



ibaPDA-Interface-SISTEAM-TCP

Data Interface to Systeem

Manual
Issue 3.0

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

Manufacturer

iba AG
Koenigswarterstrasse 44
90762 Fuerth
Germany

Contacts

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

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 2023, 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
3.0	10-2023	New version ibaPDA v8	RM/IP	8.4.0

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	5
1.1	Target group and previous knowledge	5
1.2	Notations	5
1.3	Used symbols.....	6
2	System Requirements.....	7
3	Data interface TCP to Sisteam.....	8
3.1	General information	8
3.2	ibaPDA driver decoding for SISTEAM over TCP/IP	11
3.2.1	SISTEAM BUS header	11
3.2.2	SISTEAM BUS data	12
3.3	TCP/IP data message layout.....	13
3.3.1	SISTEAM Integer	13
3.3.2	SISTEAM Real	14
3.3.3	SISTEAM Generic	14
4	Configuration and engineering ibaPDA	15
4.1	General settings.....	15
4.2	General interface settings.....	16
4.3	Adding a module.....	16
4.4	General Module Settings	17
4.4.1	Module Type Generic	18
4.4.2	Module Type Integer	18
4.4.3	Module Type Real	18
4.5	Signal configuration	19
5	Diagnostics.....	21
5.1	License	21
5.2	Visibility of the interface.....	21
5.3	Log files.....	22
5.4	Connection diagnostics with PING.....	23
5.5	Connection Check.....	24
6	Appendix	26

- 6.1 Troubleshooting..... 26
 - 6.1.1 TCP performance problems caused by Delayed Acknowledge..... 26
 - 6.1.2 TCP data corruption resulting from the Nagle’s Algorithm..... 28
- 6.2 References 29
- 7 Support and contact..... 30**

1 About this documentation

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

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 is aimed at qualified professionals who are familiar with handling electrical and electronic modules as well as communication and measurement technology. A person is regarded as professional if he/she is capable of assessing safety and recognizing possible consequences and risks on the basis of his/her specialist training, knowledge and experience and knowledge of the standard regulations.

This documentation in particular addresses persons, who are concerned with the configuration, test, commissioning or maintenance of Programmable Logic Controllers of the supported products. For the handling *ibaPDA-Interface-SISTEAM-TCP* the following basic knowledge is required and/or useful:

- Windows operating system
- Basic knowledge of *ibaPDA*
- Knowledge of configuration and operation of the relevant measuring device/system

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	Filename , Path Example: Test.docx

1.3 Used symbols

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

Danger!



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

- Observe the specified measures.

Warning!



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

- Observe the specified measures.

Caution!



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

- Observe the specified measures

Note



A note specifies special requirements or actions to be observed.

Tip



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

Other documentation



Reference to additional documentation or further reading.

2 System Requirements

The following system requirements are necessary for the use of the data interface TCP/IP to Sisteam:

- *ibaPDA* v8.0.0 or higher
- License for *ibaPDA-Interface-SISTEAM-TCP*
- Network connection 10/100 Mbits

For more requirements on the PC hardware used and the supported operating systems, see the *ibaPDA* Documentation.

Note



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

System Restrictions

- The maximum length of the SISTEAM TCP/IP message is limited to 1024 bytes. (When using the TCP/IP SISTEAM Generic module)
- Connections to *ibaPDA* can only be made with as destination port 8738 (hex: 0x2222)
- For different ways of handling the TCP/IP-Acknowledge see [↗ TCP performance problems caused by Delayed Acknowledge](#), page 26 (all *ibaPDA* versions).

The following controllers apply:

Any system that is capable of sending messages over the SISTEAM protocol as a client over TCP/IP.

Licenses

Order no.	Product Designation	Description
31.001055	ibaPDA-Interface-Sisteam-TCP	Extension license for an <i>ibaPDA</i> system by one TCP/IP interface Number of connections: 64

Table 1: Available interface license

3 Data interface TCP to Sisteam

3.1 General information

The SISTEAM TCP interface can be used by any controller capable of sending messages using the SISTEAM protocol especially the SISTEAM controllers of INGELECTRIC S.A.

Excerpt from the SISTEAM Communication manual (See References 1.5 doc 4.):

SISTEAM BUS is an Ethernet-type local network, with a bus topology, which implements the TCP/IP protocol.

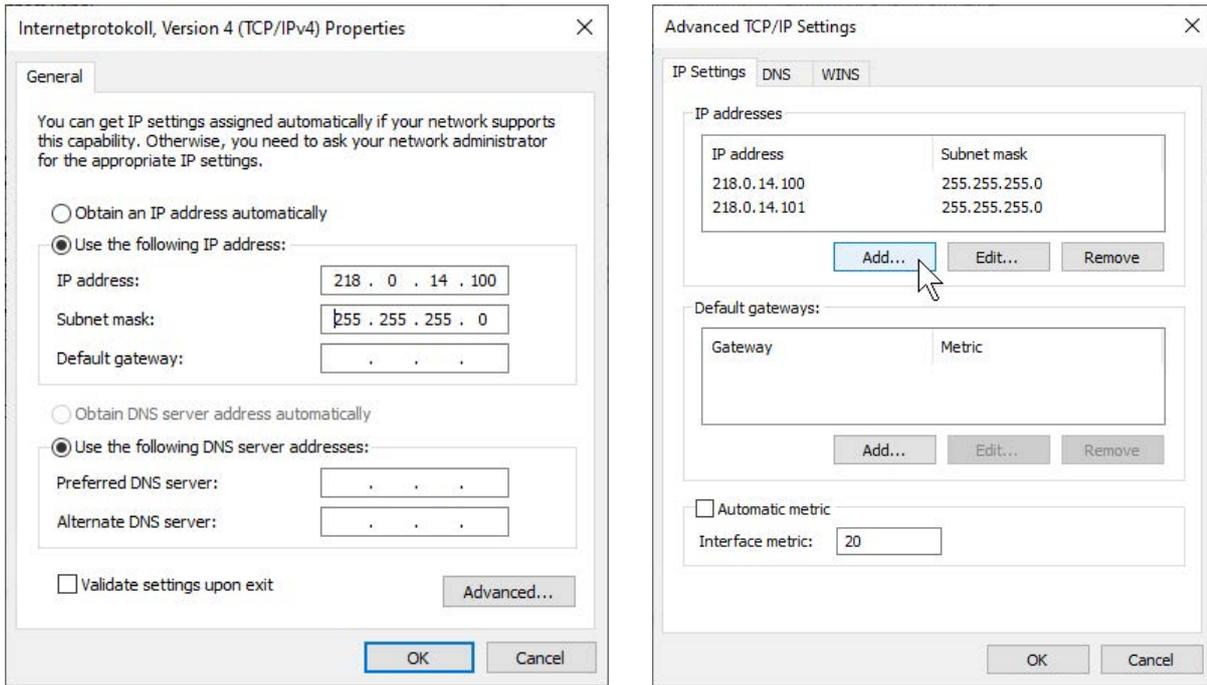
SISTEAM BUS has a bus topology, which allows serial data transfer up to 10 Mbits/second, using the CSMA/CD access method, in accordance with the IEEE 802.3 standard.

