



ibaPDA-Interface-MQTT

MQTT Communication Interface

Manual
Issue 1.2

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

Manufacturer

iba AG
Koenigswarterstr. 44
90762 Fuerth
Germany

Contacts

Main office +49 911 97282-0
Fax +49 911 97282-33
Support +49 911 97282-14
Engineering +49 911 97282-13
E-mail iba@iba-ag.com
Web www.iba-ag.com

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 2020, 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 - Chapter / Page	Author	Version SW
1.2	09-2020	Miscellaneous	st	7.0.0

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.

Content

1	About this Manual	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.....	10
3.3.3	General module settings.....	10
3.3.4	Connection settings	12
3.4	Signal configuration	14
4	Diagnostics.....	16
4.1	License	16
4.2	Connection table	16
4.3	Diagnostics in the I/O Manager	17
4.4	Connection diagnostics with PING.....	18
4.5	Diagnostic modules	19
5	Support and contact.....	22

1 About this Manual

This document describes the function and application of the software interface

ibaPDA-Interface-MQTT

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>
File names, paths	"Filename", "Path" Example: "Test.doc"

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* v7.0.0 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

Table 1: Available MQTT interface licenses, as of ibaPDA-V7.0.0

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.

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.

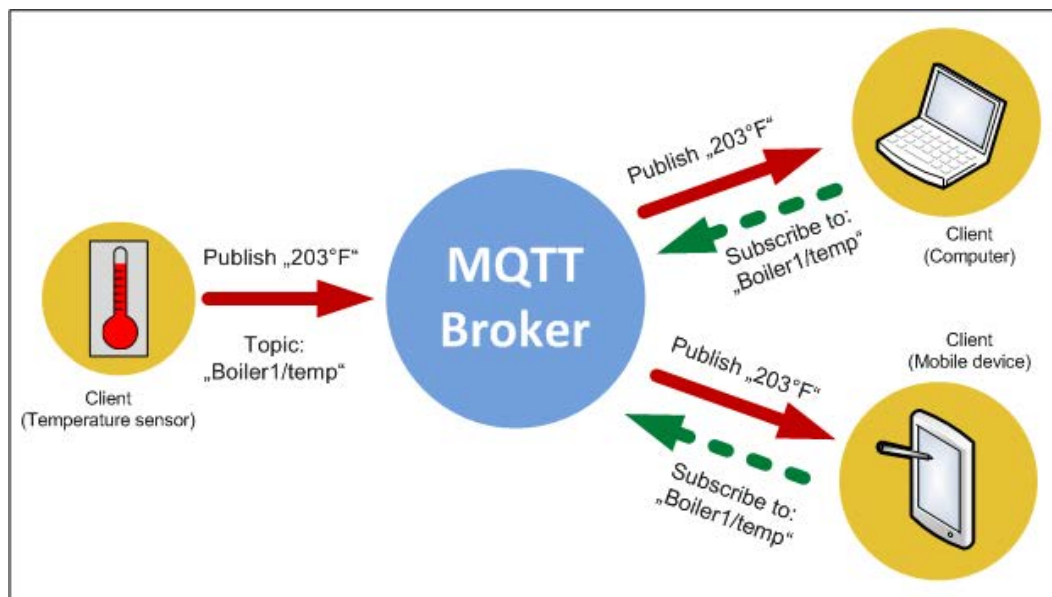


Fig. 1: Publish/Subscribe architecture of MQTT

Concerning *ibaPDA*, this means that *ibaPDA* is positioned as an MQTT client.

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.

