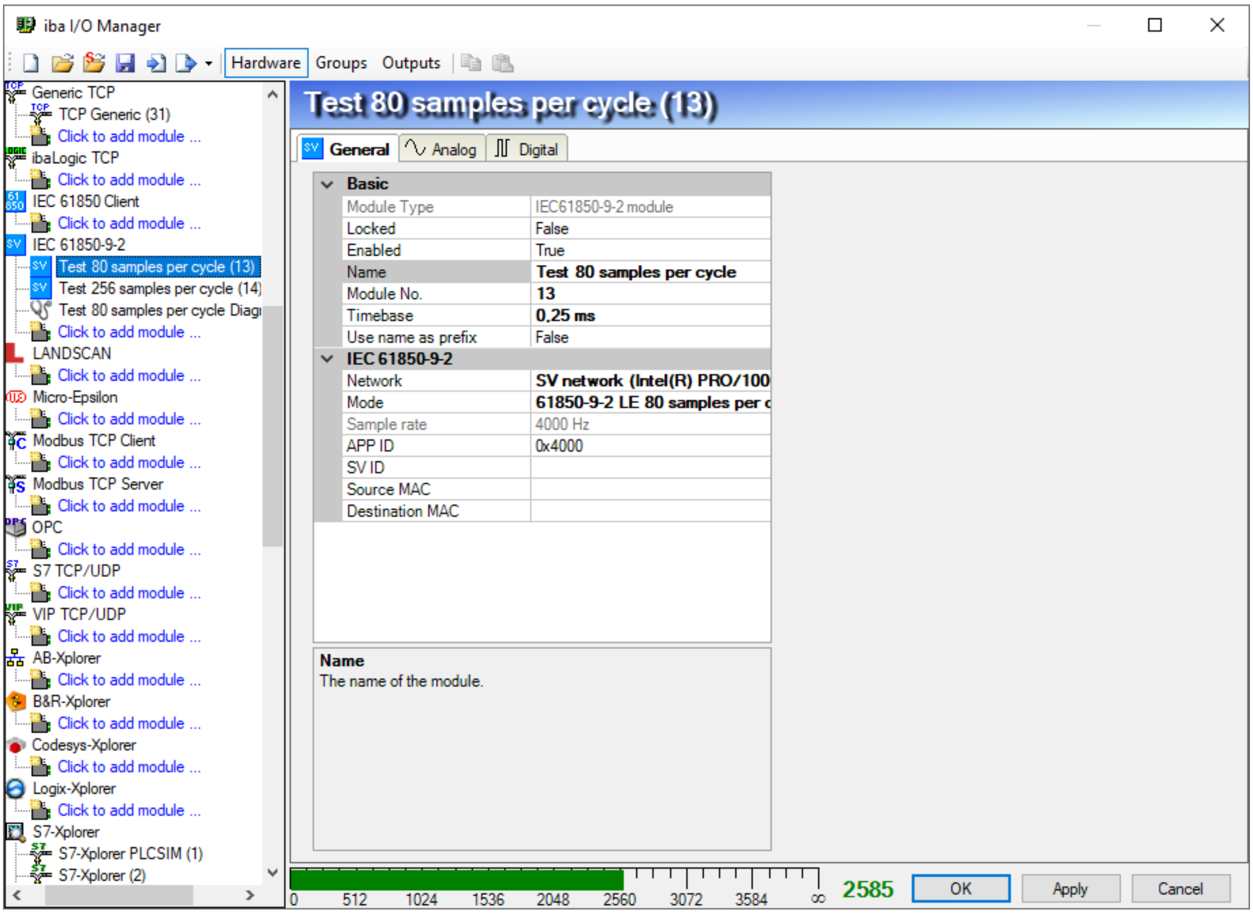
ibaFOB-4io-Dexp
Link 0
ibaBM-PN
X40: Device 0
1: Device slot decoder (54)
3: Device slot (57)
Click to add module ...
X41: Device 1
2: S7 Request Decoder (55)
3: S7 Request (58)
Click to add module ...
X42: TAP
Sniffer decoder (56)
Sniffer (59)
Click to add module ...
2..15

The device slot decoder is the decoder variant of the device slot module. It measures data from a slot of the PROFINET device embedded in the ibaBM-PN. You have to configure the address and data type WORD (little endian) or WORD_B (big endian) for each decoder.

The sniffer decoder is the decoder variant of the sniffer module. It maps to the X42: TAP connector. It measures data that is being exchanged between a PROFINET device and its PROFINET controller. You have to configure the direction, slot, subslot and address within the subslot for each decoder.

7 IEC 61850-9-2 interface

IEC 61850 is an international standard defining communication protocols for intelligent electronic devices at electrical substations. Part 9-2 describes the so-called Sampled Values. These are currents and voltages that are measured and sent via Ethernet frames in realtime. IbaPDA is now able to measure up to 8 Sampled Values streams.