The Transmission Control Protocol (TCP) is one of the core protocols of the Internet protocol suite. IP handles lower-level transmissions from computer to computer as a message makes its way across the internet, TCP operates at a higher level, concerned only with the two end systems. TCP provides reliable, ordered delivery of a stream of bytes from a program on one computer to another program on another computer. TCP is explained in chapter 6 of RFC1180 and in RFC768 (See References).

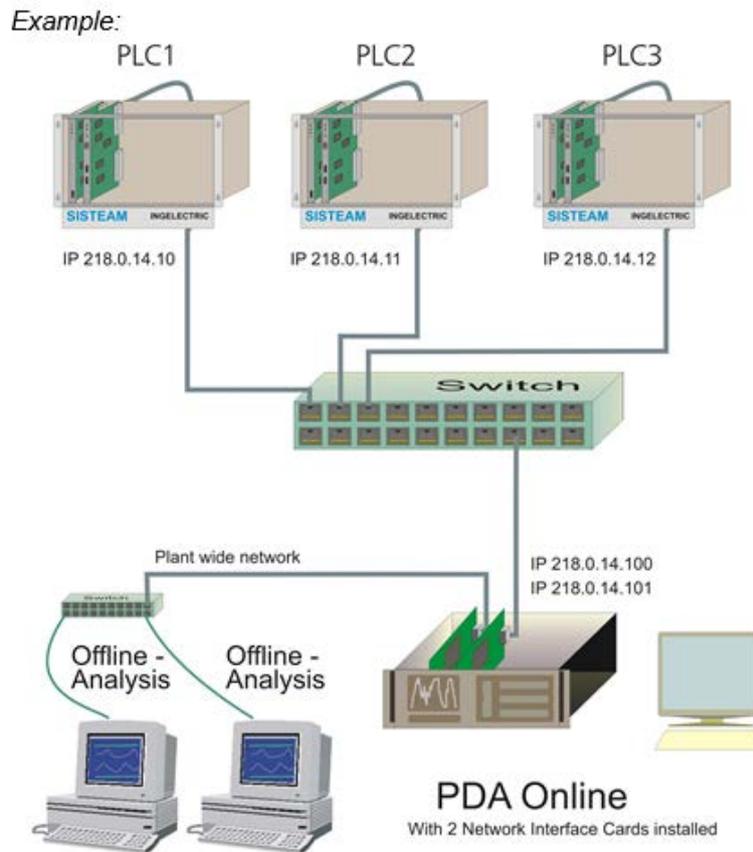
SISTEAM over TCP/IP is limited to send data packets to target port 8738 (hex: 0x2222). This means that the *ibaPDA* driver only can receive messages on port 8738. The source port will be randomly generated as expected by any TCP/IP application enabled to establish multiple connections (links). With the TCP/IP driver, *ibaPDA* works as a connection server and the controllers work as clients. This means that the *ibaPDA* station is listening for sender information and connection requests on port 8738.

It is important to note that the IP addresses should be unique but most controllers can establish multiple connections to the same destination IP address. The SISTEAM Ethernet module however is not able to make multiple connections with the same destination IP address. Hence the network interface card (NIC) in the *ibaPDA* system has to be configured for multiple IP addresses on the same NIC. From the point of view of each individual PLC, a different IP address has to be specified for each connection which corresponds to a data module in *ibaPDA*.

This method involves assigning multiple IP addresses to a single NIC (Multihoming). This is accomplished through the Windows control panel. On the control panel go to *Network and Internet - Network and Sharing Center* - and select *Change adapter settings*. Make a right mouseclick on the network adapter with the Sisteam connection and select *Properties* from the context menu. After opening it, select the Protocols tab, highlight TCP/IP and then select properties. Next, click the *Advanced* button and a dialog box will appear as shown below. Here, all the additional IP addresses can be added to any NIC in your system.



Each data link between the controller and *ibaPDA* should be considered as an *ibaPDA* module which needs a unique Module Number ID (0..63). Once a link is established, always the same module number must be send over the link. This means there is a one to one relationship between the TCP/IP link and the module number in the *ibaPDA* system.



The figure above gives an overview of a configuration where 3 PLCs are connected with one *ibaPDA* system. The 3 PLCs act as client and each PLC has its own Ethernet module connected to a network switch.

Each PLC establishes 2 connections with the *ibaPDA* system. The *ibaPDA* system has 2 IP-addresses on the same network interface card.

The following table gives an overview of the connections and the modules transferred to the *ibaPDA* system:

Conn.	Module Index	Module Type	Sender		Receiver
1	0	SISTEAM Real	PLC1 218.0.14.10	→	PDA 218.0.14.100
2	100	SISTEAM Integer	PLC1 218.0.14.10	→	PDA 218.0.14.101
3	1	SISTEAM Real	PLC2 218.0.14.11	→	PDA 218.0.14.100
4	200	SISTEAM Generic	PLC2 218.0.14.11	→	PDA 218.0.14.101
5	2	SISTEAM Real	PLC3 218.0.14.12	→	PDA 218.0.14.100
6	201	SISTEAM Generic	PLC3 218.0.14.12	→	PDA 218.0.14.101

3.2 ibaPDA driver decoding for SISTEAM over TCP/IP

SISTEAM over TCP/IP supports 3 types of modules:

SISTEAM Integer

The analog values are integer type and the digital values are bit type.

