



ibaPDA-Interface-MQTT

MQTT Communication Interface

Manual
Issue 1.6

Measurement Systems for Industry and Energy
www.iba-ag.com

Manufacturer

iba AG
Koenigswarterstrasse 44
90762 Fuerth
Germany

Contacts

Main office	+49 911 97282-0
Support	+49 911 97282-14
Engineering	+49 911 97282-13
E-mail	iba@iba-ag.com
Web	www.iba-ag.com

Unless explicitly stated to the contrary, it is not permitted to pass on or copy this document, nor to make use of its contents or disclose its contents. Infringements are liable for compensation.

© iba AG 2025, All rights reserved.

The content of this publication has been checked for compliance with the described hardware and software. Nevertheless, discrepancies cannot be ruled out, and we do not provide guarantee for complete conformity. However, the information furnished in this publication is updated regularly. Required corrections are contained in the following regulations or can be downloaded on the Internet.

The current version is available for download on our web site www.iba-ag.com.

Version	Date	Revision	Author	Version SW
1.6	07-2025	Topic browser	rm, st	8.10.0

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

Contents

1	About this documentation	4
1.1	Target group and previous knowledge	4
1.2	Notations	4
1.3	Used symbols.....	5
2	System requirements	6
3	MQTT interface	7
3.1	General information	7
3.2	System topologies.....	8
3.3	Configuration and engineering ibaPDA.....	9
3.3.1	Interface settings	9
3.3.2	Adding a module.....	9
3.3.3	General module settings.....	10
3.3.4	Connection settings	12
3.3.5	Signal configuration	15
3.3.5.1	Selecting topics via the topic browser	16
3.3.6	Outputs.....	17
4	Diagnostics.....	21
4.1	License	21
4.2	Connection table	21
4.3	Diagnostics in the I/O Manager	22
4.4	Connection diagnostics with PING.....	23
4.5	Diagnostic modules	24
5	Support and contact.....	29

1 About this documentation

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

Other documentation



This documentation is a supplement to the *ibaPDA* manual. Information about all the other characteristics and functions of *ibaPDA* can be found in the *ibaPDA* manual or in the online help.

1.1 Target group and previous knowledge

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

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

- Windows operating system
- Basic knowledge of *ibaPDA*
- Basic knowledge of the MQTT protocol

1.2 Notations

In this manual, the following notations are used:

Action	Notation
Menu command	Menu <i>Logic diagram</i>
Calling the menu command	<i>Step 1 – Step 2 – Step 3 – Step x</i> Example: Select the menu <i>Logic diagram – Add – New function block</i> .
Keys	<Key name> Example: <Alt>; <F1>
Press the keys simultaneously	<Key name> + <Key name> Example: <Alt> + <Ctrl>
Buttons	<Key name> Example: <OK>; <Cancel>
Filenames, paths	<i>Filename, Path</i> Example: <i>Test.docx</i>

1.3 Used symbols

If safety instructions or other notes are used in this manual, they mean:

Danger!



The non-observance of this safety information may result in an imminent risk of death or severe injury:

- Observe the specified measures.
-

Warning!



The non-observance of this safety information may result in a potential risk of death or severe injury!

- Observe the specified measures.
-

Caution!



The non-observance of this safety information may result in a potential risk of injury or material damage!

- Observe the specified measures
-

Note



A note specifies special requirements or actions to be observed.

Tip



Tip or example as a helpful note or insider tip to make the work a little bit easier.

Other documentation



Reference to additional documentation or further reading.

2 System requirements

The following system requirements are necessary for the use of the MQTT data interface:

- *ibaPDA* v8.10.0 or more recent
- License for *ibaPDA-Interface-MQTT*
- Network connection to one or more MQTT brokers (not part of this interface)

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

Order no.	Product name	Description
31.001112	ibaPDA-Interface-MQTT	Extension license for an ibaPDA system adding the data interface MQTT

Functional constraints

ibaPDA cannot acquire single Publish Messages if multiple of those are packed in one MQTT telegram. In this case only the last Publish Message will be acquired and made available for display and data recording.

3 MQTT interface

Using *ibaPDA-Interface-MQTT*, *ibaPDA* supports MQTT v3.1.1 to receive data via an MQTT broker or to send data to an MQTT broker.

3.1 General information

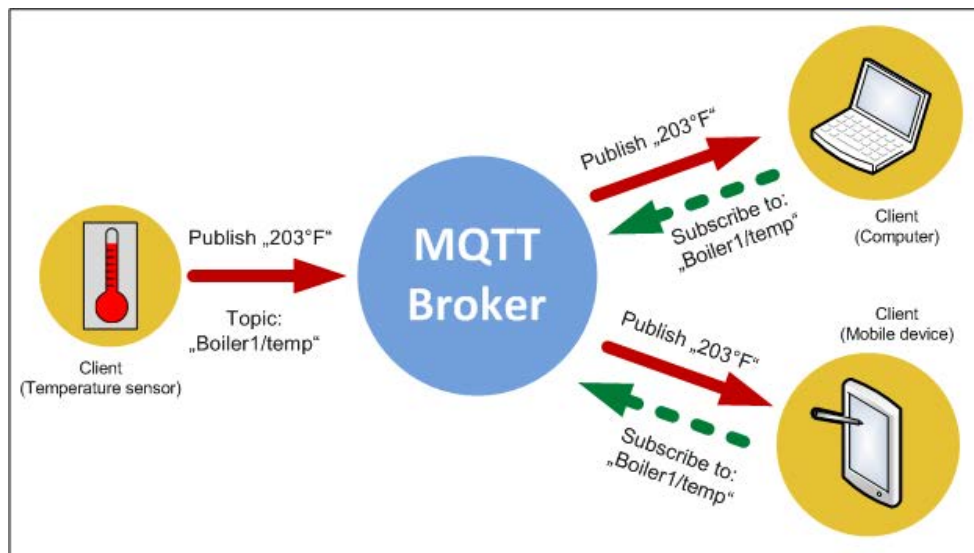
MQTT (Message Queue Telemetry Transport) is a communication protocol that has been developed especially for M2M (Machine-to-Machine) communication and has become increasingly important in the context of the Internet of Things (IoT).