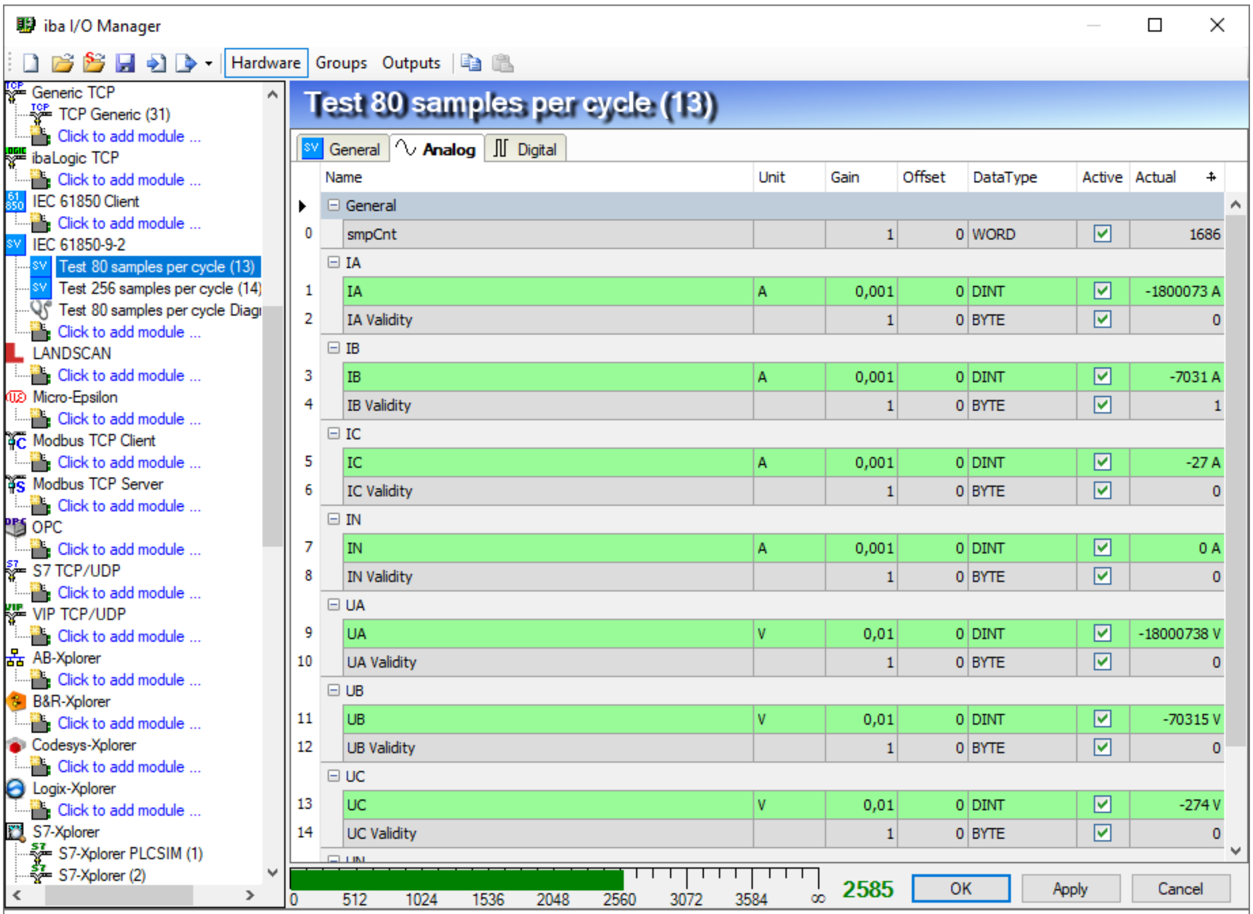
Basic	
Module Type	IEC61850-9-2 module
Locked	False
Enabled	True
Name	**Test 80 samples per cycle**
Module No.	**13**
Timebase	**0.25 ms**
Use name as prefix	False
IEC 61850-9-2	
Network	**SV network (Intel(R) PRO/100**
Mode	**61850-9-2 LE 80 samples per c**
Sample rate	4000 Hz
APP ID	0x4000
SV ID	
Source MAC	
Destination MAC	

Name
The name of the module.

Add an IEC 61850-9-2 module to the IEC 61850-9-2 interface. Each module corresponds with one Sampled Values stream. In the IEC 61850-9-2 section you have to configure all properties of the stream. The first thing to configure is the *Network* property. It determines the network interface where the stream is being received on. The *Mode* determines the format of the Sampled Values and the sample rate. There exist 6 standard modes:

- 61850-9-2 LE 80 samples per cycle on 50Hz
- 61850-9-2 LE 80 samples per cycle on 60Hz
- 61850-9-2 LE 256 samples per cycle on 50Hz
- 61850-9-2 LE 256 samples per cycle on 60Hz
- 61869-9 4800 Hz
- 61869-9 14400 Hz

In all these standard modes the data consists of 4 currents and 4 voltages. For each current and voltage there is also a quality available. This quality is automatically decoded in different analog and digital signals by ibaPDA. The screenshots below show the standard analog and digital signals. The *Sample rate* property is automatically configured for the standard modes. It is therefor readonly.

