ibaPDA-Interface-VIP-TCP/UDP
Data Interface TCP/UDP for VIP Protocol

Manual
Issue 2.8

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.

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Revision - Chapter / Page</th>
<th>Author</th>
<th>Version SW</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.8</td>
<td>03-2020</td>
<td>TCP/IP protocol variants</td>
<td>RM/IP</td>
<td>6.33.2</td>
</tr>
</tbody>
</table>

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 ............................................................................................................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/UDP for VIP .........................................................................................8
  3.1 General Information .................................................................................................8
  3.1.1 What is VIP? ..............................................................................................................8
  3.1.2 What is TCP/IP and UDP? ..........................................................................................9
  3.1.3 How does VIP work? ................................................................................................10
  3.2 Communication between ibaPDA and ABB (Controller) ..........................................11
  3.3 Data Structure ........................................................................................................ 14
  3.3.1 VIP_32_Integer: 32 integer values + 32 binary values ...........................................14
  3.3.2 VIP_8_Real: 8 Real values + 32 binary values .........................................................14
  3.3.3 VIP_16_Real: 16 Real values + 32 binary values ..................................................14
  3.3.4 VIP_32_Real: 32 Real values + 32 binary values ...................................................15
  3.3.5 VIP_Generic: max. 4096 Bytes ................................................................................15
  3.4 Configuration Guide ................................................................................................ 16
  3.5 ibaPDA Configuration & Engineering ......................................................................17
  3.5.1 General Settings ......................................................................................................17
  3.5.2 General Interface Settings ......................................................................................17
  3.5.3 General Module Settings ........................................................................................18
  3.5.4 General Signal Configuration ..................................................................................20
  3.5.5 Module Type Integer .............................................................................................. 21
  3.5.6 Module Type Real ...................................................................................................21
  3.5.7 Module Type Generic ............................................................................................. 21
  3.5.8 Module Diagnostics ................................................................................................22

4 Diagnostics......................................................................................................................23
  4.1 License Check ...........................................................................................................23
  4.2 Interface Visibility .....................................................................................................23
1 About this Manual

This document describes the function and application of the software interface *ibaPDA-Interface-VIP-TCP/UDP*.

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 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 of *ibaPDA-Interface-VIP-TCP/UDP* 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:

<table>
<thead>
<tr>
<th>Action</th>
<th>Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Menu command</td>
<td><em>Menu Logic diagram</em></td>
</tr>
<tr>
<td>Calling the menu command</td>
<td><em>Step 1 – Step 2 – Step 3 – Step x</em></td>
</tr>
<tr>
<td></td>
<td>Example: Select the menu <em>Logic diagram - Add - New function block</em>.</td>
</tr>
<tr>
<td>Keys</td>
<td>&lt;Key name&gt;</td>
</tr>
<tr>
<td></td>
<td>Example: &lt;Alt&gt;; &lt;F1&gt;</td>
</tr>
<tr>
<td>Press the keys simultaneously</td>
<td>&lt;Key name&gt; + &lt;Key name&gt;</td>
</tr>
<tr>
<td></td>
<td>Example: &lt;Alt&gt; + &lt;Ctrl&gt;</td>
</tr>
<tr>
<td>Buttons</td>
<td>&lt;Key name&gt;</td>
</tr>
<tr>
<td></td>
<td>Example: &lt;OK&gt;; &lt;Cancel&gt;</td>
</tr>
<tr>
<td>File names, paths</td>
<td>&quot;Filename&quot;, &quot;Path&quot;</td>
</tr>
<tr>
<td></td>
<td>Example: &quot;Test.doc&quot;</td>
</tr>
</tbody>
</table>
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/UDP for VIP protocols:

- ibaPDA v6.33.2 oder höher
- License for ibaPDA-Interface-VIP-TCP/UDP
- Network connection 10/100 Mbits
- ABB controller with TCP/IP communication interface, e.g. CI861

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 or UDP communication on a separate network segment to exclude a mutual influence by other network components.