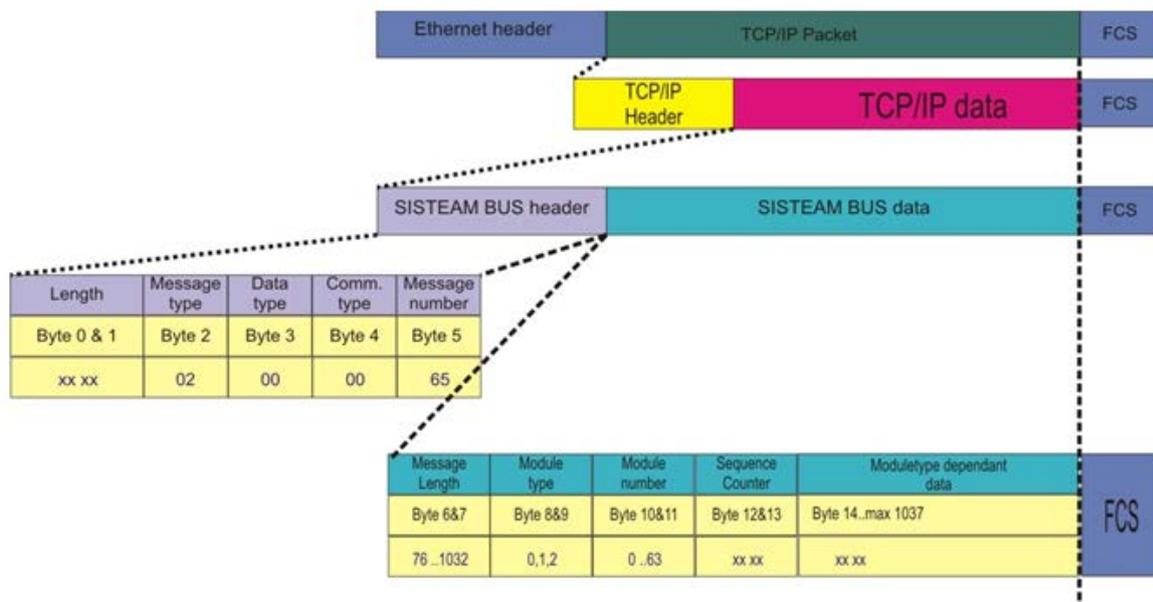
SISTEAM Real

The analog values are float type (IEEE Float format) and the digital values are bit type.

SISTEAM Generic

The values can be a mixture of the following types: SINT, BYTE, INT, WORD, DINT, DWORD, IEEE FLOAT, DOUBLE and STRING[32] with a maximum of 1024 bytes.

The SISTEAM packet structure is shown below and is encapsulated in the data part from the TCP/IP packet.



3.2.1 SISTEAM BUS header

Length

This word contains the number of bytes of the complete SISTEAM BUS message excluding this word.

Message type

This byte contains a static value of 02 which indicates the use of an information sending message.

Data type

This byte contains a non self defined value 0.

Communication type

This byte specifies the communication type used for messaging. Since a virtual circuit is used in this message type, the value 0 should be filled in.

Message number

The message number indicates the number of messages at the receiving node that interprets this message. The value is non self defined and set to 0x65.

3.2.2 SISTEAM BUS data

The data field contains a data structure depending on the module type used.

Message length (2 bytes)

The message length is the length of all data fields which are encapsulated in the SISTEAM data. This data field contains a header of 8 bytes and a number of bytes used for the analog and digital signals. The number of bytes used for the analog and digital signals depends on the used module type. The length of a SISTEAM Integer and a SISTEAM Real module are fixed but the length of a SISTEAM Generic module can change according to the customers needs.

Module type (2 bytes)

The SISTEAM interface in *ibaPDA* supports three kinds of modules. To specify which module is used, *ibaPDA* uses a specific number for each module. The following module types can be chosen:

SISTEAM Real	0
SISTEAM Integer	1
SISTEAM Generic	2

Module number (2 bytes)

The module number is a value between 0 and 63. This value needs to be unique for each module of the same module type. This value together with the Module type will be used in *ibaPDA* to determine the **module index**.

Sequence counter (2 bytes)

The sequence counter should increment by one each sending cycle and can contain a value between 0 and 65535. This counter gives *ibaPDA* the possibility to detect errors in the communication.

Module type dependant data

The data field contains the actual data that will be measured by *ibaPDA*. The type of data depends on the used SISTEAM module.

Each kind of module is restricted to the kind of values that can be sent via the specified module, except the generic module. In this module each data type can be used next to each other. In *ibaPDA* only the addresses and data types that are used need to be configured for each signal.

Example: SISTEAM Generic module analog signals configuration

Sisteam TCP Generic (0)								
General Analog Digital								
Name	Unit	Gain	Offset	Address	DataType	Active	Actual	
0		1	0	0x0	INT	<input checked="" type="checkbox"/>	0	
1		1	0	0x2	SINT	<input checked="" type="checkbox"/>	0	
2		1	0	0x4	BYTE	<input checked="" type="checkbox"/>	0	
3		1	0	0x6	INT	<input checked="" type="checkbox"/>	0	
4		1	0	0x8	WORD	<input checked="" type="checkbox"/>	0	
5		1	0	0xA	DINT	<input checked="" type="checkbox"/>	0	
6		1	0	0xC	DWORD	<input checked="" type="checkbox"/>	0	
7		1	0	0xE	STRING[32]	<input checked="" type="checkbox"/>	0	
8		1	0	0x10	DOUBLE	<input checked="" type="checkbox"/>	0	
9		1	0	0x12	INT	<input checked="" type="checkbox"/>	0	

3.3 TCP/IP data message layout

This chapter explains in details the message layout of the different module types.