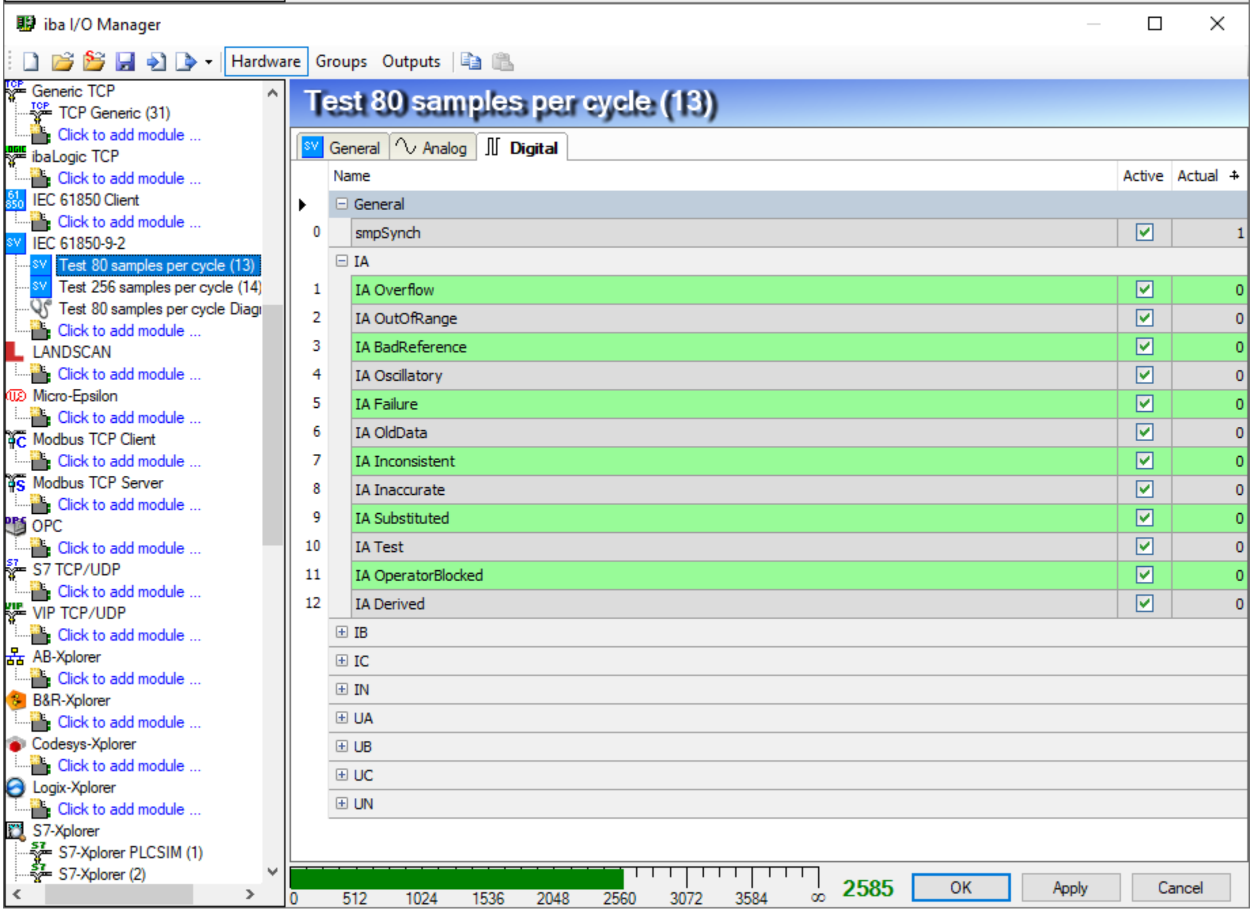


iba I/O Manager

- Generic TCP
  - TCP Generic (31)
  - Click to add module ...
- ibaLogic TCP
  - Click to add module ...
- IEC 61850 Client
  - Click to add module ...
- IEC 61850-9-2
  - Test 80 samples per cycle (13)
  - Test 256 samples per cycle (14)
  - Test 80 samples per cycle Diagı
  - Click to add module ...
- LANDSCAN
  - Click to add module ...
- Micro-Epsilon
  - Click to add module ...
- Modbus TCP Client
  - Click to add module ...
- Modbus TCP Server
  - Click to add module ...
- OPC
  - Click to add module ...
- S7 TCP/UDP
  - Click to add module ...
- VIP TCP/UDP
  - Click to add module ...
- AB-Xplorer
  - Click to add module ...
- B&R-Xplorer
  - Click to add module ...
- Codesys-Xplorer
  - Click to add module ...
- Logix-Xplorer
  - Click to add module ...
- S7-Xplorer
  - S7-Xplorer PLCSIM (1)
  - S7-Xplorer (2)

Test 80 samples per cycle (13)

	Name	Unit	Gain	Offset	DataType	Active	Actual
	⊟ General						
0	smpCnt		1	0	WORD	☑	1686
	⊟ IA						
1	IA	A	0,001	0	DINT	☑	-1800073 A
2	IA Validity		1	0	BYTE	☑	0
	⊟ IB						
3	IB	A	0,001	0	DINT	☑	-7031 A
4	IB Validity		1	0	BYTE	☑	1
	⊟ IC						
5	IC	A	0,001	0	DINT	☑	-27 A
6	IC Validity		1	0	BYTE	☑	0
	⊟ IN						
7	IN	A	0,001	0	DINT	☑	0 A
8	IN Validity		1	0	BYTE	☑	0
	⊟ UA						
9	UA	V	0,01	0	DINT	☑	-18000738 V
10	UA Validity		1	0	BYTE	☑	0
	⊟ UB						
11	UB	V	0,01	0	DINT	☑	-70315 V
12	UB Validity		1	0	BYTE	☑	0
	⊟ UC						
13	UC	V	0,01	0	DINT	☑	-274 V
14	UC Validity		1	0	BYTE	☑	0



iba I/O Manager

- Generic TCP
  - TCP Generic (31)
  - Click to add module ...
- ibaLogic TCP
  - Click to add module ...
- IEC 61850 Client
  - Click to add module ...
- IEC 61850-9-2
  - Test 80 samples per cycle (13)
  - Test 256 samples per cycle (14)
  - Test 80 samples per cycle Diagı
  - Click to add module ...
- LANDSCAN
  - Click to add module ...
- Micro-Epsilon
  - Click to add module ...
- Modbus TCP Client
  - Click to add module ...
- Modbus TCP Server
  - Click to add module ...
- OPC
  - Click to add module ...
- S7 TCP/UDP
  - Click to add module ...
- VIP TCP/UDP
  - Click to add module ...
- AB-Xplorer
  - Click to add module ...
- B&R-Xplorer
  - Click to add module ...
- Codesys-Xplorer
  - Click to add module ...
- Logix-Xplorer
  - Click to add module ...
- S7-Xplorer
  - S7-Xplorer PLCSIM (1)
  - S7-Xplorer (2)

Test 80 samples per cycle (13)

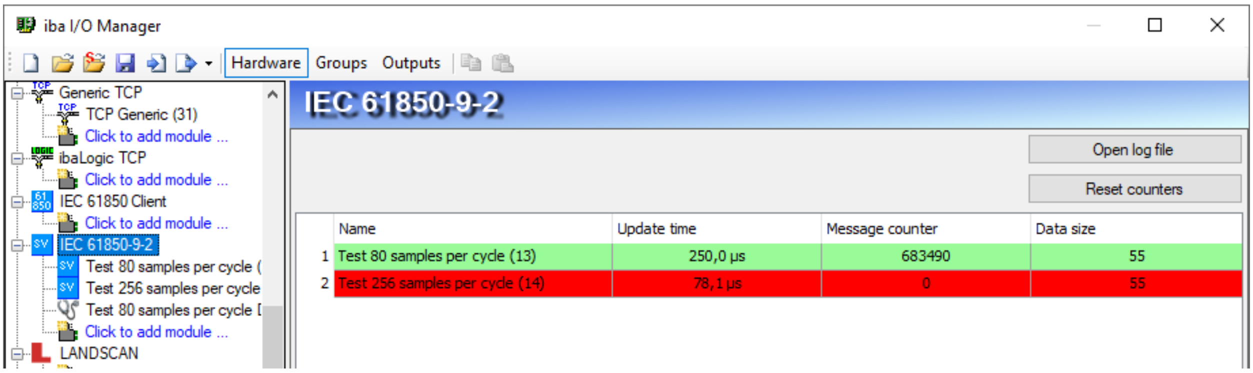
	Name	Active	Actual
	⊟ General		
0	smpSynch	☑	1
	⊟ IA		
1	IA Overflow	☑	0
2	IA OutOfRange	☑	0
3	IA BadReference	☑	0
4	IA Oscillatory	☑	0
5	IA Failure	☑	0
6	IA OldData	☑	0
7	IA Inconsistent	☑	0
8	IA Inaccurate	☑	0
9	IA Substituted	☑	0
10	IA Test	☑	0
11	IA OperatorBlocked	☑	0
12	IA Derived	☑	0
	⊞ IB		
	⊞ IC		
	⊞ IN		
	⊞ UA		
	⊞ UB		
	⊞ UC		
	⊞ UN		