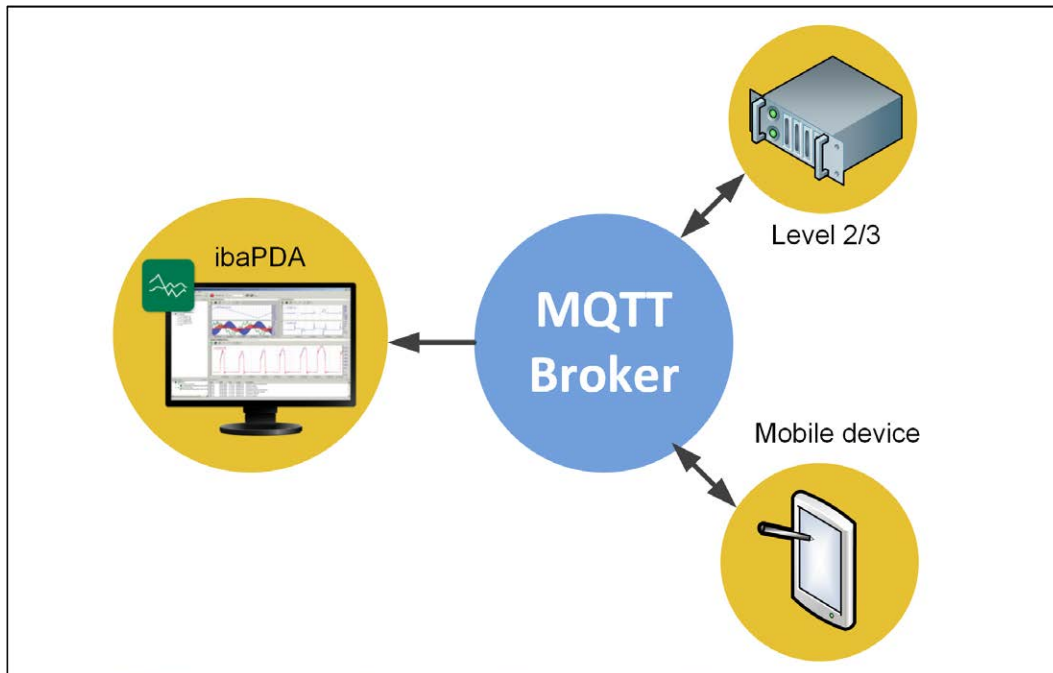


Fig. 2: ibaPDA in the MQTT architecture

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

Open the I/O manager, e.g. from the toolbar .

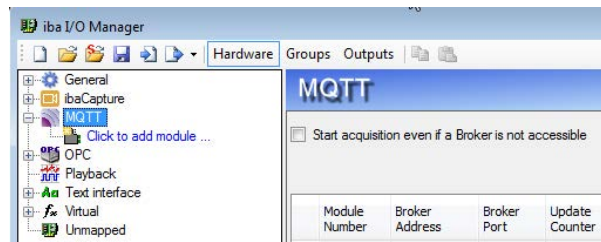


Fig. 3: MQTT interface in the I/O manager

If all system requirements are met (see above), the MQTT interface will be displayed in the signal tree.

3.3.1 Interface settings

The interface itself has the following functions and configuration options:

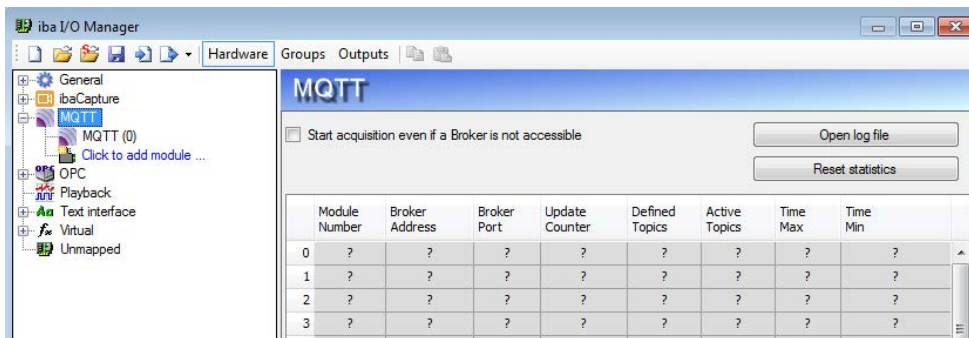


Fig. 4: MQTT interface settings

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 19](#).