It is characterized by the following properties:

- It enables simple networking of devices with few resources, such as sensors
- It offers various service qualities (Quality of service), which ensures data transmission even in unstable networks
- Metainformation is stored on the server side so that it does not have to be sent again after a connection has been interrupted and re-established
- It enables the transfer of different data types without defining a defined structure

Functional principle

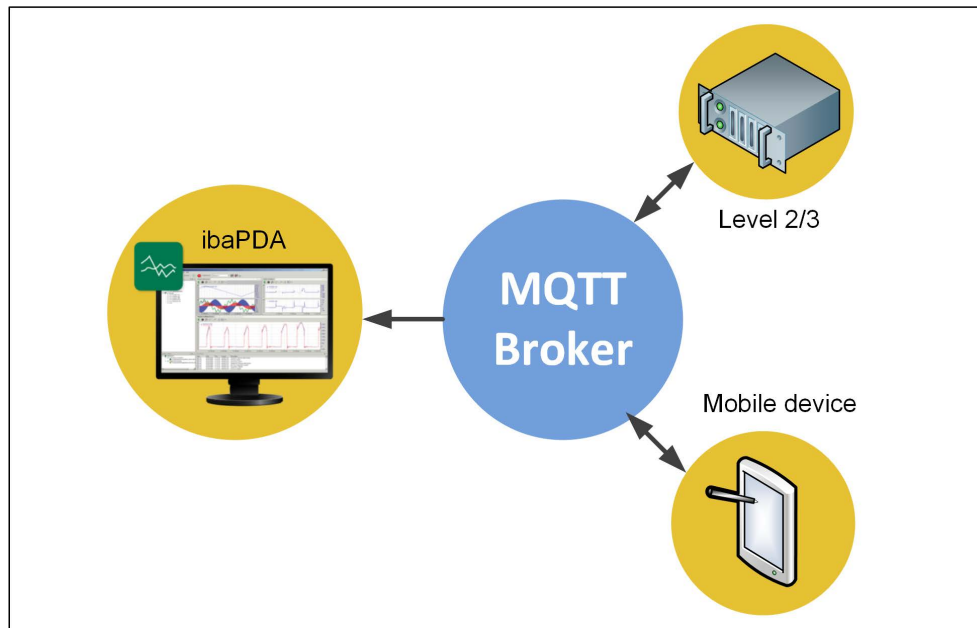
MQTT is based on an event-driven publish/subscribe architecture. The core is a central server (broker) to which both sender and receiver connect. The data is sent (published) or received (subscribed) via so-called topics. Topics are quasi communication channels into which transmitters, e.g. sensors, write their data. The broker checks which receivers (clients) have opened a channel for this data and forwards the data to the clients.



Concerning *ibaPDA*, this means that *ibaPDA* is positioned as an MQTT client that can send data to the MQTT broker and receive data from the MQTT broker.

As a client *ibaPDA* subscribes to all topics with the measured values that are to be acquired and made available via the MQTT broker. These can be, for example, data from higher-level systems of levels 2 and 3 or from sensors that publish their measured values via MQTT.

Via the outputs of the MQTT interface, *ibaPDA* can send data to the MQTT broker



Note



The MQTT-Broker is not included in *ibaPDA*. This must be purchased, installed and configured separately. The transmission behaviour is significantly influenced by the configuration and performance of the broker.

3.2 System topologies

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

No further software is necessary for operation.

Note



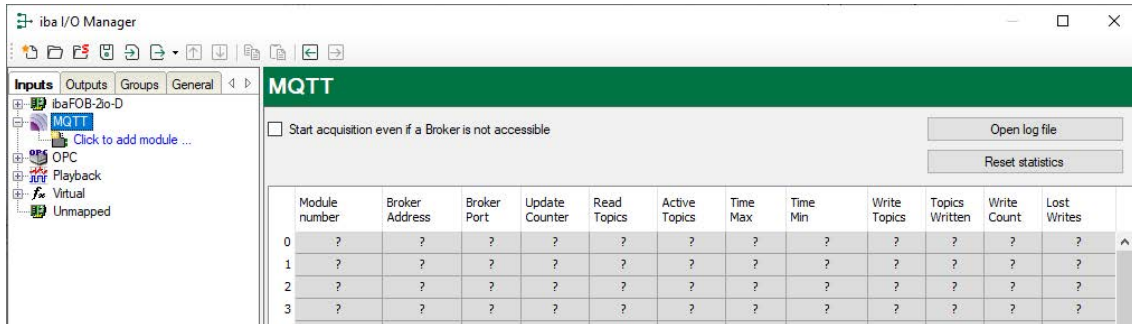
It is recommended carrying out the TCP/IP communication on a separate network segment to exclude a mutual influence by other network components.

3.3 Configuration and engineering ibaPDA

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

3.3.1 Interface settings

The interface itself has the following functions and configuration options:



Start acquisition even if a broker is not accessible

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

Connection table

The table shows the configured connections. Each table row corresponds to a MQTT module, respectively a connection to a broker.

<Open log file>

If connections to MQTT brokers have been established, all connection-specific actions are logged in a text file. Using this button, you can open and check this file. In the file system on the hard disk, you will find the log files in the program path of the ibaPDA server (...\\Programs\\ iba\\ ibaPDA\\Server\\Log\\). The file name of the current log file is `MqttLog.txt`, the name of the archived log files is `MqttLog_yyyy_mm_dd_hh_mm_ss.txt`.

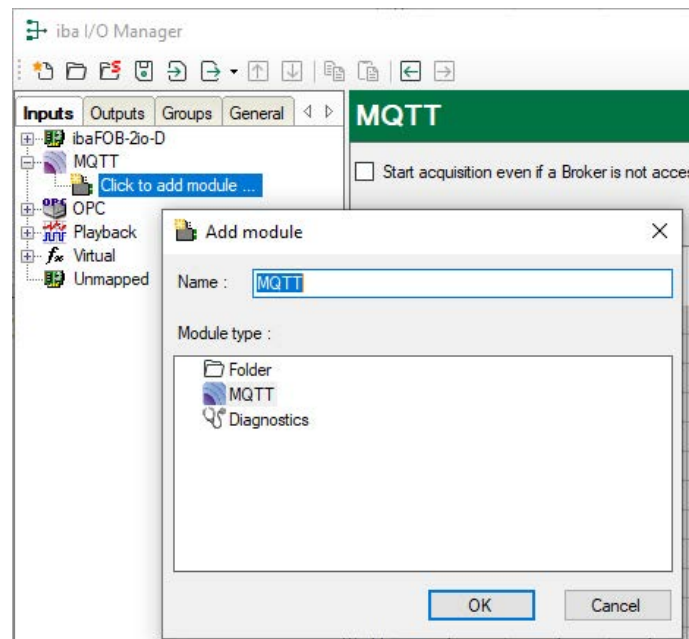
<Reset statistics>

To reset the counters for all connections, simply click on the <Reset statistics> button.