There also exists a custom mode. In this mode you have to configure the sample rate yourself. You can also configure all the signals yourself. You can change the number of signals and you can configure their offsets and data types.

The other 4 properties are used to select the correct Sampled Values stream that arrives on the configured network interface.

- The *APP ID* is the application identifier of the Sampled Values stream. It is a 16-bit integer number that is sent in each message. You have to configure it in hex.
- The *SV ID* is called the Sampled Values Identifier. It is a string of maximum 35 characters that is sent with each sample. If it is configured then only samples that match the configured value will be measured. If it is left blank then the SV ID isn't checked.
- The *Source MAC* can be used to filter on the source address of the Ethernet frames. If it is left blank then all source MAC addresses are accepted.
- The *Destination MAC* can be used to filter on the destination address of the Ethernet frames. If it is left blank then all destination MAC addresses are accepted.

The Sampled Values streams are **always resampled** with the *Timebase* configured in the module. Even when the *Timebase* is the same as the nominal sample period there will be occasional duplicate or skipped samples. This is because the sample clock in ibaPDA isn't 100% the same as the sample clock in the IEC 61850-9-2 device.



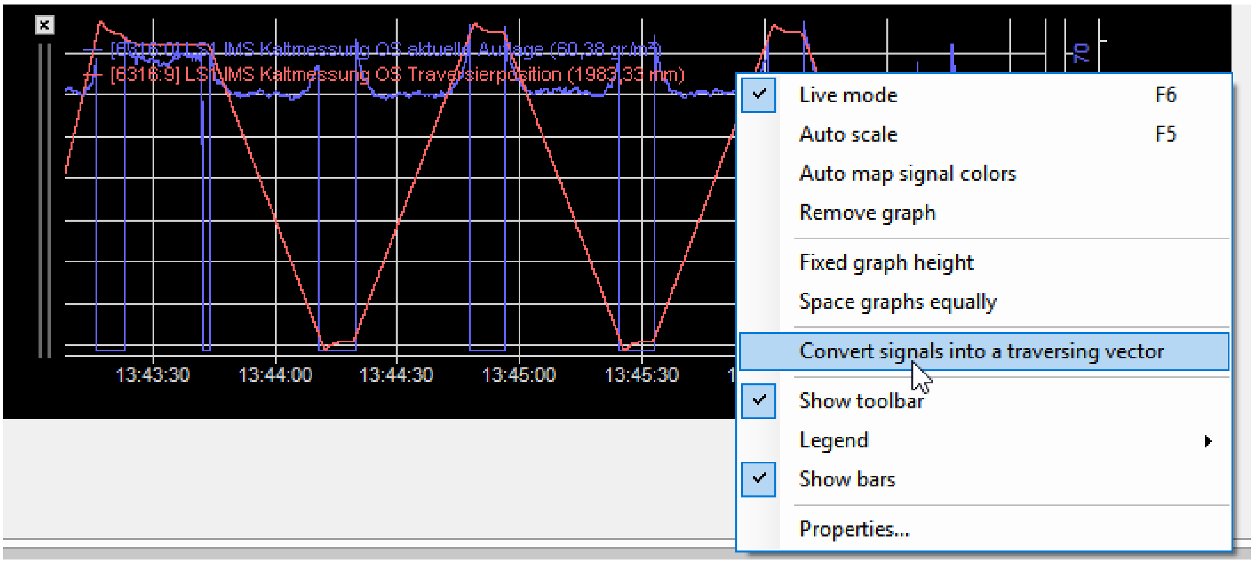
	Name	Update time	Message counter	Data size
1	Test 80 samples per cycle (13)	250,0 µs	683490	55
2	Test 256 samples per cycle (14)	78,1 µs	0	55

The IEC 61850-9-2 interface contains a table that shows the configured Sampled Values streams. The color determines if messages are being received (green) or not (red). The update time shows the measured time between samples. The message counter shows how many messages have been received. One message can contain multiple samples. This depends on the stream's mode. The *Reset counters* button can be used to reset the message counter. The data size shows how much data ibaPDA measures for each sample.