System Restrictions

- For different ways of handling the TCP/IP-Acknowledge
  see TCP/IP protocol variants, page 28 (all ibaPDA versions).

Licenses

<table>
<thead>
<tr>
<th>Order no.</th>
<th>Product Designation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31.001065</td>
<td>ibaPDA-Interface-VIP-TCP/UDP</td>
<td>Extension license for an ibaPDA system by one TCP/IP and UDP/IP interface Number of connections: 64</td>
</tr>
<tr>
<td>31.101065</td>
<td>one-step-up-Interface-VIP-TCP/UDP</td>
<td>Extension license for an existing interface ibaPDA-Interface-VIP-TCP/UDP by other 64 VIP-TCP/UDP connections, max. 3 extensions permitted</td>
</tr>
</tbody>
</table>

Table 1: Available TCP/UDP interface licenses
3 Data Interface TCP/UDP for VIP

3.1 General Information

3.1.1 What is VIP?

The Vendor Internet Protocol (VIP) serves as a priority as communication between the ABB AC450RMC controller and other computers or systems which do not come from ABB but can process this protocol.

The VIP functionality depends on the Transmission Control Protocol (TCP), the User Datagram Protocol (UDP) and the Internet Protocol (IP) for Ethernet.

Within the context of this manual, the VIP protocol is used to measure different data from an ABB controller with the data measurement system \textit{ibaPDA}.

The signals to be measured are selected by mapping the values in the telegram buffer whose data blocks are defined by the module types of \textit{ibaPDA}. The telegrams are sent to the \textit{ibaPDA} PC as standard transmitter block.

Three module types are defined in \textit{ibaPDA Interface VIP TCP/UDP}:

- **Integer**: 32 analog values (integer) and 32 binary signals
- **Real**: 8, 16 oder 32 analog values (Real) and 32 binary values
- **Generic**: any data structure with a maximum length of 40961 bytes

Every module is assigned to a connection On \textit{ibaPDA} side up to max. 256 connections can be established. On the ABB side, the maximum number of connections depends on the CPU type.

The following ABB controller can communicate with \textit{ibaPDA} via the VIP protocol:

- AC450 RMC
- AC800M
- AC80
- AC800 PEC

As a main advantage, this type of data acquisition does not require any special hardware if the controller already features an Ethernet connection.
3.1.2 What is TCP/IP and UDP?

TCP/IP is a data transport protocol which can contain data of any application or participant. This means that TCP/IP is a means to transmit data via a default protocol to default interfaces (e.g. Ethernet network cards offered by a wide range of providers).

Even if a TCP/IP driver can send and receive data, the user data content has to be interpreted by the user. Only the user data are significant to the user.

<table>
<thead>
<tr>
<th>TCP/IP Package</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
</tr>
<tr>
<td>Data</td>
</tr>
</tbody>
</table>

The TCP/IP message header normally contains not only the control information but also the source and target address of the message. This part of data contains data with a specific structure so that an ibaPDA application can interpret these.

For the connection between ABB and ibaPDA not only the understanding of the data structure but also the transmission order and the message header is important. Thus, it is possible to write a specific TCP/IP drive which reads the data packages and enables them for the ibaPDA interface for transcription and analysis. ibaPDA works with the TCP/IP drive like a connecting server, the automation devices work like clients. This means that ibaPDA monitors the information sent and the connection requirements.

ibaPDA can work on the ABB TCP/IP connection with a maximum sampling rate of 5 ms. However, additional restrictions on the sender side can occur. These will be specified later.

The Transmission Control Protocol, short TCP, is a connection-oriented protocol.

The User Datagram Protocol, short UDP, is a connectionless transport protocol and works on the layer 4, the transport layer, of the OSI layer model. Its function is similar to that of the connection-oriented TCP. However, it works connectionless and therefore insecure. This means that the sender does not know whether the data packets it has sent have actually arrived. TCP sends confirmations upon receiving data, UDP does not. This method has the advantage that the packet header is much smaller and no acknowledgments have to be sent over the transmission path. In principle, this enables a slightly higher data rate.
3.1.3 How does VIP work?