3.3.2 Adding a module

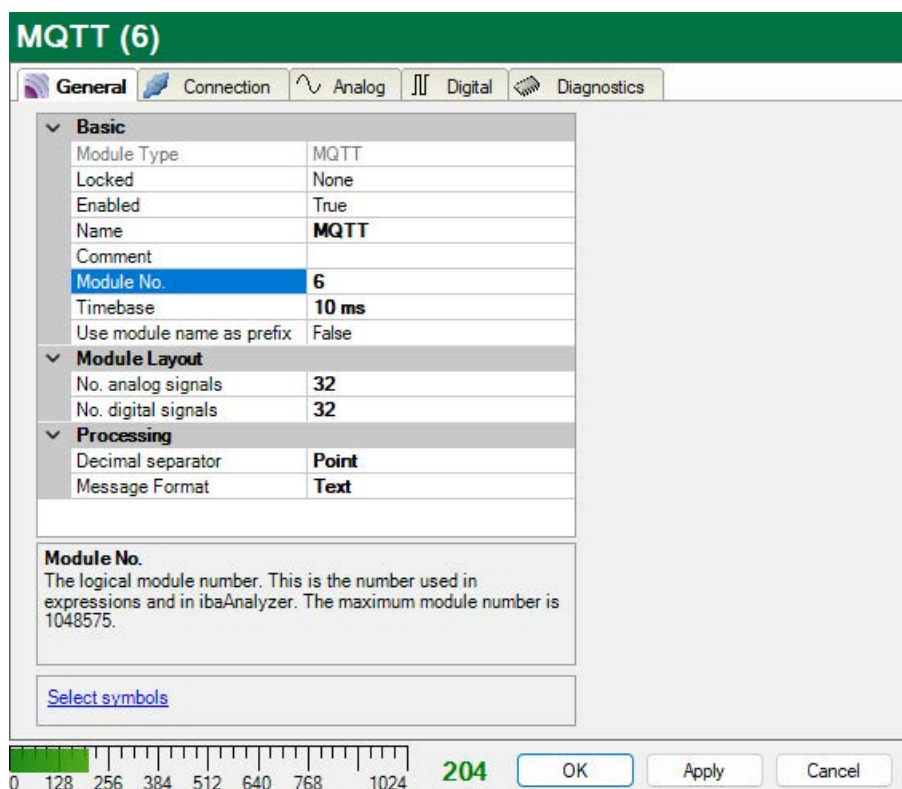
One MQTT module is required for each connection to a broker. You can also add one or more diagnostic modules. For more information on the diagnostic modules, refer to chapter [↗ Diagnostic modules](#), page 24.

Add a module by clicking below the interface. Select the MQTT module type and click <OK>.



3.3.3 General module settings

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



Basic settings

Module Type (information only)

Indicates the type of the current module.

Locked

You can lock a module to avoid unintentional or unauthorized changing of the module settings.

Enabled

Enable the module to record signals.

Name

You can enter a name for the module here.

Comment

You can enter a comment or description of the module here. This will be displayed as a tooltip in the signal tree.

Module No.

This internal reference number of the module determines the order of the modules in the signal tree of *ibaPDA* client and *ibaAnalyzer*.

Timebase

All signals of the module are sampled on this timebase.

Use module name as prefix

This option puts the module name in front of the signal names.

Module Layout**No. analog/digital signals**

Defines the number of analog/digital signals in this module. The default value is 32 for each. The maximum value is 1000.

Processing**Decimal point**

You can configure which character is used for the decimal point (point or comma).

Message format

You can select if values in the MQTT broker are stored in text format or as binary values.

Link to topic browser**Select symbols**

This link opens the topic browser. For further information, see [↗ Selecting topics via the topic browser](#), page 16.

3.3.4 Connection settings

In the *Connection* tab, configure the relevant settings to establish a connection.

Connection

Broker address

Enter the address of the broker here. The address can be an IP address, hostname or URL.

Port

Port to use for the connection. The standard MQTT port is 1883, or 8883 when using SSL.

Client ID

When connecting to a broker, each client must choose a unique name that is used only once for this broker. The ID is automatically filled in with ibaPDAMQTT followed by the module number.

Base Topic Path

This path is added in front of each registered topic (optional setting). The topic configuration can be made easier if all topics reside within a common path.

Note: Do not use a leading forward slash "/" in a topic, because this would add an empty path entry to the topic. Two separators next to each other ("/") are also not allowed.

Persistent session

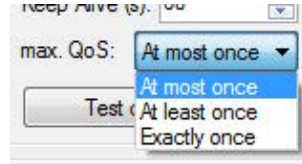
If the client is disconnected, it receives the last values sent to the broker when reconnecting with the same Client ID as before. For this, the QoS must be set to "Exactly once" or "At least once" not only for ibaPDA, but also for the sender who delivers the values to the broker.

Keep Alive

The time for sending a *Keep Alive* telegram to the broker to make sure the connection still is on-line.

max. 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)

<Test Connection>

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

Security options

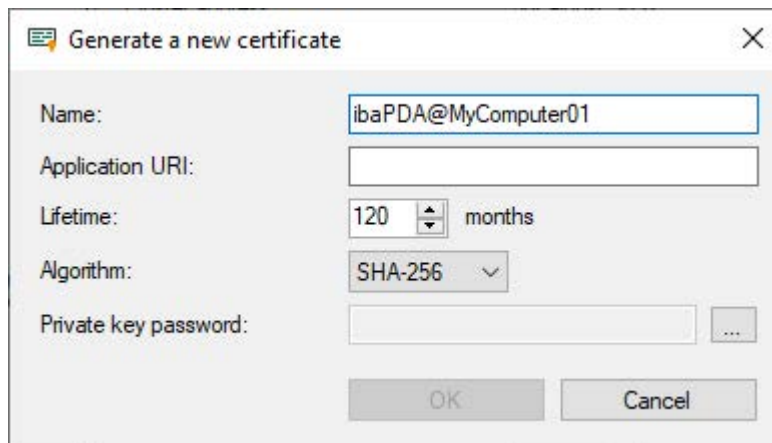
Authentication mode

The following are available:

- *None*: without authentication
- *Login*: Enter the 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.
- *Certificate*: The certificates that can be used are available for selection in the drop-down menu.

In addition more options can be selected:

- *No certificate*: no certificate is used. However, this usually causes the validation to fail.
- *Manage certificates*: opens the central certificate store, where certificates can be managed centrally in *ibaPDA*
- *Create new certificate*: *ibaPDA* creates a new self-signed certificate. Enter the necessary settings in the following dialog, see below. When the certificate is successfully generated, the new certificate is selected.



Enter a name for the certificate. You can change the default name.

Entering an Application URI is optional. Set the lifetime and select the algorithm. Available for selection are SHA-256 and SHA-1.

To enter the password for the private key, click the <...> button. For security reasons, you must enter the password twice in the following dialog. A password must be assigned, otherwise an error message appears. The password can be assigned arbitrarily, it does not have to meet any other requirements.

Use SSL connection

Use SSL instead of plain TCP connection. This option has to be configured accordingly on the broker, otherwise the connection will fail.

Last Will

Enable Last Will

If enabled, the absence of this client is announced to other clients, if the connection breaks without regular disconnect.

Topic

Enter a path to the topic used for the last will message. The base topic path is not applied to this topic.

Message

Enter a message which should be sent in the text field.

Retain

If enabled, 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. The settings are identical to the settings above.

Connect Message

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

3.3.5 Signal configuration

On the *Analog* and *Digital* tab, you can define the topic configuration. Settings for unit, gain, offset and data type refer only to the *Analog* tab.

General Connection Analog Digital Diagnostics								
	Name	Unit	Gain	Offset	Topic	Offset in Message	DataT...	Active
0			1	0		0	STRING	<input type="checkbox"/>
1			1	0		0	STRING	<input type="checkbox"/>
2			1	0		0	STRING	<input type="checkbox"/>
3			1	0		0	STRING	<input type="checkbox"/>
4			1	0		0	STRING	<input type="checkbox"/>
5			1	0		0	STRING	<input type="checkbox"/>
6			1	0		0	STRING	<input type="checkbox"/>
7			1	0		0	STRING	<input type="checkbox"/>
8			1	0		0	STRING	<input type="checkbox"/>

General Connection Analog Digital Diagnostics				
	Name	Topic	Offset in Message	Active
0			0	<input type="checkbox"/>
1			0	<input type="checkbox"/>
2			0	<input type="checkbox"/>
3			0	<input type="checkbox"/>
4			0	<input type="checkbox"/>
5			0	<input type="checkbox"/>
6			0	<input type="checkbox"/>
7			0	<input type="checkbox"/>
8			0	<input type="checkbox"/>

Name

You can enter a signal name and additionally two comments, when clicking on the symbol .