The TCP/IP message layout covers the SISTEAM BUS data as explained in ↗ *SISTEAM BUS data*, page 12

3.3.1 SISTEAM Integer

32 integer values 32 digital values

rel. #	Bytes	C - Type	Description	Comment
00	2	Short int.	Message Length	= header (8 bytes) + dig (4 bytes) + 2 * number of integers (0 .. 32) = max 76 bytes
02	2	Short int.	Module Type	1: integer module
04	2	Short int	Module Number	00..63
06	2	Unsigned short	Sequence Counter	0..65535 Incremented every cycle by the sender
08	4	Unsigned long	32 bit digital values	
12	Max 64	Short int.	32 Analog Integer Values	Max of 32 analog integer values

3.3.2 SISTEAM Real

32 float values 32 digital values

rel. #	Bytes	C - Type	Description	Comment
00	2	Short int.	Message Length	= header (8 bytes) + dig (4 bytes) + 4 * number of reals (0 .. 32) = max 140 bytes
02	2	Short int.	Module Type	0: float module
04	2	Short int	Module Number	00..63
06	2	Unsigned short	Sequence Counter	0..65535 Incremented every cycle by the sender
08	4	Unsigned long	32 bit digital values	
12	Max 128	Float	32 Analog Integer Values	Max of 32 analog float values

3.3.3 SISTEAM Generic

Generic

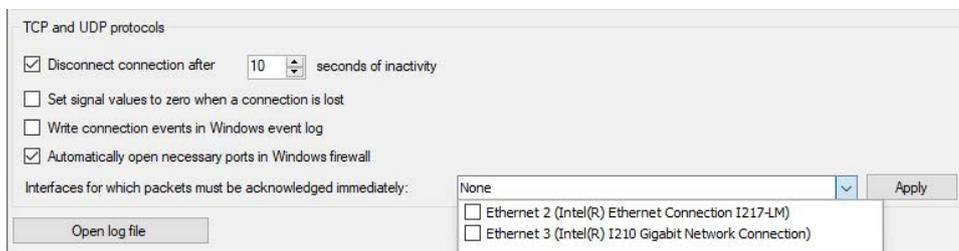
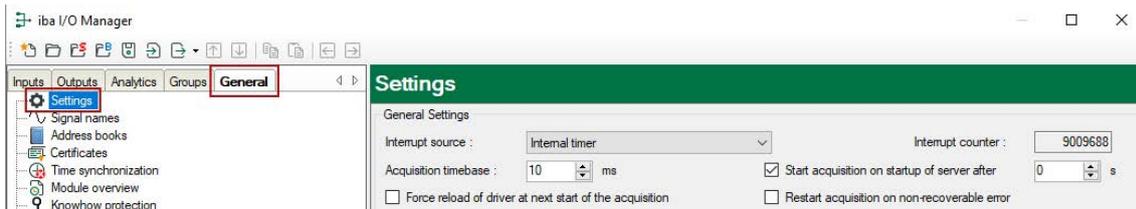
rel. #	Bytes	C - Type	Description	Comment
00	2	Short int.	Message Length	= header (8 bytes) + number of bytes used for analog and digital signals = max 1032 bytes
02	2	Short int.	Module Type	2 : generic
04	2	Short int	Module Number	00..63
06	2	Unsigned short	Sequence Counter	0..65535 Incremented every cycle by the sender
08	Max 1024	Mixture of: Byte, Int, Word, Double Int, Double Word, IEEE Float	Analog/Digital values	

4 Configuration and engineering ibaPDA

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

4.1 General settings

The "Alive timeout" is configured jointly for all TCP/IP and UDP protocols supported by *ibaPDA*.



Disconnect connection after ... seconds of inactivity

Behavior and timeout duration can be specified.

Set signal values to zero when a connection is lost

If this option is disabled, the value read last will be kept.

Write connection events in Windows event log

Current events are logged in Windows.

Automatically open necessary ports in Windows firewall

If this option is enabled, all ports required for the currently licensed interfaces are automatically opened in the firewall by the *ibaPDA* server service.

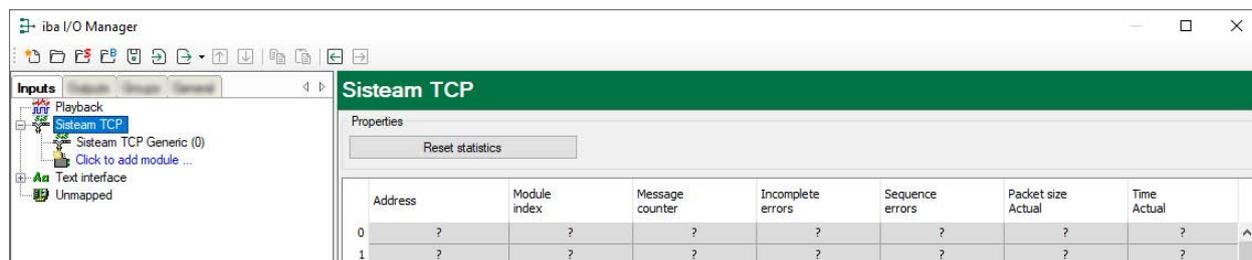
If this option is disabled, the required ports can be opened manually in the I/O Manager of the licensed interfaces via <Allow port through firewall>.

Interfaces for which packets must be acknowledged immediately

Selection of required interfaces.

4.2 General interface settings

The interface provides the following functions and configuration options.



<Reset statistics>

Click this button to reset the calculated times and error counters in the table to 0.

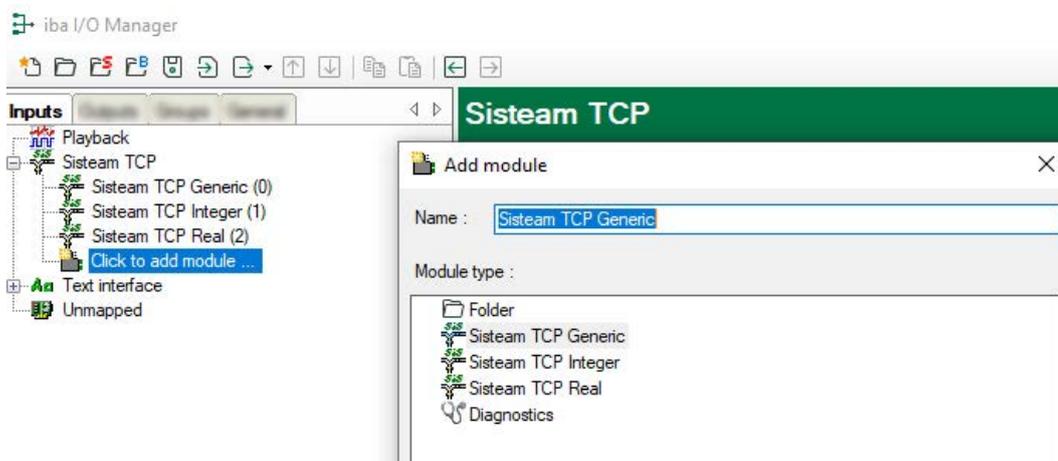
Connection table

See [↗ Connection Check](#), page 24

4.3 Adding a module

Procedure

1. Click on the blue command *Click to add module...* located under each data interface in the *Inputs* or *Outputs* tab.
2. Select the desired module type in the dialog box and assign a name via the input field if required.
3. Confirm the selection with <OK>.