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

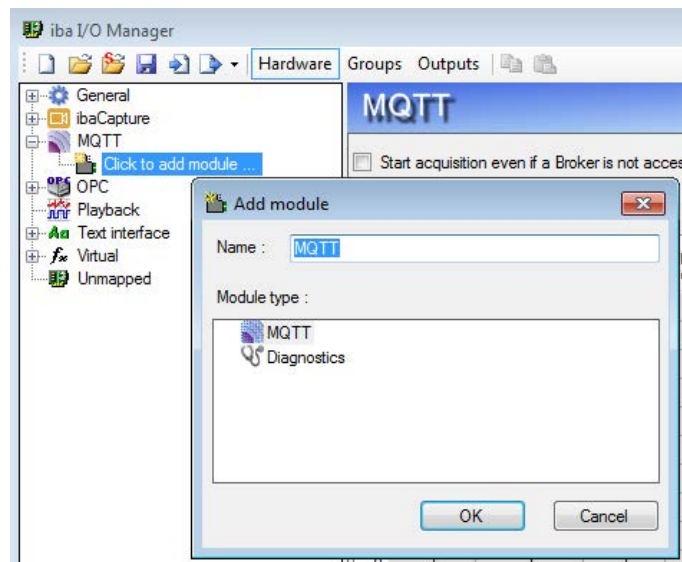


Fig. 5: Adding MQTT module

3.3.3 General module settings

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

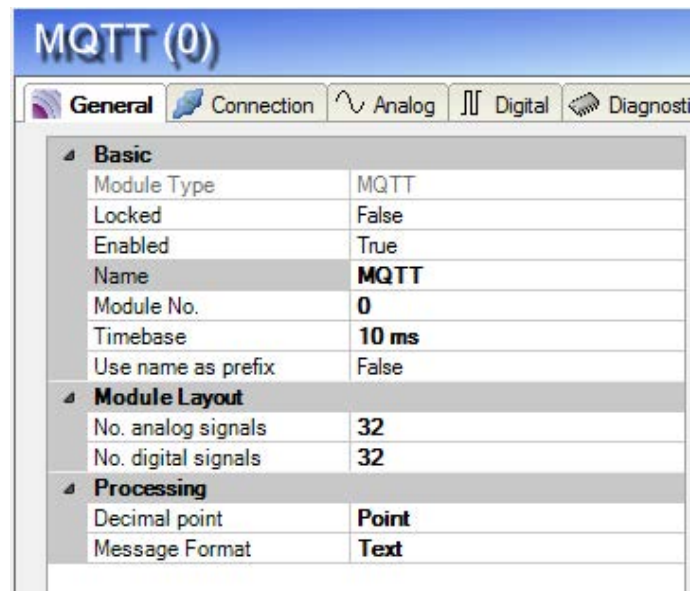


Fig. 6: General settings of a MQTT module

Basic settings

Module Type (information only)

Indicates the type of the current module.

Locked

A module can be locked to avoid unintentional or unauthorized changing of the module settings.

Enabled

Disabled modules are excluded from signal acquisition.

Name

The plain text name should be entered here as the module designation.

Module No.

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

Timebase

All signals of the module will be sampled on this time base.

Use name as prefix

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.

3.3.4 Connection settings

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

The screenshot shows the 'Connection' tab of the MQTT interface. It contains several sections:

- Connection:** Broker address: localhost, Port: 1883, Client ID: ibaPDAMQTT0, Base Topic Path: (empty). There are checkboxes for 'Persistent session' and 'Keep Alive (s): 60'. A dropdown for 'max. QoS' is set to 'At most once'. A 'Test connection' button is present.
- Security options:** 'Use authentication' is unchecked, and 'Use SSL connection' is checked. There are input fields for 'User name' and 'Password'.
- Last Will:** 'Enable Last Will' is unchecked. There are input fields for 'Topic' and 'Message'. A 'Retain' checkbox is unchecked, and a 'QoS' dropdown is set to 'At most once'.
- Connect Message:** 'Enable Connect Message' is unchecked. There are input fields for 'Topic', 'Connect Message', and 'Disconnect Message'. A 'Retain' checkbox is unchecked, and a 'QoS' dropdown is set to 'At most once'.