Unit

You can enter a physical unit.

Gain / Offset

The settings in the columns *Gain* and *Offset* are useful to scale normalized values to physical values.

Topic

The column *Topic* contains the data path on the broker. If a *base topic path* is set in the configuration, all topics configured here are found inside that path. A path for the topic is then used as a sub path under the base path.

Note: Do not use a leading forward slash "/" in a topic, because this would add an empty path entry to the topic. Two separators next to each other ("/") are also not allowed.

When you click on a cell in the *Topic* column, the <...> button appears. Click this button to open the topic browser. In the topic browser, you can select topics and add them to the signal table. See also [Selecting topics via the topic browser](#), page 16.

Offset in Message

The *Offset in Message* defines the byte offset inside the received message, to skip a fixed number of bytes or characters before decoding the desired value.

Data Type

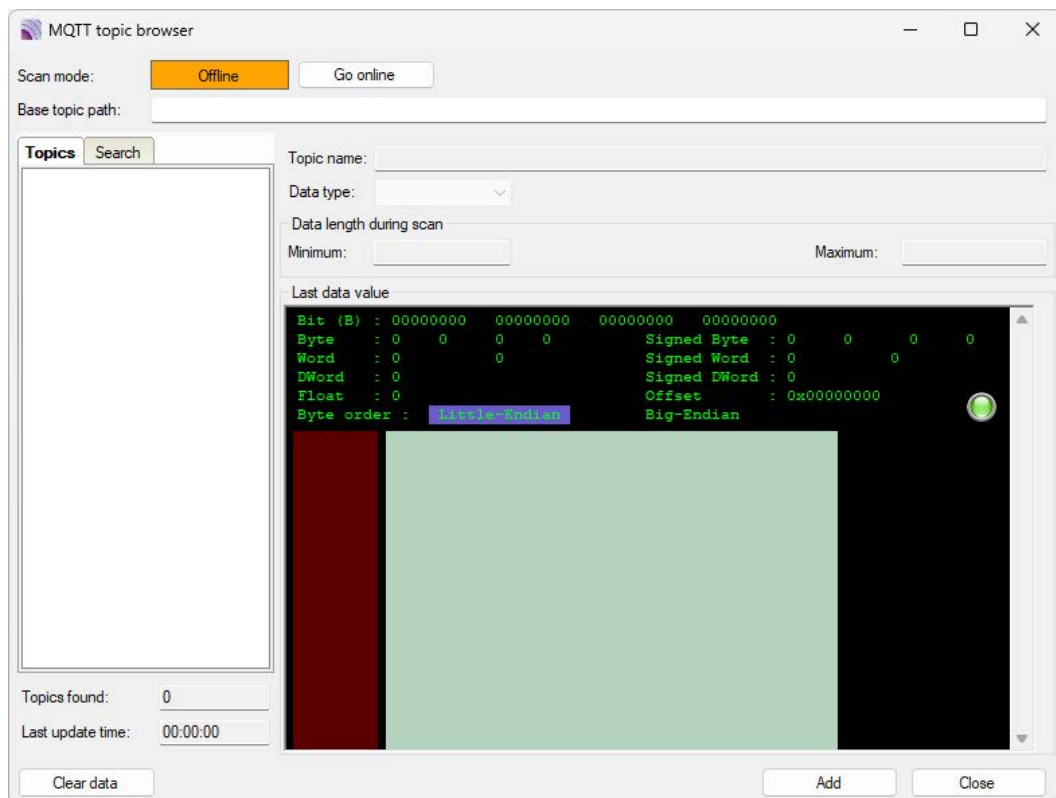
The *Data Type* sets how the value is stored in *ibaPDA*. Direct text to value conversion is supported directly in the MQTT module, using the configured decimal point character for interpreting float values.

Active

Enable/disable the signal for acquisition in *ibaPDA*.

3.3.5.1 Selecting topics via the topic browser

The MQTT topic browser can be used to select active topics. The topic browser can be opened with a click on the *Select symbols* link on the *General* tab.



Scan mode

Displays the connection status (online or offline).

A click on <Go online> establishes a connection to the broker, when the status is offline. Then the topics sent to the broker are read.

When the status is online, the connection is disconnected with a click on <Go offline>.

Base topic path

The topic base path entered in the *Connection* tab is displayed here.

Topics

In the *Topics* window, the topics that are published to the base topic path are displayed.

Select a topic in the tree view and the corresponding information is displayed on the right side of the dialog. You can transfer it to the signal table with <Add>.

Search

There is a search function available for finding and selecting topics when you click on the *Search* tab.

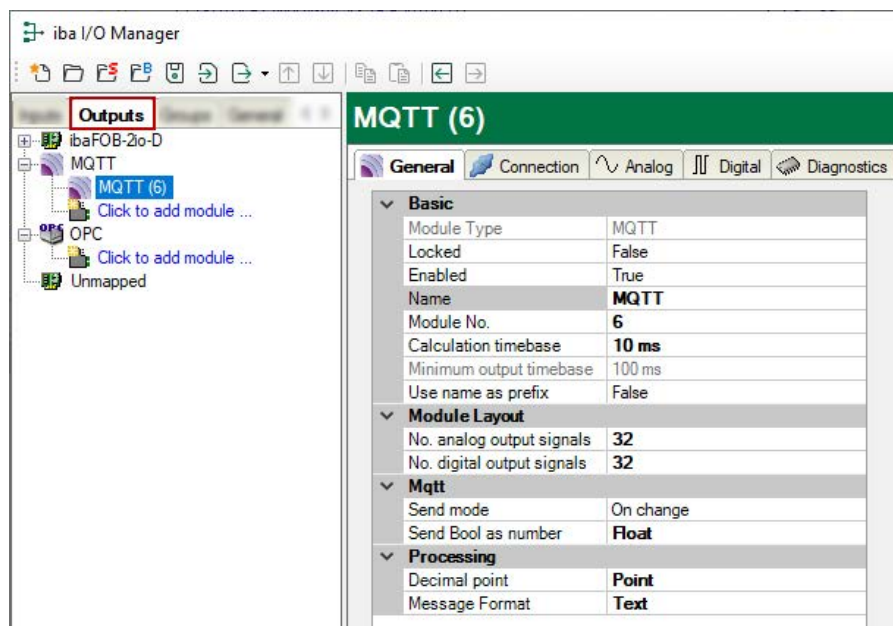
1. Enter a complete topic name or parts of it in the entry line.
 2. Then press <RETURN> or <Search>.
- The result list shows all topics containing the search item entered.

3.3.6 Outputs

The output module is not an autonomous module, but an extension of the MQTT module. With the output module you can write data from *ibaPDA* to an MQTT Broker.