Note



If a TCP/IP connection to one or more SISTEAM PLCs is established, right-click on the interface and choose *Autodetect*. Then the modules are automatically created for all available connections.

Module types

You can add the following module types to the interface:

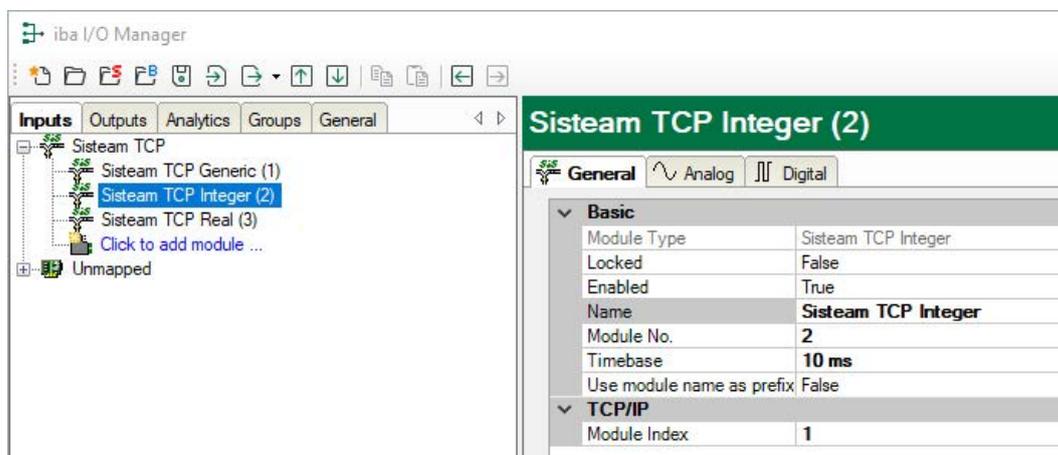
- Sisteam TCP Generic
- Sisteam TCP Integer
- Sisteam TCP Real

See [↗ General Module Settings, page 17](#)

4.4 General Module Settings

To configure a module, select it in the tree structure.

All modules have the following setting options.



Basic settings

Module Type (information only)

Indicates the type of the current module.

Locked

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

Enabled

Enable the module to record signals.

Name

You can enter a name for the module here.

Module No.

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

Timebase

All signals of the module are sampled on this timebase.

Use module name as prefix

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

TCP/IP

Module Index

The module indices are created by a serial number 00...63 and an offset that corresponds to the module type and the license. See ↗ *SISTEAM BUS data*, page 12

For the different module types the possible module indexes are listed below.

SISTEAM Integer module:	00..63 (for module type 1)
SISTEAM Real module:	100..163 (for module type 0)
SISTEAM Generic module:	200..263 (for module type 2)

4.4.1 Module Type Generic

Any data block with max. length of 4096 bytes can be measured by means of the Generic module.

The following module settings are module-specific:

Text encoding

You can select the type of text encoding or the code page here for a correct interpretation and display of the received text data for inputs as well as of the text data to be sent for outputs. Available for selection are, beside system locale according to the Windows system settings (default) and UTF-8 Unicode, all other encodings.

Module Layout

Swap analog signals, Swap digital signals

Option to change the order of the byte evaluation

No. of analog signals/digital 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.

4.4.2 Module Type Integer

For the Integer module, no module specific settings are required. See ↗ *General Module Settings*, page 17

4.4.3 Module Type Real

The Real module allows up to 32 analog signals (Real) and 32 digital signals to be acquired.

The following module settings are module-specific:

Number of analog signals

The number of analog signals is configurable from 1 to 32 (number of digital signals is fixed at 32).

4.5 Signal configuration

Basically, the principle of signal configuration is the same for all modules.

In the tabs *Analog* and *Digital* of each module in the I/O Manager, you can assign name, comment and if required unit and scaling factor to the signals and enable or disable them. It is essential to enter address and data type.

The data to be measured are selected on the SISTEAM side by mapping the signals in the telegram buffer.

Analog and digital tab

Sisteam TCP Generic (0)

General Analog Digital

Name	Unit	Gain	Offset	Address	DataType	Active	Actual
0		1	0	0x0	INT	<input checked="" type="checkbox"/>	0
1		1	0	0x2	INT	<input checked="" type="checkbox"/>	0
2		1	0	0x4	INT	<input checked="" type="checkbox"/>	0
3		1	0	0x6	INT	<input checked="" type="checkbox"/>	0
4		1	0	0x8	INT	<input checked="" type="checkbox"/>	0
5		1	0	0xA	INT	<input checked="" type="checkbox"/>	0
6		1	0	0xC	INT	<input checked="" type="checkbox"/>	0
7		1	0	0xE	INT	<input checked="" type="checkbox"/>	0
8		1	0	0x10	INT	<input checked="" type="checkbox"/>	0
9		1	0	0x12	INT	<input checked="" type="checkbox"/>	0
10		1	0	0x14	INT	<input checked="" type="checkbox"/>	0
11		1	0	0x16	INT	<input checked="" type="checkbox"/>	0
12		1	0	0x18	INT	<input checked="" type="checkbox"/>	0

Other documentation



You can find further information in the *ibaPDA* manual.

Address

The address space is depending on the data type. Hence, an adjustment of address entries may be necessary after change of data types.

The digital signals are addressed via the *Address* and *Bit no.* (0 – 31) columns.

Data Type (analog signals only)

In the fields of this column you can select the data type of each signal. Just click in the corresponding field and select the data type from the drop-down list. The following data types are available: SINT, BYTE, INT, WORD, DINT, DWRD, FLOAT, DOUBLE, STRING[32].

Note



The module *TCP Generic* supports the acquisition and processing of strings as text signals. Therefore, you can select the datatype STRING[32] in the *Analog* tab. In order to convert a text signal order to split it up into several text signals use the *text splitter* module under the *Virtual* interface.

Tip

You can use the automatic fill function in the columns (see *ibaPDA* manual).

5 Diagnostics

5.1 License

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

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

License information		Licenses:
License container:	3-4	ibaPDA-Interface-Codesys-Xplorer (16)
Customer name:
License time limit:	Unlimited	...
Container type:	WIBU CmStick v4.40	...
Container host:
Required EUP date:	01.02.2023	...
EUP date:	31.12.2025	...