Fig. 7: MQTT connection settings

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

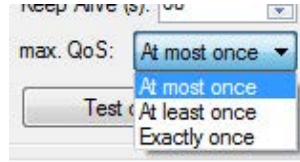
If the client is disconnected, it receives the last values sent to the broker when reconnecting with the same Client ID as before. For this, the QoS must be set to "Exactly once" or "At least once" not only for ibaPDA, but also for the sender who delivers the values to the broker.

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

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 connection

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

The Windows Certificate Store is used to store the certificate.

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.

3.4 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.

	Name	Unit	Gain	Offset	Topic	Offset in Message	Data...	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"/>

Fig. 8: Analog tab

	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"/>

Fig. 9: Digital tab

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.

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*.

4 Diagnostics

4.1 License

If the "MQTT" interface is not displayed in the signal tree, you can either check in *ibaPDA* under *General - Settings- License info* in the I/O manager or in the *ibaPDA* service status application, whether your license "Interface MQTT" has been properly recognised.



Fig. 10: License display in the ibaPDA I/O manager

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.

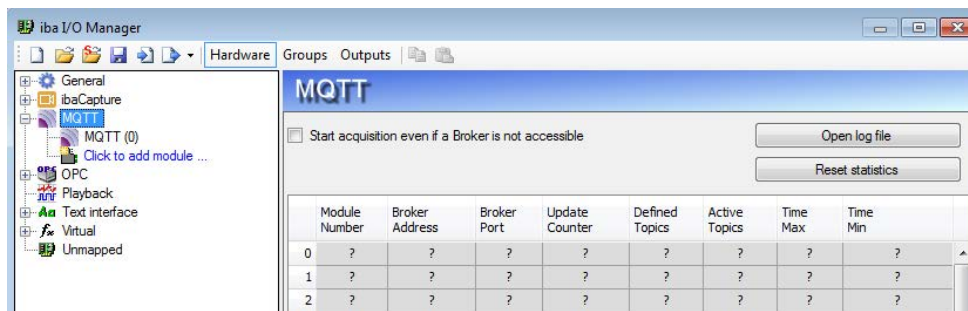


Fig. 11: MQTT connection table

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 2: 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.

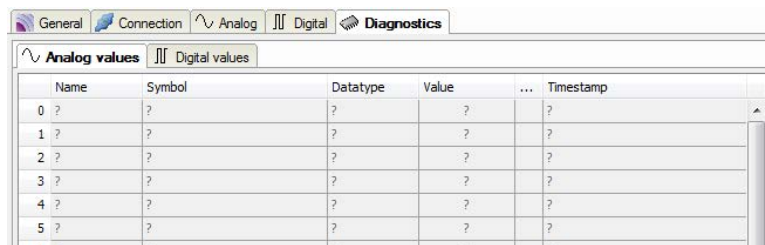


Fig. 12: Diagnostics tab of an MQTT module

4.4 Connection diagnostics with PING

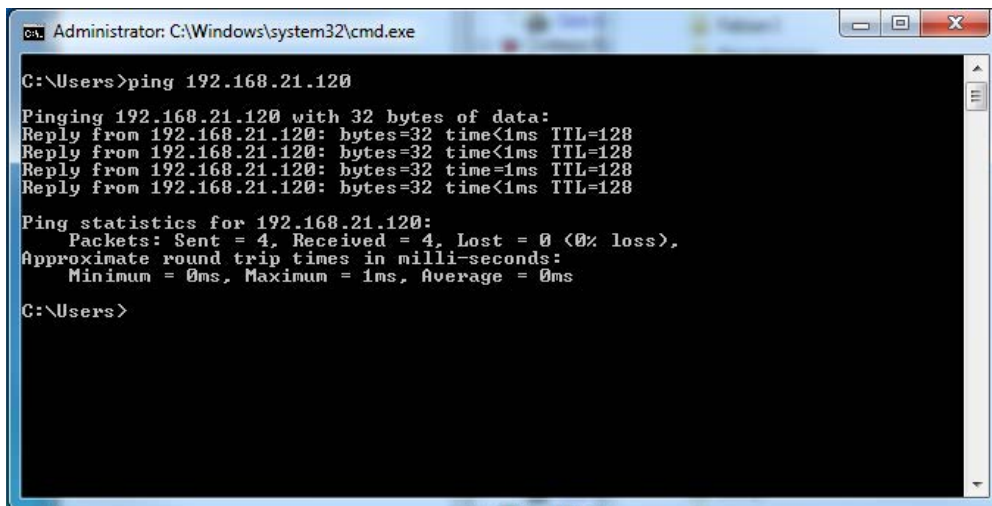
PING is a system command with which you can check if a certain communication partner can be reached in an IP network.