You can access the configuration of the module by selecting the *Outputs* tab in the module tree. You do not have to add it separately. The module will be available as soon as the MQTT module has been added.

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



For more information on the module settings see ↗ *General module settings*, page 10.

Module-specific settings

Calculation timebase

Timebase (in ms) used for the calculation of the output values.

Technically, the calculation timebase is identical to the timebase of the input module. This means a change in the calculation timebase also changes the module timebase of the input side and vice versa!

The calculation timebase is not the same as the output timebase, with which the values are output!

Minimum output timebase

Timebase with which the outputs can be updated as quickly as possible.

The value is acquired automatically by the system based on the current I/O configuration and is only displayed here. The output timebase results from the smallest common multiple of all module timebases or is at least 50 ms.

Number of analog/digital output signals

Define the number of configurable analog and digital signals in the signal tables. The default value is 32 for each. The maximum value is 1000. The signal tables are adjusted accordingly.

Mqtt**Send mode**

Determines when new data is written to the controller:

- **Cyclic:** Data is written cyclically at the set update time.
- **On change:** Data is written each time the signal data is changed.
- **On trigger:** Data is written with every rising edge of the trigger signal.

All signals of a module are always written, regardless of the write mode.

Trigger signal

This field only appears when the "on trigger" send mode is selected. Select here a digital signal. A rising edge on this digital signal writes the signal values taken at the time of the rising edge.

Send Bool as number

Select if boolean values in text mode are written as float values (1 and 0) or boolean values (TRUE and FALSE).

Processing**Decimal point**

Configure which character is used for the decimal point.

Message format

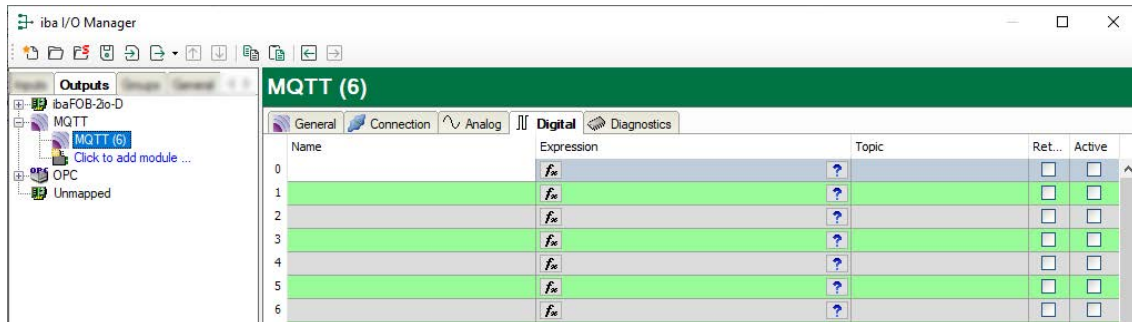
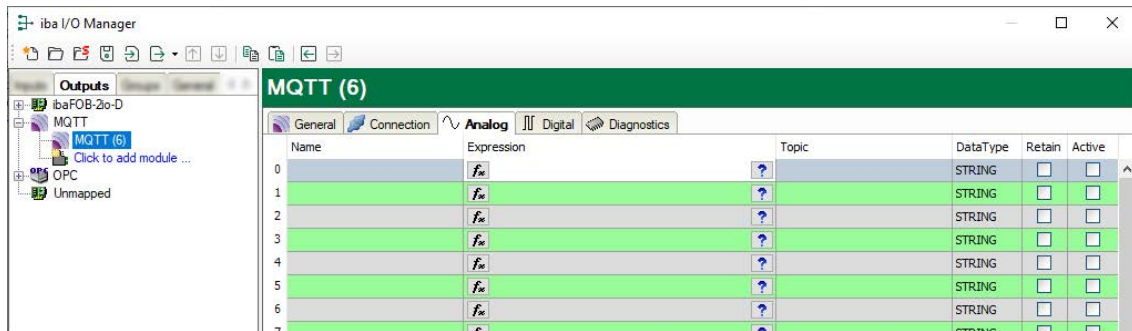
Select if values in MQTT Broker are stored in text format or as binary values.

Output signals

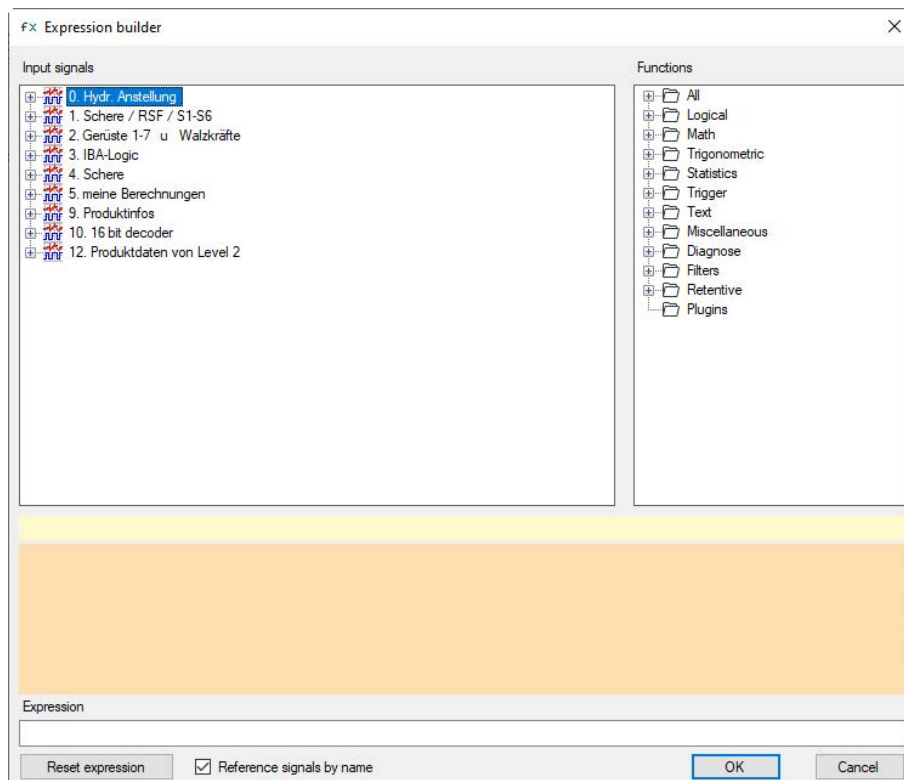
The analog and digital signals to be output are configured in the expression builder. The expression builder can be opened by the <fx> button from each signal row.

In the *Analog* and *Digital* tab, enter a name for the signal and a data path to the broker. Analog signals support the data types SINT, BYTE, INT, WORD, DINT, DWORD, FLOAT, DOUBLE, STRING.

If you enable the *Retain* option, the signals are stored on the broker until they are overwritten.



