ibaPDA with ABB drives ACS880
or DCS880
via Modbus-TCP

Manual
Issue 1.2

Measurement and Automation Systems
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

This manual must not be circulated or copied, or its contents utilized and disseminated, without our express written permission. Any breach or infringement of this provision will result in liability for damages.

©iba AG 2017, All Rights Reserved

The content of this publication has been checked for compliance with the described hardware and software. Nevertheless, deviations cannot be excluded completely so that the full compliance is not guaranteed. However, the information in this publication is updated regularly. Required corrections are contained in the following issues or can be downloaded on the Internet.

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

<table>
<thead>
<tr>
<th>Issue</th>
<th>Date</th>
<th>Revision</th>
<th>Author</th>
<th>Version SW</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2</td>
<td>10/16/2017</td>
<td>Corrections and amendments</td>
<td>Ko</td>
<td>V6.33.1</td>
</tr>
</tbody>
</table>

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

1 About this Manual ........................................................................................................ 2
  1.1 Target group ........................................................................................................ 2
  1.2 Notations ............................................................................................................ 2
  1.3 Used symbols ..................................................................................................... 3

2 System requirements .............................................................................................. 4

3 Configuration ......................................................................................................... 5
  3.1 General .............................................................................................................. 5
  3.2 Device configuration .......................................................................................... 7
  3.2.1 Drive ............................................................................................................. 7
  3.2.2 Software ....................................................................................................... 7
  3.2.3 Network settings ......................................................................................... 7
  3.2.4 Network configuration ............................................................................... 8
  3.3 Configuration ibaPDA ..................................................................................... 9
    3.3.1 Configuration ............................................................................................ 9
    3.3.2 Configuring more connections ................................................................. 11
    3.3.3 Defining analog signals .......................................................................... 11
    3.3.4 Defining digital signals .......................................................................... 15
    3.3.5 Starting the acquisition ......................................................................... 16
    3.3.6 Scaling signals ......................................................................................... 17
    3.3.7 Displaying signals ................................................................................... 18
  3.4 Diagnostics ........................................................................................................ 19
    3.4.1 Connection diagnostics with PING ......................................................... 19
    3.4.2 Checking the connection ....................................................................... 19
    3.4.3 Checking the data .................................................................................... 20
    3.4.4 Response times ....................................................................................... 20

4 Appendix .............................................................................................................. 22
  4.1 TCP/IP protocol variants ................................................................................. 22

5 Support and contact ............................................................................................. 24
1 About this Manual

This documentation contains a comprehensive description of the ibaPDA with ABB drive ACS880 or DCS880 software interface.

1.1 Target group

This manual addresses in particular the 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.

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>Menu Logic diagram</td>
</tr>
<tr>
<td>Calling the menu command</td>
<td>Step 1 – Step 2 – Step 3 – Step x</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>Select the menu Logic diagram - Add - New function block.</td>
</tr>
<tr>
<td>Keys</td>
<td>&lt;Key name&gt;</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>&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:</td>
</tr>
<tr>
<td></td>
<td>&lt;Alt&gt; + &lt;Ctrl&gt;</td>
</tr>
<tr>
<td>Buttons</td>
<td>&lt;Key name&gt;</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>&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:</td>
</tr>
<tr>
<td></td>
<td>&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:
  
  - From an electric shock!
  - Due to the improper handling of software products which are coupled to input and output procedures with control function!

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

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

- **Note**
  
  A note specifies special requirements or actions to be observed.

- **Important note**
  
  Note if some special features must be observed, for example exceptions from the rule.

- **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.

- **Example**
  
  Configuration and application examples for a better understanding.
2 System requirements

The following system requirements are necessary:

ibAPDA:
- ibaPDA version 6.33.1 or more recent
- ibaPDA base license
- License for ibaPDA-Interface-Modbus-TCP-Client (31.001022)
- Network connection 10/100 Mbit

ABB:
- ACS880 or DCS880 drive
- FENA-11 or FENA-21 communication interface
- Drive Composer version V1.8 or more recent

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

---

Note

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

---

Note

When operated on a virtual machine, a dongle with a valid license must be plugged on the host for each virtual machine. The USB ports used are assigned explicitly to the respective virtual machines.
3 Configuration

Note
In the following chapters, all specifications which are described for the usage of drive ACS880 also apply for the usage of drive DCS880.