Open a Windows command prompt.



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

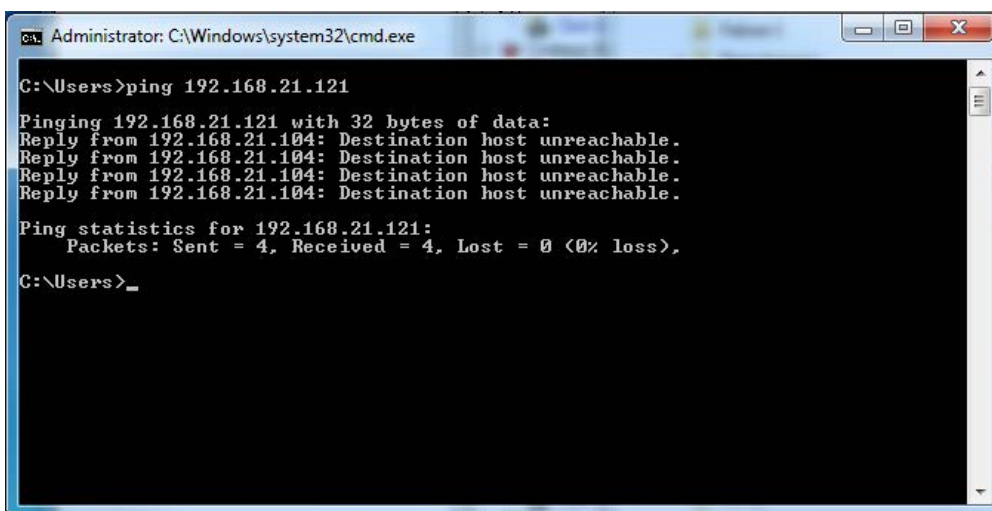
With an existing connection you receive several replies.

A screenshot of a Windows command prompt window titled 'Administrator: C:\Windows\system32\cmd.exe'. The prompt shows the command 'C:\Users>ping 192.168.21.120' and its output: 'Pinging 192.168.21.120 with 32 bytes of data: Reply from 192.168.21.120: bytes=32 time<1ms TTL=128 Reply from 192.168.21.120: bytes=32 time<1ms TTL=128 Reply from 192.168.21.120: bytes=32 time=1ms TTL=128 Reply from 192.168.21.120: bytes=32 time<1ms TTL=128 Ping statistics for 192.168.21.120: Packets: Sent = 4, Received = 4, Lost = 0 (0% loss), Approximate round trip times in milli-seconds: Minimum = 0ms, Maximum = 1ms, Average = 0ms C:\Users>'.

```
Administrator: C:\Windows\system32\cmd.exe
C:\Users>ping 192.168.21.120
Pinging 192.168.21.120 with 32 bytes of data:
Reply from 192.168.21.120: bytes=32 time<1ms TTL=128
Reply from 192.168.21.120: bytes=32 time<1ms TTL=128
Reply from 192.168.21.120: bytes=32 time=1ms TTL=128
Reply from 192.168.21.120: bytes=32 time<1ms TTL=128
Ping statistics for 192.168.21.120:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
C:\Users>
```

Fig. 13: PING successful

With no existing connection you receive error messages.