Expression builder:



Other documentation

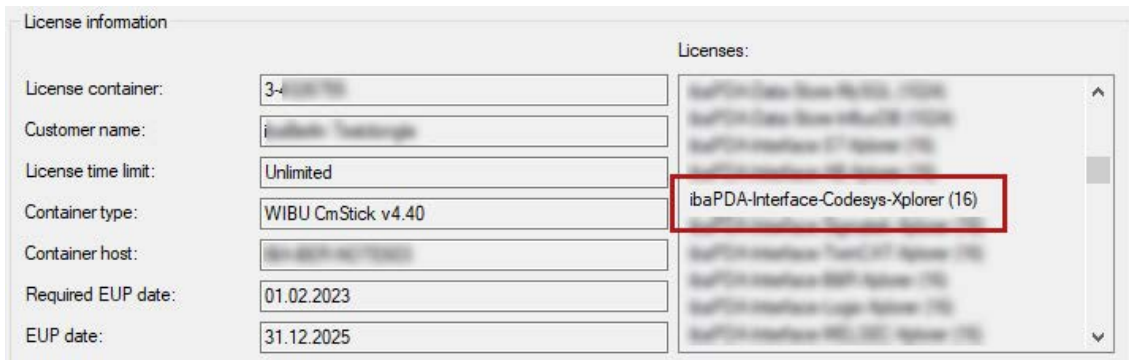
For more information about how to use the expression builder, see the *ibaPDA* manual.

4 Diagnostics

4.1 License

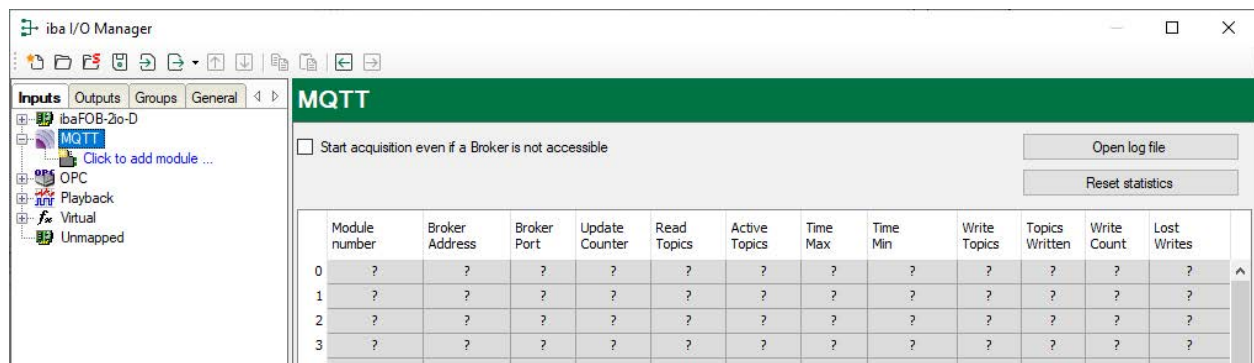
If the interface is not displayed in the signal tree, you can either check in *ibaPDA* in the I/O Manager under *General – Settings* or in the *ibaPDA* service status application whether your license for the interface *ibaPDA-Interface-MQTT* has been properly recognized. The number of licensed connections is shown in brackets.

The figure below shows the license for the *Codesys Xplorer* interface as an example.



4.2 Connection table

The MQTT interface shows a table of the configured connections. Each line in the table corresponds to a configured MQTT module and respectively a connection to a broker.



The columns in the table and their meaning:

- **Module Number:** Module number of the MQTT module
- **Broker Address:** Shows the address of the connected broker (configured in the *Connection* tab).
- **Broker Port:** Shows the port for the communication with the broker (configured in the *Connection* tab).
- **Update Counter:** Counts the updates sent from the broker. An update contains data for only one topic.
- **Defined topics:** Shows the number of topics defined in the module configuration.

- **Active topics:** Shows the number of topics which received an update since acquisition start or a click on <Reset statistics>.
- **Time Max. / Time Min.:** Show the maximum and minimum time (in ms) between two updates per topic.

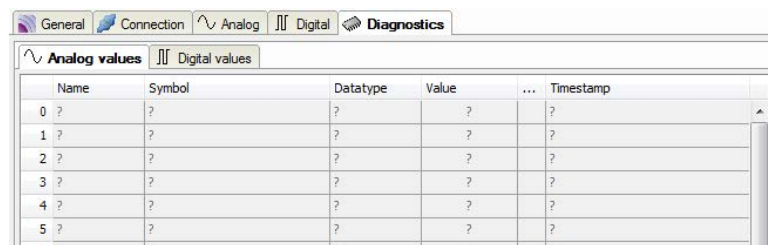
Additional information is provided by the background color of the table rows:

Color	Meaning
Green	The connection is OK and the data is read.
Red	The connection has failed.
Grey	No connection configured.

Table 1: Meaning of background colors

4.3 Diagnostics in the I/O Manager

The current values of the requested topics can be monitored in the *Diagnostics* tab of the MQTT module in the I/O manager. Analog values and digital values are displayed in separate tabs. The last column shows the timestamp of the last received message for each topic.



General Connection Analog Digital Diagnostics					
Analog values Digital values					
Name	Symbol	Datatype	Value	...	Timestamp
0 ?	?	?	?	?	?
1 ?	?	?	?	?	?
2 ?	?	?	?	?	?
3 ?	?	?	?	?	?
4 ?	?	?	?	?	?
5 ?	?	?	?	?	?

4.4 Connection diagnostics with PING

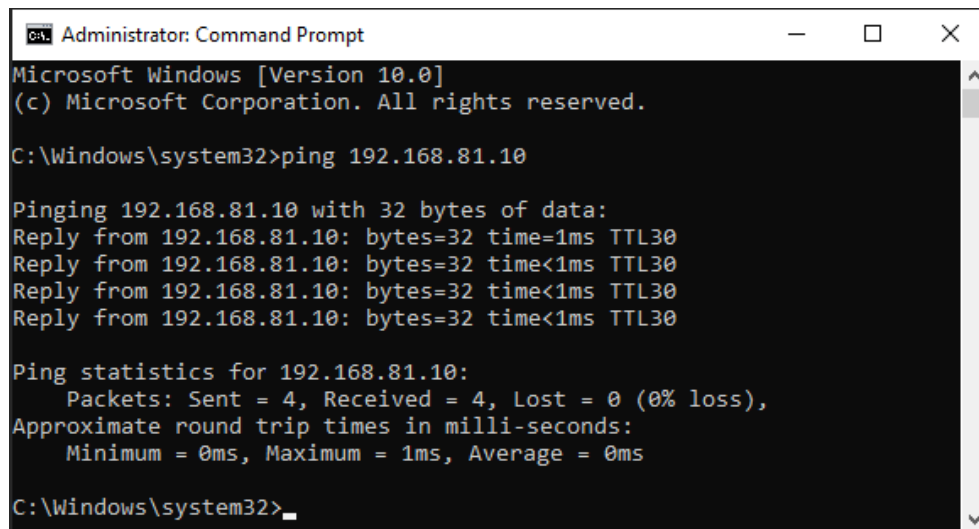
PING is a system command with which you can check if a certain communication partner can be reached in an IP network.

1. Open a Windows command prompt.



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

→ With an existing connection you receive several replies.