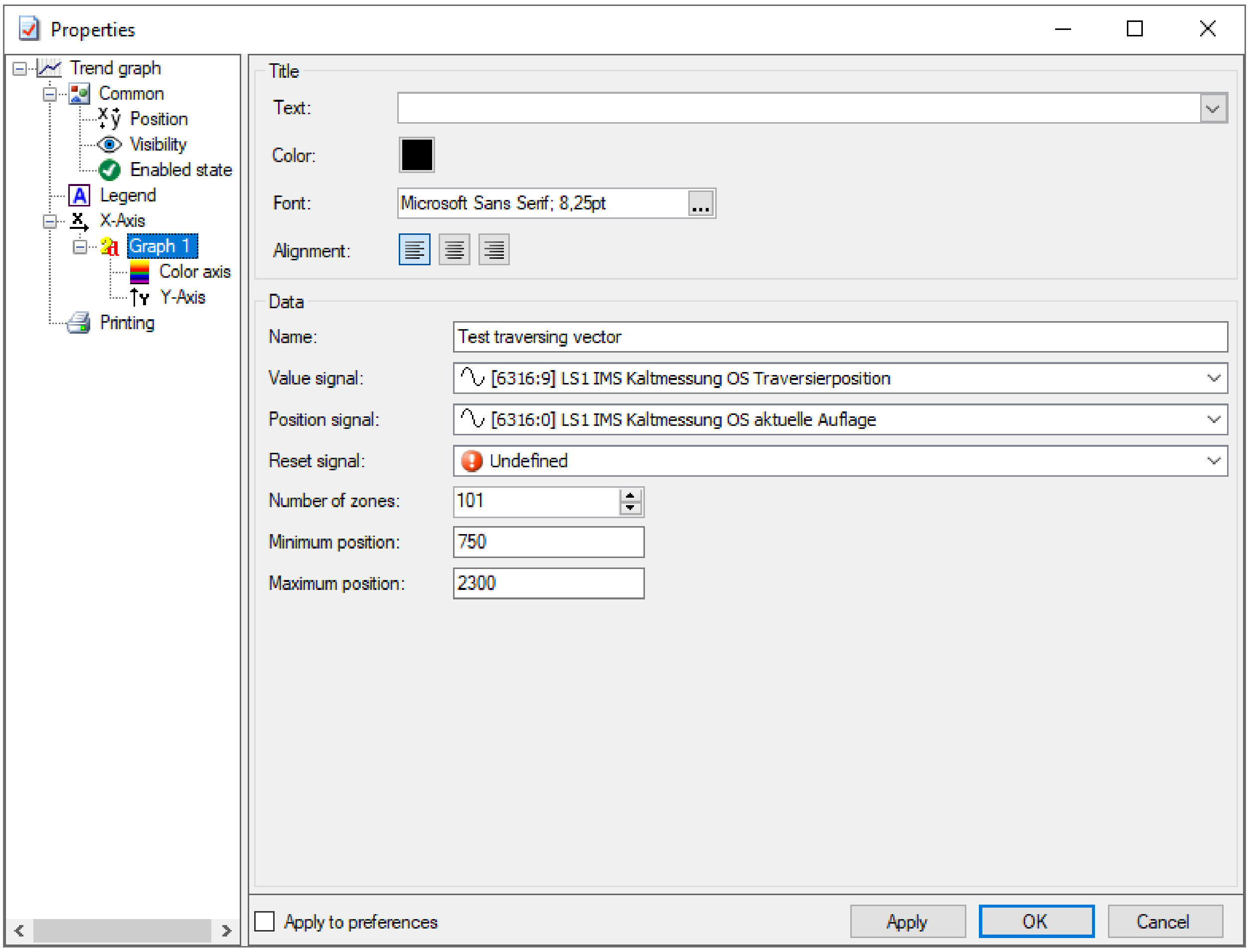
The diagnostic data from this connection table can also be measured via a diagnostics module. Each diagnostics module can be coupled to a Sampled Values stream via its *Target module* property.

8 Traversing vector

A traversing vector is a vector that is generated by 2 signals: a position signal and a value signal. It is supported in the timebased trend graph in ibaQPanel and in the timebased and lengthbased offline trend graph in ibaQPanel.



When you have a graph with 2 signals in it then a new command appears in the context menu: *Convert signals into a traversing vector*. When you click this the traversing vector properties form is opened.



Title

Text:

Color: ■

Font: Microsoft Sans Serif; 8,25pt

Alignment:

Data

Name: Test traversing vector

Value signal: [6316:9] LS1 IMS Kaltmessung OS Traversierposition

Position signal: [6316:0] LS1 IMS Kaltmessung OS aktuelle Auflage

Reset signal: Undefined

Number of zones: 101

Minimum position: 750

Maximum position: 2300

☐ Apply to preferences

Here you can give the vector a name. You also have to make sure that the correct signals are assigned to the value and position signals. You have to define the number of zones you want to generate. You also need to define the range of the position signal that will be divided into zones. In the example 101 zones will be generated between 750 and 2300. The zones are created as follows. The zone width is calculated as (Max - Min) / (Number of zones - 1). The zones are positioned like this:

```
 |----|--------|--------|--------|----|

Min       M1       M2       M3       Max
```