The communication topology is displayed in the following figure:

```
network
    node1
        link1
            channel1
            channel2
                ...
            channel n
        link2
    node2
        link1
```

This means that an unique node number (node 1 ... 99) as well as an unique IP address were assigned to every station on the ethernet. Every server can establish connections to a maximum of five different data sources. Every station (controller) can be configured as client or server (call VIP NETW, VIP-NODE, VIP-LINK, see also VIP manual, pages 43, 45 and 39).

The following figure TCP/IP VIP Telegram Structure provides details on the VIP package structure within a TCP/IP or UDP package:

![TCP/IP VIP Telegram Structure](image)

The maximum length of TCP/IP message blocks is 65535 bytes. Usually, the message length contains the number of bytes of the complete data package, header included. Message ID can be configured and can adopt any value needed by the user. Then follow any number of data blocks which are always identically structured. The first data are always INT16, then Bool16 etc., even until the byte arrays at the end of each block. When data types are missing, no blank characters will be filled in. The overall structure of the data block has to be determined bindingly for both sides (sender and recipient).
Example
If the user only sends two INT16, one bool32 and one INT32 value with the message ID 2, the data structure will be as follows:

| 16 | 2 | Int16 Val1 | Int16 Val2 | Bool32 Val | Int32 Val |

This means that the length in bytes is 16. 2 for message length, 2 for message ID, 2x2 for both INT16 values and 4 each for Bool32 and INT32.

Check the Data Package: Please note that the notation is in hex format and that the single bytes are represented by two characters, however the bytes are separated by a small blank portion.

The complete TCP/IP package is displayed in different colours depending on their affiliation. At the end of the section marked in black in the table of TCP/IP VIP Content of the Data Package, the data section of the VIP packages begins. The first two bytes 00 4A represent the message length in bytes, consequently 74 bytes. The two following bytes 00 and 01 represent the message ID, here 1. Then follows the 2 byte INT16 value 4C 9D (19613 decimal). Thereafter, 32 INT16 values follow with the content 00 00 to 00 1F (0 to 31 decimal). In the end, there are 4 bytes with the value INT32 1. Thus, together a total length of 74 bytes.

3.2 Communication between ibaPDA and ABB (Controller)
iba AG developed a TCP/IP drive to extract data from the ABB VIP package and to display this data on ibaPDA.

Supported connections in ibaPDA:

- With ibaPDA from version 6.14.0 in total 64 modules of the type Integer, Real and/or “Generic” are configured for the interface TCP/IP-VIP. Every integer or real module can contain up to 32 analog and digital signals. Generic modules can contain up to 1000 analog and digital signals.
From *ibaPDA* version 6.31.0 up to 64 modules (connections) per interface as before will be supported. However, 4 licenses in total can be used in *ibaPDA*, as a result up to 256 connections are possible. The maximum length of the message is limited to 4096 bytes. Generic modules can contain up to 1000 analog and 1000 digital signals.

From *ibaPDA* version 6.33.2 the UDP protocol will be supported as well. The sender can send the data via TCP/IP or UDP. Communication parameter and data structure are identical. *ibaPDA* works as a server which listens to clients who have a valid connect-request and then send data (see also table *Communication Principle ibaPDA - ABB AC450RMC*). Every TCP/IP or UDP connection can be seen in the diagnostics window of *ibaPDA* or in the I/O manager. *ibaPDA* is already preset so that the port 5001 monitors the ABB VIP information. Therefore this has to be entered as target port. It is important to outline that the IP address can only occur once, however more than one connection per IP address is possible. Every connection corresponds to a module on *ibaPDA*, therefore an unique message ID (complies with the *ibaPDA* module index) is needed.

The message ID has to be unique in the complete system, even if several ABB controller are used.