A screenshot of a Windows command prompt window titled 'Administrator: C:\Windows\system32\cmd.exe'. The prompt shows the command 'C:\Users>ping 192.168.21.121' and its output: 'Pinging 192.168.21.121 with 32 bytes of data: Reply from 192.168.21.104: Destination host unreachable. Reply from 192.168.21.104: Destination host unreachable. Reply from 192.168.21.104: Destination host unreachable. Reply from 192.168.21.104: Destination host unreachable. Ping statistics for 192.168.21.121: Packets: Sent = 4, Received = 4, Lost = 0 (0% loss), C:\Users>_'.

```
Administrator: C:\Windows\system32\cmd.exe
C:\Users>ping 192.168.21.121
Pinging 192.168.21.121 with 32 bytes of data:
Reply from 192.168.21.104: Destination host unreachable.
Reply from 192.168.21.104: Destination host unreachable.
Reply from 192.168.21.104: Destination host unreachable.
Reply from 192.168.21.104: Destination host unreachable.
Ping statistics for 192.168.21.121:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
C:\Users>_
```

Fig. 14: PING unsuccessful

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, since 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.

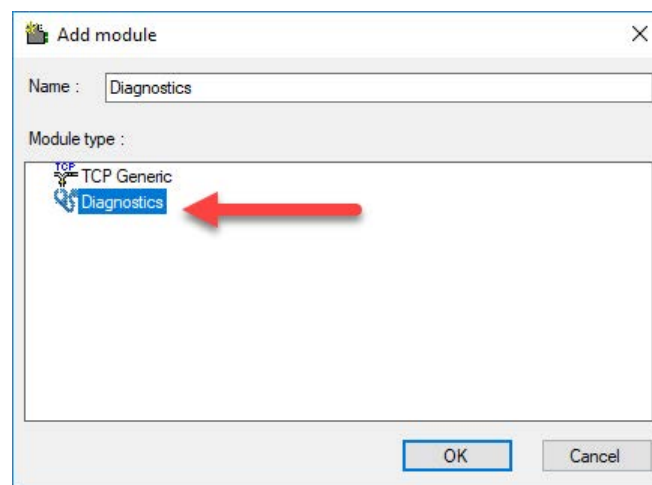


Fig. 15: Add diagnostic module, example Generic TCP

Module settings diagnostic module

For a diagnostic module, you can make the following settings:

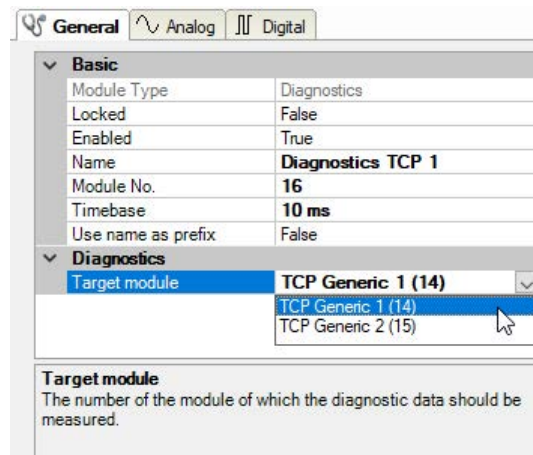


Fig. 16: Module settings diagnostic module, example TCP Generic

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.

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"/>	
11 Time between data (max)	ms	1	0	<input checked="" type="checkbox"/>	

Fig. 17: Example: Analog values of a diagnostic module for a TCP Generic module

For example, the IP (v4-) address of a TCP Generic 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.

General Analog Digital		
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"/>	

Fig. 18: Example: Digital values of a diagnostic module for a TCP Generic module

5 Support and contact

Support

Phone: +49 911 97282-14
Fax: +49 911 97282-33
Email: support@iba-ag.com

Note



If you require support, indicate the serial number (iba-S/N) of the product or the license number.

Contact

Headquarters

iba AG
Koenigswarterstrasse 44
90762 Fuerth
Germany

Phone: +49 911 97282-0
Fax: +49 911 97282-33
Email: iba@iba-ag.com

Mailing address

iba AG
Postbox 1828
D-90708 Fuerth, Germany

Delivery address

iba AG
Gebhardtstrasse 10
90762 Fuerth, Germany

Regional and Worldwide

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

www.iba-ag.com.