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

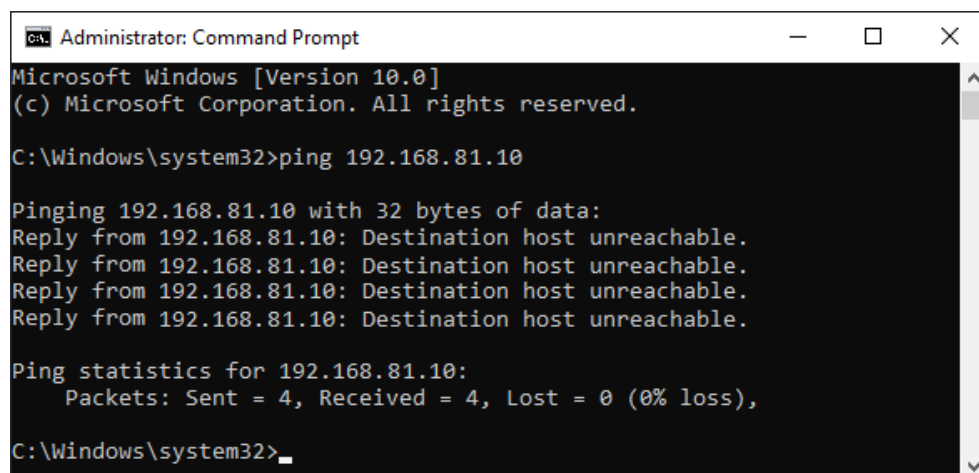
C:\Windows\system32>ping 192.168.81.10

Pinging 192.168.81.10 with 32 bytes of data:
Reply from 192.168.81.10: bytes=32 time=1ms TTL=30
Reply from 192.168.81.10: bytes=32 time<1ms TTL=30
Reply from 192.168.81.10: bytes=32 time<1ms TTL=30
Reply from 192.168.81.10: bytes=32 time<1ms TTL=30

Ping statistics for 192.168.81.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\Windows\system32>
```

→ With no existing connection you receive error messages.



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

C:\Windows\system32>ping 192.168.81.10

Pinging 192.168.81.10 with 32 bytes of data:
Reply from 192.168.81.10: Destination host unreachable.
Reply from 192.168.81.10: Destination host unreachable.
Reply from 192.168.81.10: Destination host unreachable.
Reply from 192.168.81.10: Destination host unreachable.

Ping statistics for 192.168.81.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

C:\Windows\system32>
```

4.5 Diagnostic modules

Diagnostic modules are available for most Ethernet based interfaces and Xplorer interfaces. Using a diagnostic module, information from the diagnostic displays (e.g. diagnostic tabs and connection tables of an interface) can be acquired as signals.

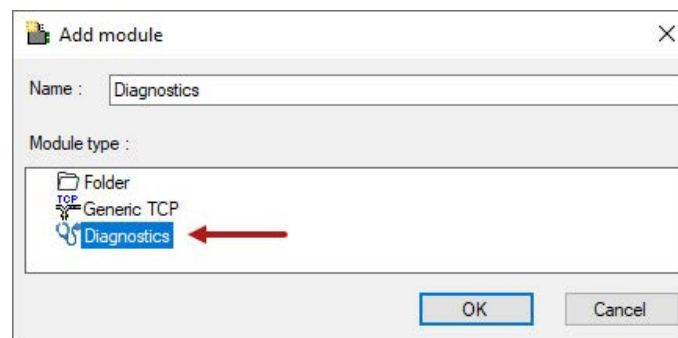
A diagnostic module is always assigned to a data acquisition module of the same interface and supplies its connection information. By using a diagnostic module, you can record and analyze the diagnostic information continuously in the *ibaPDA* system.

Diagnostic modules do not consume any license connections because they do not establish their own connection but refer to another module.

Example for the use of diagnostic modules:

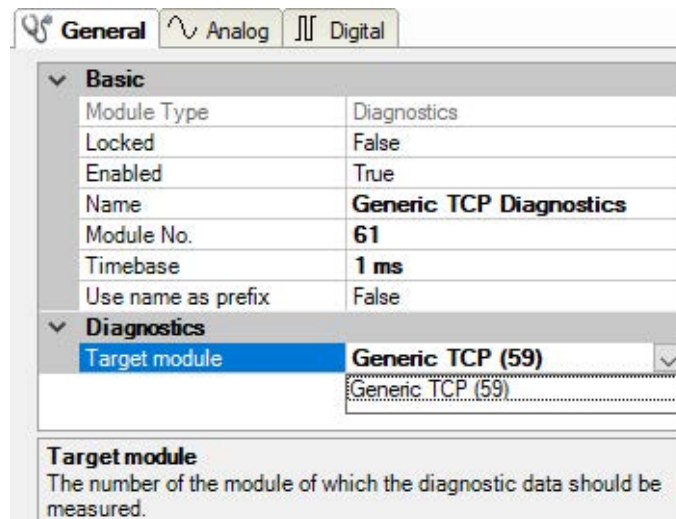
- A notification can be generated, whenever the error counter of a communication connection exceeds a certain value or the connection gets lost.
- In case of a disturbance, the current response times in the telegram traffic may be documented in an incident report.
- The connection status can be visualized in *ibaQPanel*.
- You can forward diagnostic information via the SNMP server integrated in *ibaPDA* or via OPC DA/UA server to superordinate monitoring systems like network management tools.

In case the diagnostic module is available for an interface, a "Diagnostics" module type is shown in the *Add module* dialog (example: Generic TCP).



Module settings diagnostic module

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



Basic	
Module Type	Diagnostics
Locked	False
Enabled	True
Name	Generic TCP Diagnostics
Module No.	61
Timebase	1 ms
Use name as prefix	False

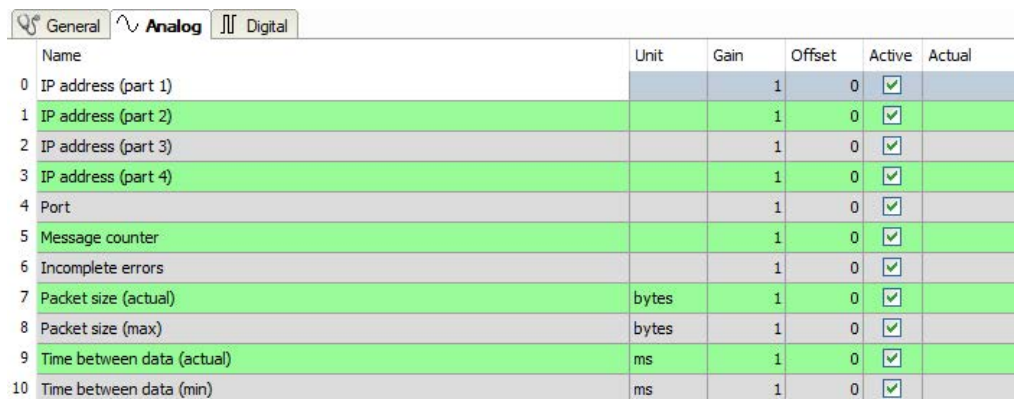
Diagnostics	
Target module	Generic TCP (59)

Target module
The number of the module of which the diagnostic data should be measured.