This means that the ABB VIP controller can establish up to 5 links and this as a fact every controller can use up to 5 *ibaPDA* modules. Every module needs a unique source port number for this which will be provided automatically by the system when the user configures a new link for every module.

**Module index:**

Module Index is the identifier for assigning the data record to the interface module in *ibaPDA*. The module type is also encrypted in this index: The index is created by a serial number 00....63 and an offset that corresponds to the module type and license.

<table>
<thead>
<tr>
<th>Module type</th>
<th>1. License</th>
<th>2. License</th>
<th>3. License</th>
<th>4. License</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integer</td>
<td>0-63</td>
<td>1000-1063</td>
<td>2000-2063</td>
<td>3000-3063</td>
</tr>
<tr>
<td>Real</td>
<td>100-163</td>
<td>1100-1163</td>
<td>2100-2163</td>
<td>3100-3163</td>
</tr>
<tr>
<td>Generic</td>
<td>200-263</td>
<td>1200-1263</td>
<td>2200-2263</td>
<td>3200-3263</td>
</tr>
</tbody>
</table>

The module index complies with the *ibaPDA* module settings. This value must not be changed during data transmission.

The message ID has to be entered in the module index in order to inform *ibaPDA* about which module should be addressed. This ID has to be entered on the ABB VIP page.

**Sequence counter:**

*ibaPDA* is furthermore able to detect errors in communication. For this function *iba* has determined that the first data bytes should form an upwards counter which has to be filled by increments of 1 with every transmission cycle. This counter has to be the first INT16 value on the ABB side. In the event of an overflow, the counter must jump from 32767 to -32768 (0x7FFF 0x8000) or from 65535 to 0 (0xFFFF 0x0000).
Regarding figure *Communication Principle ibaPDA - ABB AC450RMC* 3 please note that two ABB AC450 controller act as clients, whereas every controller has its own Ethernet module which is connected to the network with one cable each. There is another cable which connects *ibaPDA* with the hub (the cables are displayed here in different thickness and colour). The network has three communication partner with unique IP address each.

Note: If the connections are carried out as UTP, ThinNetCoax or optical has to be determined by the respective application. Since three participants are available, a minimum of three modules within *ibaPDA* are used (indicated here with three curves in different colors). Please note that every participant has a random source port number (940, 831 and 832), however they all have the same target port number (Target Port No.) have 5001.

**Note**

Important for the use of a cross over cable with the ABB controller and *ibaPDA*:

Since most of the ABB controller communicate with 10MBit/s in half duplex mode, we recommend to set the Ethernet interface of the *ibaPDA* PC to 10 Mbit/s, half duplex as well if you use a cross-over cable.

If you cannot set the transmission speed to half duplex, then use a network hub or switch between the controller and *ibaPDA*. 

---

**ibaPDA-Interface-VIP-TCP/UDP**

**Data Interface TCP/UDP for VIP**

---

**Fig. 2: Communication principle ibaPDA - ABB AC450RMC**
3.3 Data Structure

Since *ibaPDA* has to know the exact situation and type of each value,iba has determined the following telegram structure for the different data formats:

For each message only one telegram structure can be used.

### 3.3.1 VIP_32_Integer: 32 integer values + 32 binary values

<table>
<thead>
<tr>
<th>rel. #</th>
<th>bytes</th>
<th>C type</th>
<th>Description</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>2</td>
<td>short int</td>
<td>Message Length</td>
<td>set to 74 Byte.</td>
</tr>
<tr>
<td>02</td>
<td>2</td>
<td>short int</td>
<td>Message ID</td>
<td></td>
</tr>
<tr>
<td>04</td>
<td>2</td>
<td>unsigned short</td>
<td>1. INT16-Signal</td>
<td>Sequence counter: Increment with every transmission cycle! When overflow 65535 0</td>
</tr>
<tr>
<td>06</td>
<td>64</td>
<td>short int</td>
<td>32 analog values</td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>4</td>
<td>long int</td>
<td>32 digital values</td>
<td></td>
</tr>
</tbody>
</table>