5.2 Visibility of the interface

If the interface is not visible despite a valid license, it may be hidden.

Check the settings in the *General* tab in the *Interfaces* node.

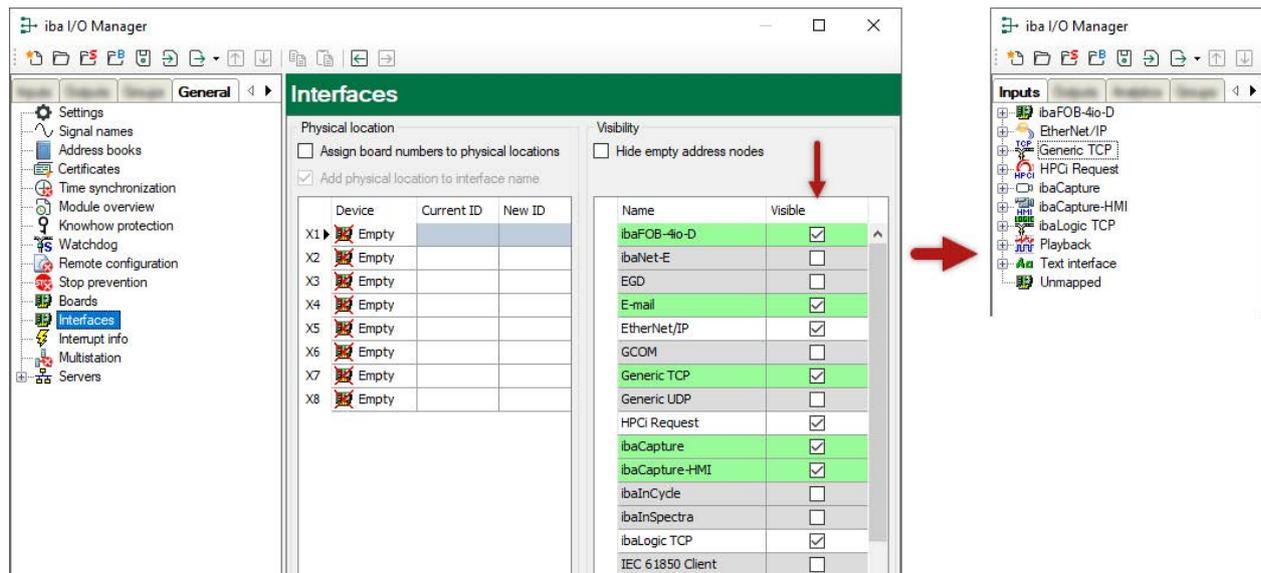
Visibility

The table *Visibility* lists all the interfaces that are available either through licenses or installed cards. These interfaces can also be viewed in the interface tree.

You can hide or display the interfaces not required in the interface tree by using the checkbox in the *Visible* column.

Interfaces with configured modules are highlighted in green and cannot be hidden.

Selected interfaces are visible, the others are hidden:



5.3 Log files

If connections to target platforms or clients have been established, all connection-specific actions are logged in a text file. You can open this (current) file and, e.g., scan it for indications of possible connection problems.

You can open the log file via the button <Open log file>. The button is available in the I/O Manager:

- for many interfaces in the respective interface overview
- for integrated servers (e.g. OPC UA server) in the *Diagnostics* tab.

In the file system on the hard drive, you can find the log files of the *ibaPDA* server (`...\ProgramData\iba\ibaPDA\Log`). The file names of the log files include the name or abbreviation of the interface type.

Files named `interface.txt` are always the current log files. Files named `Interface_yyyy_mm_dd_hh_mm_ss.txt` are archived log files.

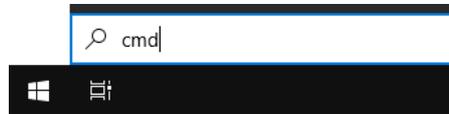
Examples:

- `ethernetipLog.txt` (log of EtherNet/IP connections)
- `AbEthLog.txt` (log of Allen-Bradley Ethernet connections)
- `OpcUAServerLog.txt` (log of OPC UA server connections)

5.4 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 192.168.1.10

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

Ping statistics for 192.168.1.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\Windows\system32>
```

→ With no existing connection you receive error messages.

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

C:\Windows\system32>ping 192.168.1.10

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

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

C:\Windows\system32>
```

5.5 Connection Check

After the configuration was accepted, all connections will be shown in the connections overview sorted according to their module index.

Sisteam TCP							
Properties							
Reset statistics							
	Address	Module index	Message counter	Incomplete errors	Sequence errors	Packet size Actual	Time Actual
0	192.168.110.101	2	739	0	0	82	99.9 ms
1	192.168.110.101	102	175	0	0	146	99.9 ms
2	?	?	?	?	?	?	?
3	?	?	?	?	?	?	?

The background color of the lines has the following meaning:

Color	Meaning
Green:	The connection is OK. The <i>ibaPDA</i> module timebase is equally quick or slower than the telegram cycle. The current telegram cycle is shown in the column "Time Actual".
Orange:	The connection is OK, but the telegram cycle is significantly slower than the <i>ibaPDA</i> module timebase. It is recommended to adjust the module timebase to the telegram cycle.

If the connections are not displayed or only partially, this may have the following causes:

- Sender is in Stop mode
- No Ethernet connection between *ibaPDA* PC and the SISTEAM control
- Error in the connection configuration:
 - incorrect remote IP address
 - The *ibaPDA* port number and the connection configuration do not match.
 - The port number is blocked by the firewall.
- Wrong module index specified in the telegram header

Other errors:

- If the telegram counters do not increment continuously, the telegrams are not called cyclically on the sender side.
- If values in the columns "Incomplete errors" and/or "Sequence errors" are incremented, this points to one of the following errors:
 - The "message_length" in the telegram header does not meet the expected value.
 - The "sequence_counter" in the telegram is not incremented correctly.
 - The "Delayed Acknowledge" problem occurs, see [↗ TCP performance problems caused by Delayed Acknowledge](#), page 26

6 Appendix

6.1 Troubleshooting

6.1.1 TCP performance problems caused by Delayed Acknowledge

Symptoms:

ibaPDA measurements of automation devices using TCP/IP sometimes do not work with cycle times < 200 ms.