Z0 = from Min to Min+ZoneWidth/2

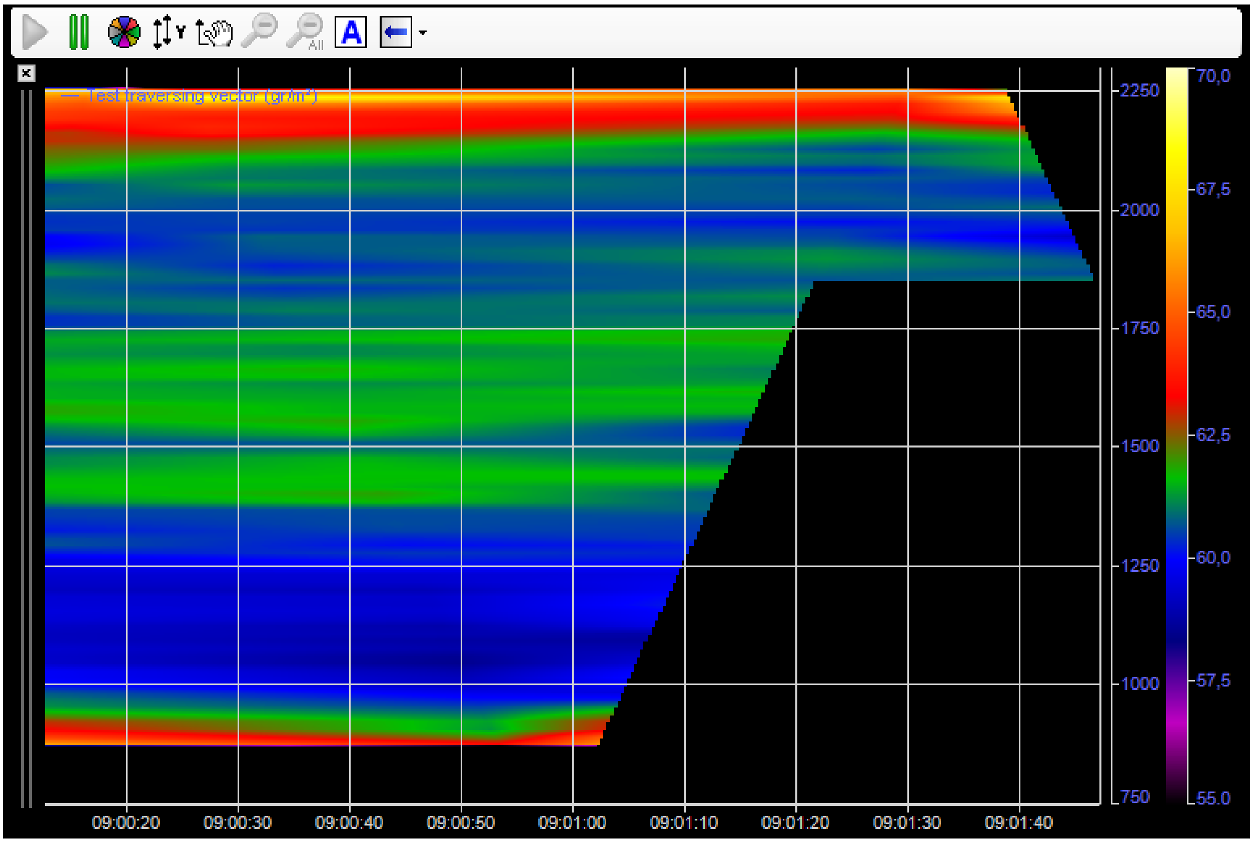
Z1 = from M1 - ZoneWidth/2 to M1 + ZoneWidth/2 with M1 = Min + ZoneWidth

Z2 = from M2 - ZoneWidth/2 to M2 + ZoneWidth/2 with M2 = Min + ZoneWidth*2

Z3 = from M3 - ZoneWidth/2 to M3 + ZoneWidth/2 with M3 = Min + ZoneWidth*3

Z4 = from Max - ZoneWidth/2 to Max

So the first and the last zones have half the width of the other zones. As long as the position signal is within a zone the values of the value signal are averaged. When the position signal leaves a zone the averaged value is saved at the middle of the zone. When the position signal is less than Min or greater than Max the values are ignored and no new zone data is created.

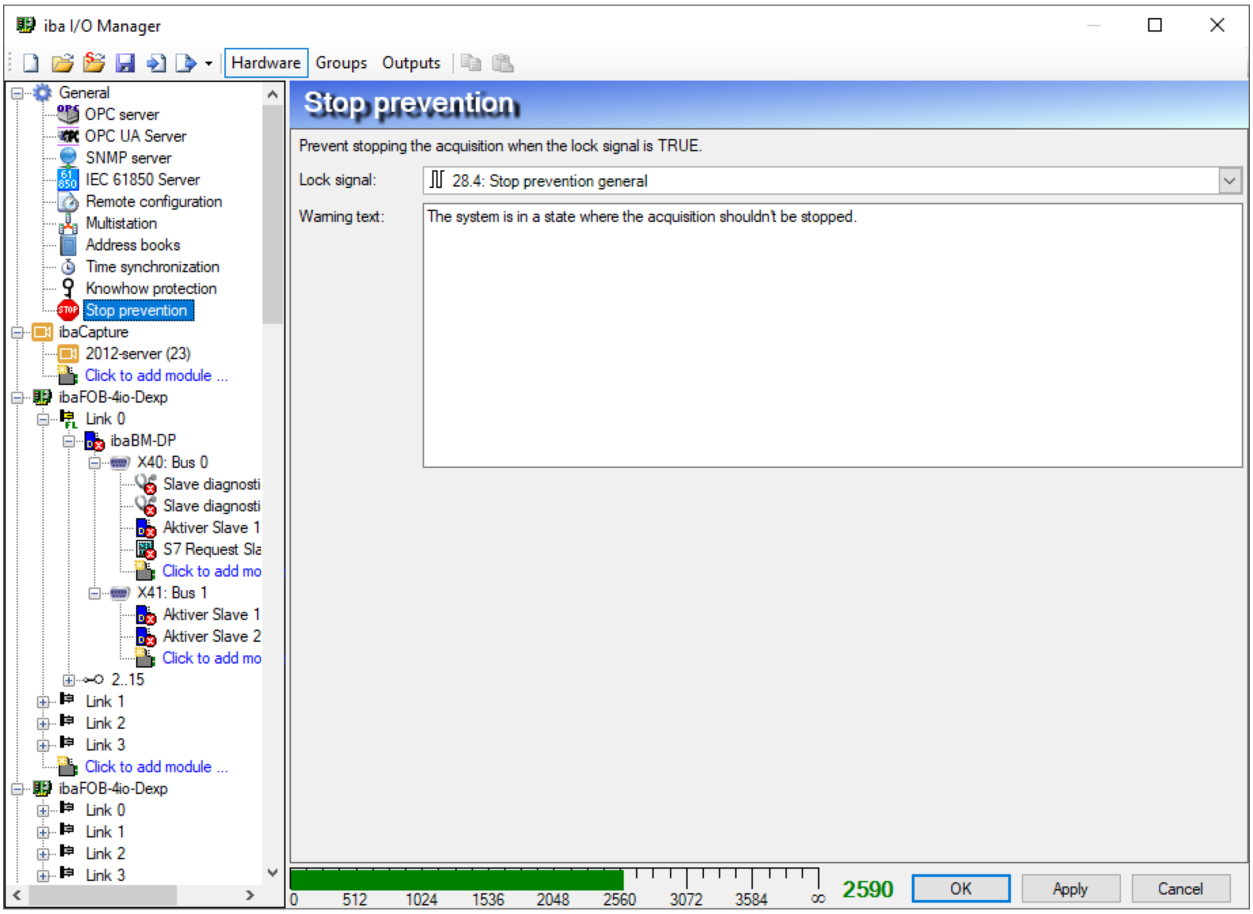


The values between 2 samples of the same zone are interpolated. This means that the data can only be drawn when there are 2 samples available for a zone. That is why you see these sloping edges.