3.1 General
The drive parameter data can be read in two different ways:

- Direct access to the parameter via Modbus register numbers.
  Modbus register in ibaPDA:
  100 x Par. group + Par. number (16-Bit register) or
  20000 + 200 x Par. group + 2 x Par. number (32-Bit register).
  Thus most of the drive parameters are accessible. Access to parameters with numbers higher than 99 is not possible.

- Reading the data from „Drive Profile Registers“. The range of Modbus register numbers smaller than 100 is used in order to read signals according to „ABB Drive Profiles“.

For „ABB Drive Profile - Enhanced“ or „ABB Drive Profile Transparent 16-bit“ the DATA IN registers are read by Modbus registers 51-65 and the DATA OUT registers are described by Modbus registers 1-15.

Register setting 1-15 (DATA OUT, Parameter group 53 or 56):

<table>
<thead>
<tr>
<th>Register address</th>
<th>Register data (16-bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>00001</td>
<td>ABB Drives Profile Control</td>
</tr>
<tr>
<td>00002</td>
<td>ABB Drives Profile Reference 1</td>
</tr>
<tr>
<td>00003</td>
<td>ABB Drives Profile Reference 2</td>
</tr>
<tr>
<td>00004</td>
<td>DATA OUT 1</td>
</tr>
<tr>
<td>00005</td>
<td>DATA OUT 2</td>
</tr>
<tr>
<td>00006</td>
<td>DATA OUT 3</td>
</tr>
<tr>
<td>00007</td>
<td>DATA OUT 4</td>
</tr>
<tr>
<td>00008</td>
<td>DATA OUT 5</td>
</tr>
<tr>
<td>00009</td>
<td>DATA OUT 6</td>
</tr>
<tr>
<td>00010</td>
<td>DATA OUT 7</td>
</tr>
<tr>
<td>00011</td>
<td>DATA OUT 8</td>
</tr>
<tr>
<td>00012</td>
<td>DATA OUT 9</td>
</tr>
<tr>
<td>00013</td>
<td>DATA OUT 10</td>
</tr>
<tr>
<td>00014</td>
<td>DATA OUT 11</td>
</tr>
<tr>
<td>00015</td>
<td>DATA OUT 12</td>
</tr>
</tbody>
</table>
Register setting 51-65 (DATA IN, Parameter group 52 or 55):

<table>
<thead>
<tr>
<th>Register address</th>
<th>Register data (16-bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>00051</td>
<td>ABB Drives Profile Status</td>
</tr>
<tr>
<td>00052</td>
<td>ABB Drives Profile Actual 1</td>
</tr>
<tr>
<td>00053</td>
<td>ABB Drives Profile Actual 2</td>
</tr>
<tr>
<td>00054</td>
<td>DATA IN 1</td>
</tr>
<tr>
<td>00055</td>
<td>DATA IN 2</td>
</tr>
<tr>
<td>00056</td>
<td>DATA IN 3</td>
</tr>
<tr>
<td>00057</td>
<td>DATA IN 4</td>
</tr>
<tr>
<td>00058</td>
<td>DATA IN 5</td>
</tr>
<tr>
<td>00059</td>
<td>DATA IN 6</td>
</tr>
<tr>
<td>00060</td>
<td>DATA IN 7</td>
</tr>
<tr>
<td>00061</td>
<td>DATA IN 8</td>
</tr>
<tr>
<td>00062</td>
<td>DATA IN 9</td>
</tr>
<tr>
<td>00063</td>
<td>DATA IN 10</td>
</tr>
<tr>
<td>00064</td>
<td>DATA IN 11</td>
</tr>
<tr>
<td>00065</td>
<td>DATA IN 12</td>
</tr>
</tbody>
</table>

Thereby only a maximum of 15 signals per drive can be read, however with the advantage of a very fast response time (<3 ms).

At first the drive parameter for DATA IN and DATA OUT has to be specified by Drive Composer.

**Other documentation**

You will find more detailed information in the „FENA-11/-21 User’s Manual“, chapter „Modbus-TCP-Communication profiles“.
3.2 Device configuration

3.2.1 Drive
For testing the access, we have used an ACS880 Democase with Ethernet Adapter Module FENA 21 on slot 1.

3.2.2 Software
ABB Drive Composer pro v1.8.0.9
ibaPDA v6.35.0

3.2.3 Network settings
Note
Here we use the parameter group “FBA B” (54 f.), because for this test the FENA module is plugged on slot B.