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

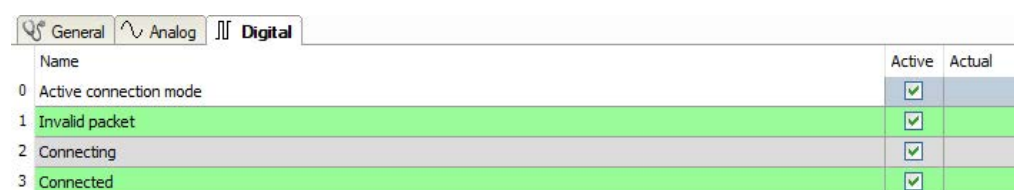
There is only one setting which is specific for the diagnostic module: the target module.

By selecting the target module, you assign the diagnostic module to the module on which you want to acquire information about the connection. You can select the supported modules of this interface in the drop-down list of the setting. You can assign exactly one data acquisition module to each diagnostic module. When having selected a module, the available diagnostic signals are immediately added to the *Analog* and *Digital* tabs. It depends on the type of interface, which signals exactly are added. The following example lists the analog values of a diagnostic module for a Generic TCP module.



Name	Unit	Gain	Offset	Active	Actual
0 IP address (part 1)		1	0	<input checked="" type="checkbox"/>	
1 IP address (part 2)		1	0	<input checked="" type="checkbox"/>	
2 IP address (part 3)		1	0	<input checked="" type="checkbox"/>	
3 IP address (part 4)		1	0	<input checked="" type="checkbox"/>	
4 Port		1	0	<input checked="" type="checkbox"/>	
5 Message counter		1	0	<input checked="" type="checkbox"/>	
6 Incomplete errors		1	0	<input checked="" type="checkbox"/>	
7 Packet size (actual)	bytes	1	0	<input checked="" type="checkbox"/>	
8 Packet size (max)	bytes	1	0	<input checked="" type="checkbox"/>	
9 Time between data (actual)	ms	1	0	<input checked="" type="checkbox"/>	
10 Time between data (min)	ms	1	0	<input checked="" type="checkbox"/>	

For example, the IP (v4) address of a Generic TCP module (see fig. above) will always be split into 4 parts derived from the dot-decimal notation, for better reading. Also other values are being determined, as there are port number, counters for telegrams and errors, data sizes and telegram cycle times. The following example lists the digital values of a diagnostic module for a Generic TCP module.



Name	Active	Actual
0 Active connection mode	<input checked="" type="checkbox"/>	
1 Invalid packet	<input checked="" type="checkbox"/>	
2 Connecting	<input checked="" type="checkbox"/>	
3 Connected	<input checked="" type="checkbox"/>	

Diagnostic signals

Depending on the interface type, the following signals are available:

Signal name	Description
Active	Only relevant for redundant connections. Active means that the connection is used to measure data, i.e. for redundant standby connections the value is 0. For normal/non-redundant connections, the value is always 1.
Buffer file size (actual/avg/max)	Size of the file for buffering statements
Buffer memory size (actual/avg/max)	Size of the memory used by buffered statements
Buffered statements	Number of unprocessed statements in the buffer
Buffered statements lost	Number of buffered but unprocessed and lost statements
Connected	Connection is established
Connected (in)	A valid data connection for the reception (in) is available
Connected (out)	A valid data connection for sending (out) is available
Connecting	Connection being established
Connection attempts (in)	Number of attempts to establish the receive connection (in)
Connection attempts (out)	Number of attempts to establish the send connection (out)
Connection ID O->T	ID of the connection for output data (from the target system to <i>ibaPDA</i>). Corresponds to the assembly instance number
Connection ID T->O	ID of the connection for input data (from <i>ibaPDA</i> to target system). Corresponds to the assembly instance number
Connection phase (in)	Status of the ibaNNet-E data connection for reception (in)
Connection phase (out)	Status of the ibaNNet-E data connection for sending (out)
Connections established (in)	Number of currently valid data connections for reception (in)
Connections established (out)	Number of currently valid data connections for sending (out)
Data length	Length of the data message in bytes
Data length O->T	Size of the output message in byte
Data length T->O	Size of the input message in byte
Destination IP address (part 1-4) O->T	4 octets of the IP address of the target system Output data (from target system to <i>ibaPDA</i>)
Destination IP address (part 1-4) T->O	4 octets of the IP address of the target system Input data (from <i>ibaPDA</i> to target system)
Disconnects (in)	Number of currently interrupted data connections for reception (in)
Disconnects (out)	Number of currently interrupted data connections for sending (out)
Error counter	Communication error counter
Exchange ID	ID of the data exchange
Incomplete errors	Number of incomplete messages

Signal name	Description
Incorrect message type	Number of received messages with wrong message type
Input data length	Length of data messages with input signals in bytes (<i>ibaPDA</i> receives)
Invalid data points	Number of received data points with missing configuration
Invalid packet	Invalid data packet detected
IP address (part 1-4)	4 octets of the IP address of the target system
Keepalive counter	Number of Keepalive messages received by the OPC UA Server
Lost images	Number of lost images (in) that were not received even after a retransmission
Lost Profiles	Number of incomplete/incorrect profiles
Message counter	Number of messages received
Messages per cycle	Number of messages in the cycle of the update time
Messages received since configuration	Number of received data telegrams (in) since start of acquisition
Messages received since connection start	Number of received data telegrams (in) since the start of the last connection setup. Reset with each connection loss.
Messages sent since configuration	Number of sent data telegrams (out) since start of acquisition
Messages sent since connection start	Number of sent data telegrams (out) since the start of the last connection setup. Reset with each connection loss.
Multicast join error	Number of multicast login errors
Number of request commands	Counter for request messages from <i>ibaPDA</i> to the PLC/CPU
Output data length	Length of the data messages with output signals in bytes (<i>ibaPDA</i> sends)
Packet size (actual)	Size of the currently received message
Packet size (max)	Size of the largest received message
Ping time (actual)	Response time for a ping telegram
Port	Port number for communication
Producer ID (part 1-4)	Producer ID as 4-byte unsigned integer
Profile Count	Number of completely recorded profiles
Read counter	Number of read accesses/data requests
Receive counter	Number of messages received
Response time (actual/average/max/min)	<p>Response time is the time between measured value request from <i>ibaPDA</i> and response from the PLC or reception of the data.</p> <p>Actual: current value</p> <p>Average/max/min: static values of the update time since the last start of the acquisition or reset of the counters.</p>
Retransmission requests	Number of data messages requested again if lost or delayed

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

5 Support and contact

Support

Phone: +49 911 97282-14
Email: support@iba-ag.com

Note



If you need support for software products, please state the number of the license container. For hardware products, please have the serial number of the device ready.

Contact

Headquarters

iba AG
Koenigswarterstrasse 44
90762 Fuerth
Germany

Phone: +49 911 97282-0
Email: iba@iba-ag.com

Mailing address

iba AG
Postbox 1828
D-90708 Fuerth, Germany

Delivery address

iba AG
Gebhardtstrasse 10
90762 Fuerth, Germany

Regional and Worldwide

For contact data of your regional iba office or representative please refer to our web site:

www.iba-ag.com