There is also an optional reset signal available. When this signal is TRUE the position signal is ignored. When the signal is FALSE again new zone data is generated but it will not be connected to the previous zone data.

9 Stop prevention

9.1 Acquisition stop prevention

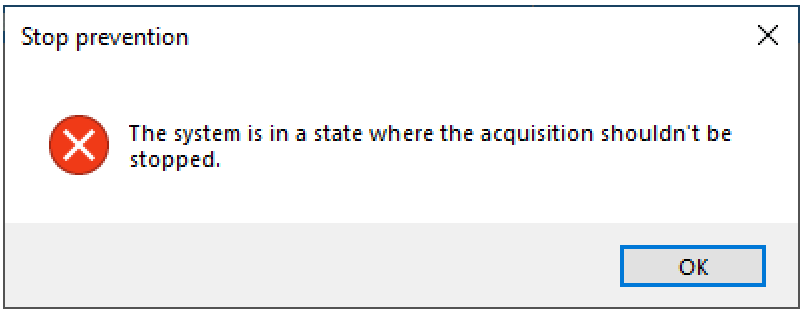


Prevent stopping the acquisition when the lock signal is TRUE.

Lock signal: 28.4: Stop prevention general

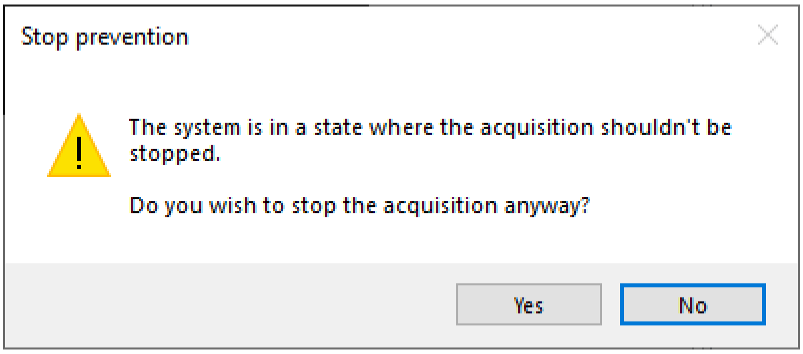
Warning text: The system is in a state where the acquisition shouldn't be stopped.

In the I/O manager you can configure a lock signal that signals a situation in which it isn't allowed to stop the acquisition. You can enter your own warning text. This text will be shown in the client when it tries to stop the acquisition while the lock signal is TRUE.



The system is in a state where the acquisition shouldn't be stopped.

In some cases this stop prevention can be overruled. The following message box will be shown then.



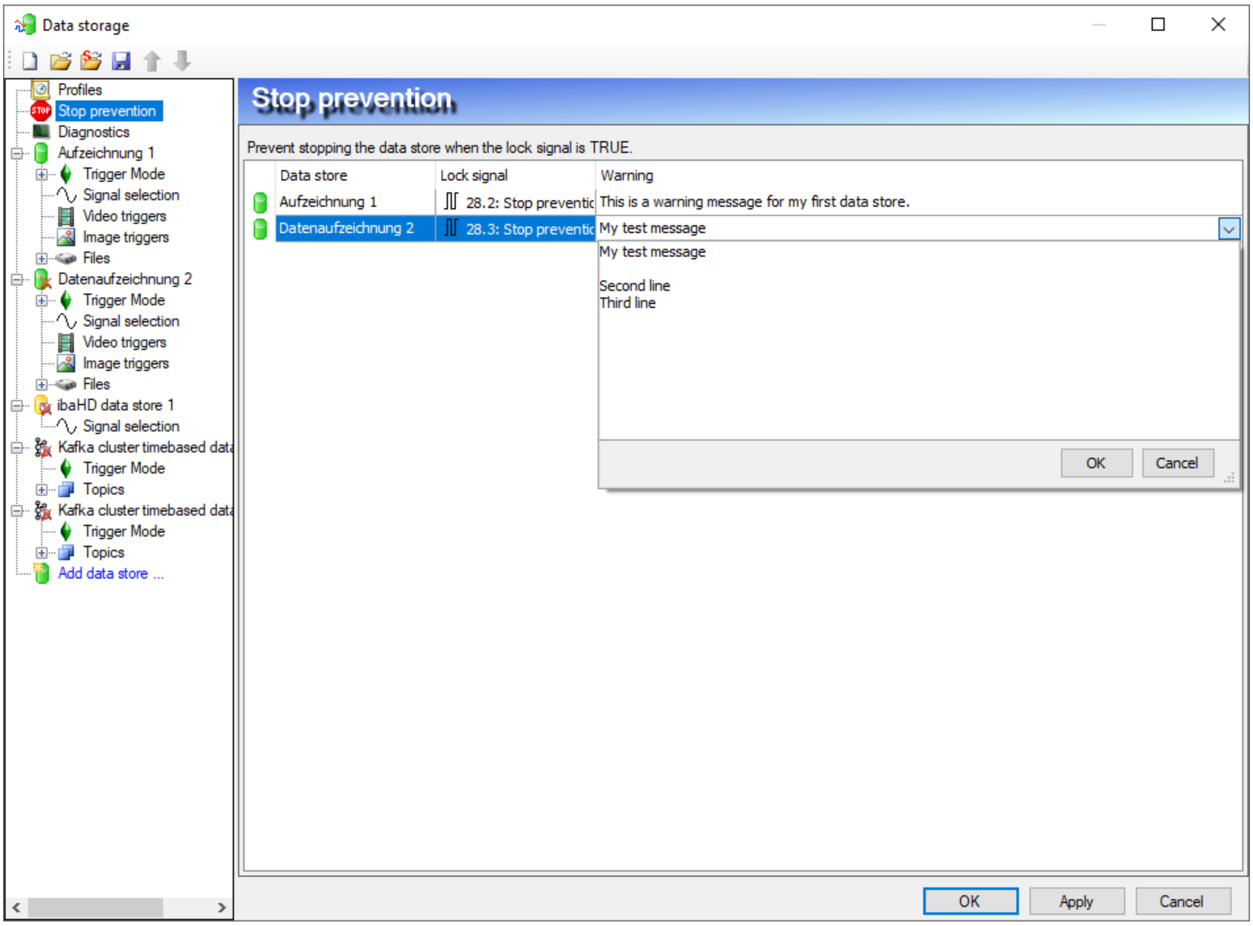
Stop prevention

The system is in a state where the acquisition shouldn't be stopped.