Errors shown in *ibaPDA*:

Incomplete telegrams and/or spikes in data values (depending on the sending controller type)

Cause:

There are different variants of handling "acknowledge" in the TCP/IP protocol:

The standard WinSocket works in accordance with RFC1122 using the "delayed acknowledge" mechanism (Delayed ACK). It specifies that the "acknowledge" is delayed until other telegrams arrive in order to acknowledge them jointly. If no other telegrams arrive, the ACK telegram is sent after 200 ms at the latest (depending on the socket).

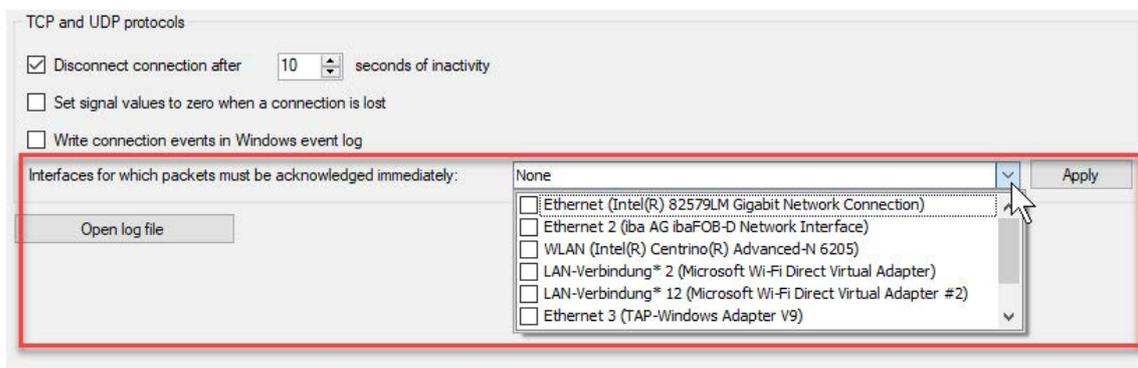
The data flow is controlled by a "sliding window" (parameter Win=nnnn). The recipient specifies how many bytes it can receive without sending an acknowledgment.

Some controllers do not accept this response, but instead, wait for an acknowledgment after each data telegram. If it does not arrive within a certain period of time (200 ms), it will repeat the telegram and include any new data to be sent, causing an error with the recipient, because the old one was received correctly.

Remedy:

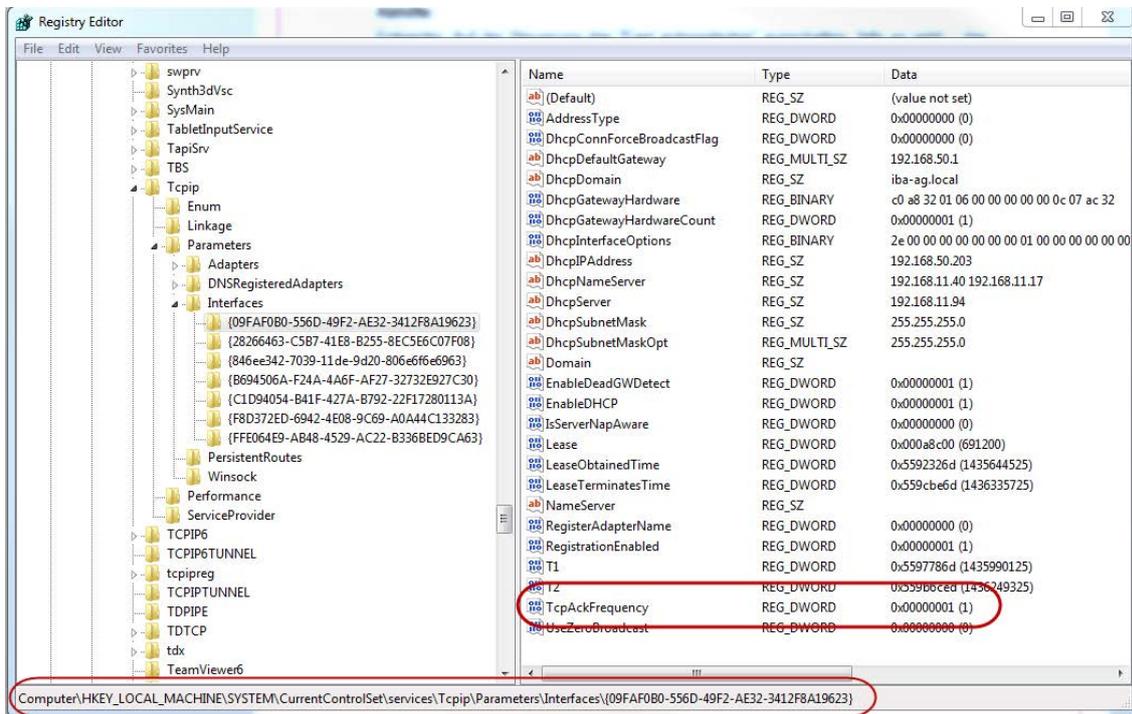
The "delayed acknowledge" can be switched off individually for each network adapter via an entry in the Windows Registry. For easy modification, *ibaPDA* offers a corresponding dialog in the I/O manager under *General* in the tab *Settings*.

In the list of network adapters, select those for which you want to disable "delayed acknowledge" and click <Apply>.



Thus, the parameter "TcpAckFrequency" (REG_DWORD = 1) is created in the registry path of the selected network adapters:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{InterfaceGUID}



Note



Basically, you can avoid such TCP-specific problems by using *UDP* instead of *TCP*.

The User Datagram Protocol (UDP) is a minimal network protocol that is not connection-oriented and is unsecured against telegram loss. Among other things, reception acknowledgement of the sent data is dispensed with. In stable and high-performance networks, however, this is not of significant importance and can be neglected due to the cyclic data transmission common with *ibaPDA*.