### 3.3.2 VIP_8_Real: 8 Real values + 32 binary values

<table>
<thead>
<tr>
<th>rel. #</th>
<th>bytes</th>
<th>C type</th>
<th>Description</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>2</td>
<td>short int</td>
<td>Message Length</td>
<td>Set to 42 byte.</td>
</tr>
<tr>
<td>02</td>
<td>2</td>
<td>short int</td>
<td>Message ID</td>
<td></td>
</tr>
<tr>
<td>04</td>
<td>2</td>
<td>unsigned short</td>
<td>1. INT16-Signal</td>
<td>Sequence counter: Increment with every transmission cycle! When overflow 65535 0</td>
</tr>
<tr>
<td>06</td>
<td>4</td>
<td>long int</td>
<td>32 digital values</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>32</td>
<td>float</td>
<td>8 analog values</td>
<td>IEEE format</td>
</tr>
</tbody>
</table>

### 3.3.3 VIP_16_Real: 16 Real values + 32 binary values

<table>
<thead>
<tr>
<th>rel. #</th>
<th>bytes</th>
<th>C type</th>
<th>Description</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>2</td>
<td>short int</td>
<td>Message Length</td>
<td>set to 74 Byte.</td>
</tr>
<tr>
<td>02</td>
<td>2</td>
<td>short int</td>
<td>Message ID</td>
<td></td>
</tr>
<tr>
<td>rel. #</td>
<td>bytes</td>
<td>C type</td>
<td>Description</td>
<td>Comment</td>
</tr>
<tr>
<td>--------</td>
<td>-------</td>
<td>--------------</td>
<td>----------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>04</td>
<td>2</td>
<td>unsigned short</td>
<td>1. INT16-Signal</td>
<td>Sequence counter: Increment with every transmission cycle! When overflow 65535 0</td>
</tr>
<tr>
<td>06</td>
<td>4</td>
<td>long int</td>
<td>32 digital values</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>64</td>
<td>float</td>
<td>16 analog values</td>
<td>IEEE format</td>
</tr>
</tbody>
</table>

### 3.3.4 VIP_32_Real: 32 Real values + 32 binary values

<table>
<thead>
<tr>
<th>rel. #</th>
<th>bytes</th>
<th>C type</th>
<th>Description</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>2</td>
<td>short int</td>
<td>Message Length</td>
<td>Set to 138 byte</td>
</tr>
<tr>
<td>02</td>
<td>2</td>
<td>short int</td>
<td>Message ID</td>
<td>Module index in <em>ibaPDA</em>, see ¶ <em>Communication between <em>ibaPDA</em> and <em>ABB</em> (Controller)</em>, page 11</td>
</tr>
<tr>
<td>04</td>
<td>2</td>
<td>unsigned short</td>
<td>1. INT16-Signal</td>
<td>Sequence counter: Increment with every transmission cycle! When overflow 65535 0</td>
</tr>
<tr>
<td>06</td>
<td>4</td>
<td>long int</td>
<td>32 digital values</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>128</td>
<td>float</td>
<td>32 analog values</td>
<td>IEEE format</td>
</tr>
</tbody>
</table>

### 3.3.5 VIP_Generic: max. 4096 Bytes

<table>
<thead>
<tr>
<th>rel. #</th>
<th>bytes</th>
<th>C type</th>
<th>Description</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>2</td>
<td>short int</td>
<td>Message Length</td>
<td>maximum 4102 Bytes</td>
</tr>
<tr>
<td>02</td>
<td>2</td>
<td>short int</td>
<td>Message ID</td>
<td>Module index in <em>ibaPDA</em>, see ¶ <em>Communication between <em>ibaPDA</em> and <em>ABB</em> (Controller)</em>, page 11</td>
</tr>
<tr>
<td>04</td>
<td>2</td>
<td>unsigned short</td>
<td>1. INT16-Signal</td>
<td>Sequence counter: Increment with every transmission cycle! When overflow 65535 0</td>
</tr>
<tr>
<td>06</td>
<td>Max. 4096</td>
<td>byte, int, word, double int, double word, float also mixed</td>
<td>Analog / digital values</td>
<td></td>
</tr>
</tbody>
</table>
3.4 Configuration Guide