Do you wish to stop the acquisition anyway?

If the stop prevention is overruled then this is logged in the event log. The event log message contains the user that overruled the stop prevention.

9.2 Data storage stop prevention



Prevent stopping the data store when the lock signal is TRUE.

Data store	Lock signal	Warning
Aufzeichnung 1	28.2: Stop preventic	This is a warning message for my first data store.
Datenaufzeichnung 2	28.3: Stop preventic	My test message

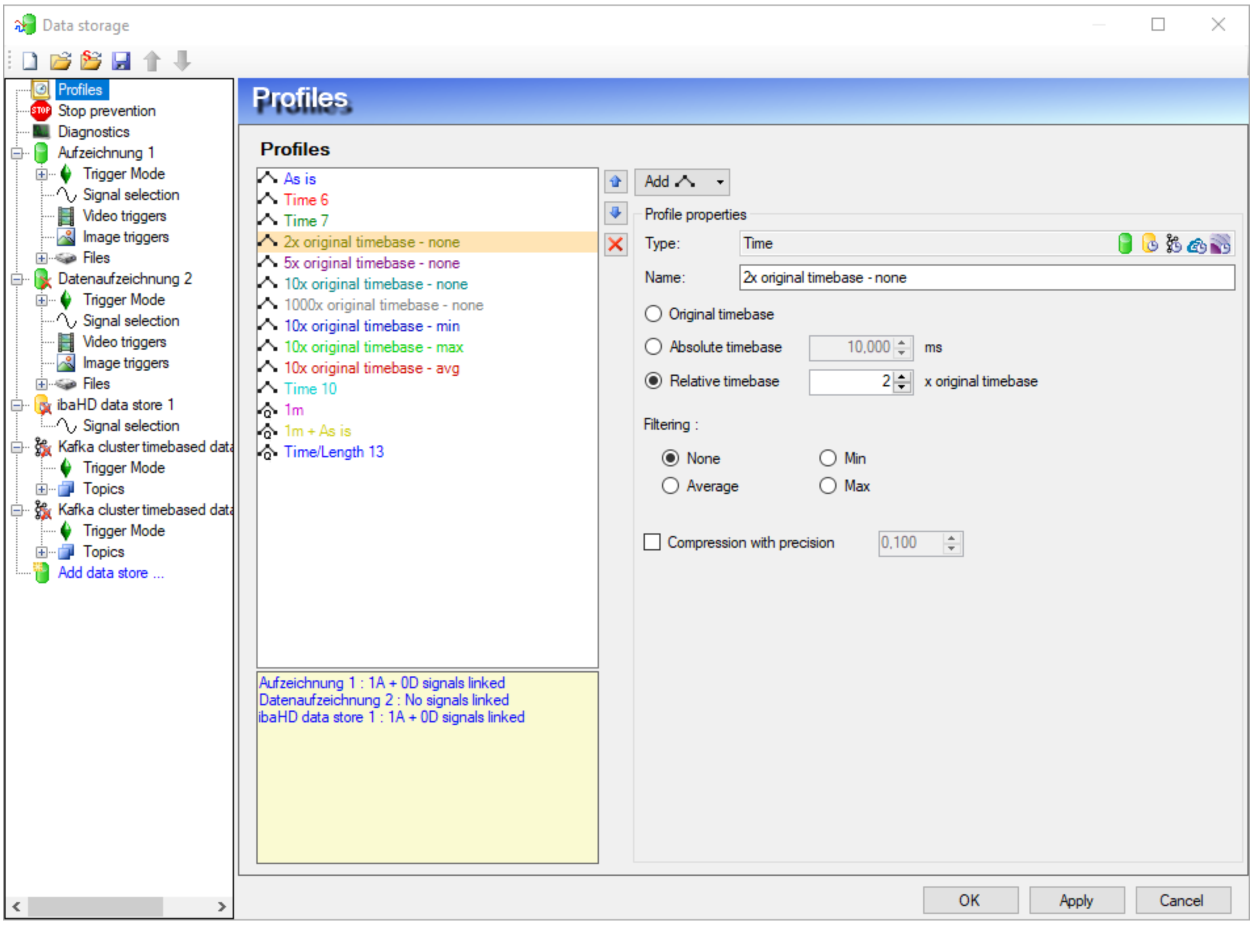
My test message

Second line
Third line

In the data storage manager you can configure a lock signal per data store that signals a situation in which it isn't allowed to interrupt the data store. You can also configure a warning message per store. The message can contain multiple lines.

The data storage stop prevention works in the same way as the acquisition stop prevention.

10 Storage profiles with relative timebase



Profiles

- As is
- Time 6
- Time 7
- 2x original timebase - none
- 5x original timebase - none
- 10x original timebase - none
- 1000x original timebase - none
- 10x original timebase - min
- 10x original timebase - max
- 10x original timebase - avg
- Time 10
- 1m
- 1m + As is
- Time/Length 13

Aufzeichnung 1 : 1A + 0D signals linked
Datenaufzeichnung 2 : No signals linked
ibaHD data store 1 : 1A + 0D signals linked

Profile properties

Type: Time

Name: 2x original timebase - none

○ Original timebase
○ Absolute timebase 10,000 ms
● Relative timebase 2 x original timebase

Filtering :
● None ○ Min
○ Average ○ Max

□ Compression with precision 0,100

The standard storage profile has been extended with the option to use a relative timebase instead of an absolute timebase. The relative timebase is expressed as a number of times the original timebase.