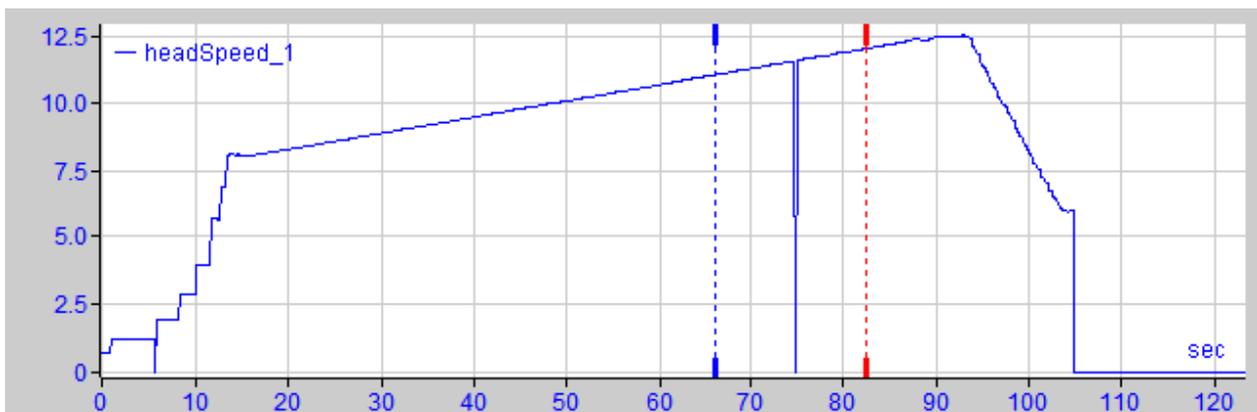
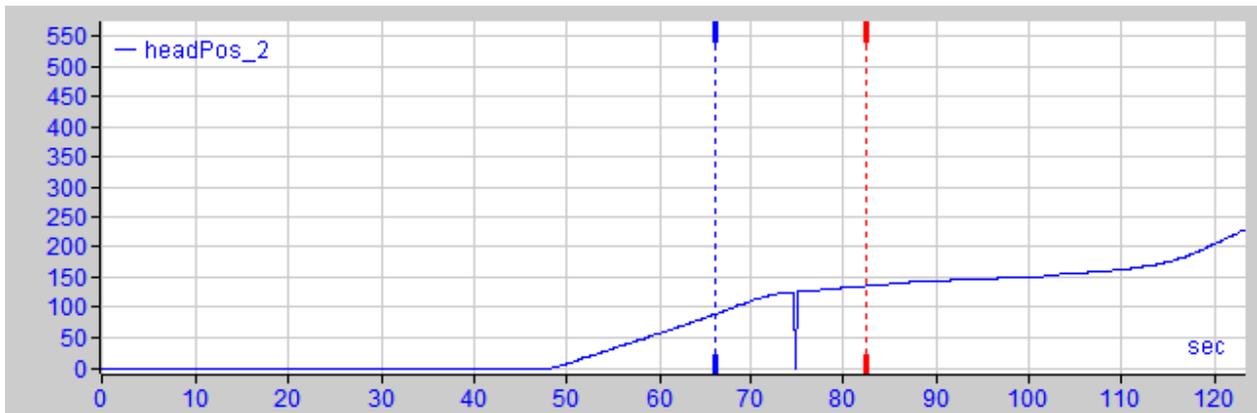
6.1.2 TCP data corruption resulting from the Nagle's Algorithm

Symptoms:

ibaPDA measurements of automation devices using TCP/IP show spikes in the data.

Errors shown in *ibaPDA*:

Incomplete telegrams and/or spikes in the data values (see examples in the following figures)



Cause:

Nagle's algorithm, named after its creator John Nagle, is one mechanism for improving TCP efficiency by reducing the number of small packets sent over the network and collecting several data blocks before sending the data over the network.

Since the Generic TCP interface does not use an application level protocol, the receiver *ibaPDA* cannot handle these merged messages correctly. The Generic TCP interface expects only 1 datagram per TCP message with always the same layout and length.

But the Nagle's Algorithm and the option *Delayed ACK* (Delayed Acknowledge, see 6.1.1, page 26) do not play well together in a TCP/IP Network:

The Delayed ACK mechanism tries to send more data per segment if it can.

But part of Nagle's algorithm depends on an ACK to send data.

Nagle's algorithm and Delayed ACKs together create a problem because Delayed ACKs are waiting around to send the ACK while "Nagle's" is waiting around to receive the ACK!

This creates random stalls of 200 ms - 500 ms on segments that could otherwise be sent immediately and delivered to the receive-side stack of *ibaPDA* as application.

Remedy:

We recommend starting with disabling the *Delayed ACK* mechanism as explained in chapter 6.1.1, page 26. In a typical real-time application, the transmitter will then send the new data to *ibaPDA* with a certain cycle time, since the previous data has been acknowledged immediately. Depending on the implementation of the TCP/IP stack on the sender's side, the Nagle's algorithm can still become active and automatically aggregate a number of small buffer messages, causing the algorithm to purposely slow down the transmission.

This can also happen sporadically due to a momentary overload on the sender side that causes the stack to merge some messages.

To disable Nagle's buffering algorithm, use the *TCP_NODELAY* socket option. The *TCP_NODELAY* socket option allows the network to bypass Nagle's-induced Delays by disabling Nagle's algorithm, and sending the data as soon as it is available.

Enabling *TCP_NODELAY* forces a socket to send the data in its buffer, whatever the packet size. The *TCP_NODELAY* flag is an option that can be enabled on a per-socket basis and is applied when a TCP socket is created.

(See *Socket.NoDelay* property in .NET applications in the *System.Net.Sockets* namespace.)

Note

Basically, you can avoid such TCP-specific problems by using *UDP* instead of *TCP*.

The User Datagram Protocol (UDP) is a minimal network protocol that is not connection-oriented and is unsecured against telegram loss. Among other things, reception acknowledgement of the sent data is dispensed with. In stable and high-performance networks, however, this is not of significant importance and can be neglected due to the cyclic data transmission common with *ibaPDA*.

6.2 References

- ibaPDA manual, <https://www.iba-ag.com/en/downloads>
- A TCP/IP tutorial , RFC1180 <ftp://ftp.ripe.net/rfc/rfc1180.txt>
- Transmission Control Protocol, RFC793 <ftp://ftp.ripe.net/rfc/rfc793.txt>
- Ingelectric DOCUMENT No. AI-6432 1106A: Description of Ethernet Communication

7 Support and contact

Support

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

Note



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

Contact

Headquarters

iba AG
Koenigswarterstrasse 44
90762 Fuerth
Germany

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

Mailing address

iba AG
Postbox 1828
D-90708 Fuerth, Germany

Delivery address

iba AG
Gebhardtstrasse 10
90762 Fuerth, Germany

Regional and Worldwide

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

www.iba-ag.com