With ABB controllers which support the TCP/IP or UDP-VIP protocol up to 64 connections can be used to send data to *ibaPDA*, whereas in *ibaPDA* up to 64 modules can be configured.

From *ibaPDA* version 6.31.0 up to 4 TCPIP VIP interfaces will be supported, thus up to 256 connections are possible.

It is ensured that every message ID (module index) is unique and in the TCP/IP or UDP telegram of *ibaPDA* and therefore only one module is addressed in *ibaPDA* respectively.

*ibaPDA* reacts to port 5001 with regards to ABB VIP.

---

**Note**

*From *ibaPDA* version 6.33.2 another port can be set up in *ibaPDA*.*

---

ABB controller can only be used in this port. When using the technostrings it has to be switched to another available port number.

With every link the first INT126 signal has to be configured as upwards counter which has to be incremented for each transmission cycle.

The length of the message (telegram length / message length) depends on the telegram chosen (32 INT, 8, 16 or 32 REAL or GENERIC). The ABB controller have to meet the layout requirements on the sender side, otherwise the message “Incomplete Datapackages” might be displayed in the diagnostics window (I/O manager) of *ibaPDA*.

Only the *ibaPDA* and the controller/s should be connected to the bus.

In order to be able to generate the time base (base clock) for *ibaPDA*, it is recommended that, in any case, an ibaFOB card has to be installed as an interrupt source.

---

**Note**

*Within the VIP network only the *ibaPDA* and the controller used should be connected but no other participants in order to ensure a high dynamic of the system.*
3.5  ibaPDA Configuration & Engineering

If all system requirements are met (see, page System Requirements, page 7), the interface “VIP TCP/UDP” is displayed in the signal tree. There is no need to add the interface manually.

3.5.1  General Settings

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

- Disconnect connection after x 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.

3.5.2  General Interface Settings

The tree structure of the ibaPDA I/O Manager contains the data interface “VIP TCP/UDP”. The interface provides the following functions and configuration options:

Fig. 3: General Interface Settings
Port
Used port PC side. The port number has to be used identically in the VIP connection configuration. From *ibaPDA* version 6.33.2 the port number here can be freely chosen.

*<Reset port to default>*
The port number 5001 is set.

Allow port through firewall
When installing *ibaPDA*, the standard port numbers of the protocols used are automatically entered in the firewall. When the port number is changed here or when the interface was activated, subsequently it is necessary to allow this port in the firewall.

TCP Port / UDP Port
OK is displayed here if the socket can be opened on this port. ERROR is displayed if conflicts occur, e.g. if the port is already occupied.

Connection table
see ![Connection Check, page 26](image)

Adding a module
To add a module, click below the interface and select the desired module type.

Tip
If a TCP/IP or UDP connection already exists, right-click the interface and select Auto Autodetect. Then the correct modules are automatically created for all available connections.
3.5.3 General Module Settings

All modules have the following common setting options.

**Basic settings**

**Module Type (information only)**
Indicates the type of the current module

**Locked**
A module can be locked in order to prevent change of module settings by accident or unauthorized users.

**Enabled**
Deactivated modules are excluded from the 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 timebase. It is recommended to adjust the timebase to the expected TCP/IP or UDP telegram cycle (multiple of the general *ibaPDA* sampling timebase).

**Use name as prefix**
Puts the module prefix in front of the signal names.

**Advanced**

**Swap analog signals**
Possibility to change the evaluation order of the bytes

**Swap digital signals**
Possibility to change the evaluation order of the bytes
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 Communication between ibaPDA and ABB (Controller), page 11.
For a detailed description of the parameters, see the ibaPDA manual.

3.5.4 General Signal Configuration

The data to be measured are selected on the ABB side by mapping the signals in data blocks,
which are cyclically sent to ibaPDA.

In the I/O manager, the signals can be given name and unit (only analog signals) and can be
marked as active and inactive.

Tip
You can use the autofill function for the column (see ibaPDA manual or online
help).

Other documentation
For a detailed description of additional options, see the ibaPDA manual.
3.5.5 Module Type Integer
The integer module allows up to 32 analog values (integer) and 32 binary signals to be acquired. The module does not have any module-specific settings.

3.5.6 Module Type Real
The real module allows up to 32 analog values (real) and 32 binary signals to be acquired. The following module settings are module-specific:

**Number of analog signals**
The number of analog signals to be acquired is configurable in the increments 8, 16 and 32 (the number of digital signals is fixed at 32).

3.5.7 Module Type Generic
Any data block with max. length of 4096 bytes can be measured by means of the moduleGeneric.
The following module settings are module-specific:

**Number of analog signals**
maximum number of configurable analog signals.

**Number of digital signals**
maximum number of configurable digital signals.

For signal configuration, enter the address, i.e. the offset in the telegram buffer and the data type for each variable. Bear in mind that counting starts from the beginning of user data without header.
3.5.8 Module Diagnostics

The tables Analog and Digital of the VIP-TCP/UDP modules show the telegram contents.

The following errors may occur:

- No data are displayed:
  - The telegram buffer on the sender side is not filled correctly
  - The connectors of the transmitter block are wired incorrectly

- Incorrect values are displayed:
  - The telegram buffer on the sender side is not filled correctly (offset error)
  - The byte order is set incorrectly (see General Module Settings, page 18).
  - There are multiple modules with the same module index.

- The digital signals are sorted incorrectly.
  - The byte order is set incorrectly (see General Module Settings, page 18).

- The telegrams arrive not faster than ca. 200 ms with sequence error
  - Problem with "Delayed Acknowledge", see TCP/IP protocol variants, page 28
4 Diagnostics

4.1 License Check

If the interface „VIP TCP/UDP“ is not shown in the signal tree, you can check in the I/O manager under General - Settings - License info or in the ibaPDA service status application whether your license has been correctly detected. The number of licensed connections is shown in brackets.

![Image of the I/O manager showing license information]

4.2 Interface Visibility

If the interface is not visible despite a valid license, it might be hidden. Click the register Interfaces and enable the "Interface VIP TCP/UDP".

![Image of the I/O manager showing interface visibility]

Fig. 4: Enable the interface in the I/O manager
4.3 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.

The log file can be opened 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 will find the log files in the program path of the ibaPDA server (...\Programs\iba\ibaPDA\Server\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)
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.

![Fig. 5: PING successful](image)

With no existing connection you receive error messages.

![Fig. 6: PING unsuccessful](image)
4.5 Connection Check

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

The background color of the lines has the following meaning:

<table>
<thead>
<tr>
<th>Color</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Green:</td>
<td>The connection is OK. The ibaPDA module timebase is equally quick or slower than the telegram cycle. The current telegram cycle is shown in the column “Time Actual”.</td>
</tr>
<tr>
<td>Orange:</td>
<td>The connection is OK, but the telegram cycle is significantly slower than the ibaPDA module timebase. It is recommended to adjust the module timebase to the telegram cycle.</td>
</tr>
</tbody>
</table>

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 ABB 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/IP protocol variants, page 28
5 Appendix

5.1 Restrictions

5.1.1 TCP/IP protocol variants

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

Errors shown in ibaPDA:
Sequence errors and incomplete telegrams.

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. 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>.

![Image of network adapter settings dialog]
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}
6 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 and the license number.

Contact

Head office
iba AG
Koenigswarterstrasse 44
90762 Fuerth
Germany

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

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