Settings on the ACS880/DCS880 with Drive Composer:
- IP address of the ACS880: 192.168.50.53
- Protocol/Profile: "Modbus TCP ABB Classic" and "Modbus TCP ABB Enhanced", respectively; or „Modbus TCP ABB Transparent 16-bit“.
  - Parameter group 50 "Fieldbus adapter (FBA)"
Parameter group 54 "FBA B settings"

### 3.2.4 Network configuration

**ACSS80 and DCS880 drives with FENA ethernet modules**

---

**iba PDA**

---

**Switch**

daisy chaining through FENA-21

... more nodes
3.3 Configuration ibaPDA

Other documentation
In this document, we only describe the specific settings for connecting the ACS880. In the "ibaPDA-Interface-Modbus-TCP-Client" manual, we describe all other parameters.

Important note
Please consider the settings for the TCP/IP protocol version as described in the appendix.

Start the I/O manager and have a look whether the “Modbus TCP client” license is available and if the “Modbus TCP client” is displayed in the tree structure of the interface.

3.3.1 Configuration
- Add a module “Modbus Client” to the “Modbus TCP Client” interface.
Select the General tab and set the following parameters:

- **Default parameters:** See „ibaPDA“ manual and „Modbus TCP“ manual.

**Specific settings:**
All parameters in bold deviate from the default parameters.

- **Name, Module No., Timebase:** You can either adopt the default settings or change them according to your demands.

- **Swap analog signals:** Depending on data type
  Yields the right byte sequence in combination with the ACS880 parameter 54.22 (word order) “HiLo”.

- **IP address:** IP address of the ACS880
  Corresponds to setting in ACS880 parameter 54.04 ff.

- **Analog type:** Holding registers

- **Digital type:** Holding registers

- **Addresses start at 1:** True
  (Registers are numbered beginning with 1.)

- **Send messages in parallel:** False

- **Maximum gap between registers:** 1

- **No. analog signals:** Here, you can set the length of the analog table (default value 32)

- **No. digital signals:** Here, you can set the length of the digital table (default value 32)
3.3.2 Configuring more connections

If you want to establish connections to more drives, just add more Modbus client modules to the interface.

- Assign a new module name.
- The module number is incremented automatically.
- Enter the IP address of the drive.

You can establish a maximum of 64 connections with one *ibaPDA-Interface-Modbus-TCP client* license. If you want to establish more connections, you have to purchase more licenses (maximum 4).

**Tip**

In case you want to read the same parameters on each drive, then simply copy one module and just adapt the IP address and possibly the module name.

You find the “Use name as prefix” parameter in the module parameters. Using this parameter, you can distinguish the signals in course of the process. The module name is put in front of the signal name.

3.3.3 Defining analog signals

- Select the *Analog* tab and define the following parameters (example):

  ![Modbus-Client (1) Table](image)

Description of the columns:

- **Name**: Assign the signal name, here. For a better orientation, you can enter the parameter number xx.yy and define comment rows.
- **Unit**: Enter the unit of the measurement value.
- **Gain, Offset**: The settings depend on the type of access. Also see *Scaling signals*, chapter 3.3.6.
- **Active**: You have to activate the check box in the rows containing valid parameter settings. Make sure that the check box is disabled in the comment rows.
3.3.3.1 Register address for direct access to the drive parameters:

- 16-bit register:
  Enter the parameter number in the following format:
  Register address = Parameter group*100 + Parameter number
  Hence,
  register 101 equals group 1, parameter 1
  register 161 equals group 1, parameter 61
  register 1211 equals group 12, parameter 11.
  - Data Type: Always enter INT.

- 32-bit register:
  Enter the parameter number in the following format:
  Register address = 20000 + Parameter group*200 + Parameter number*2
  Hence,
  register 20202 equals group 1, parameter 1
  register 20322 equals group 1, parameter 61
  register 22422 equals group 12, parameter 11.
  - Data Type: Always enter DINT.

3.3.3.2 Register address for access to the ABB Drive Profile Registers

- Essential settings with Drive Composer
  - The drive profile is changed to “MB/TCP ABB E” (for ACS880) or “MB/TCP ABB T16” (for ACS880 or DCS880) via parameter 51.2 or 54.2 (Protocol/Profile) respectively.
  - Depending on the slot of the FENA module, the ABB Drives Profile Registers DATA IN are configured in the parameter group 52 (FBA A) or parameter group 55 (FBA B) respectively.
  - The drive parameters to be read are assigned to the DATA IN registers in parameter group 52 or 55 respectively.
ABB Drive Profile Register setting 51-65:

<table>
<thead>
<tr>
<th>Register address</th>
<th>Register data</th>
<th>Parameter for FBA A</th>
<th>Parameter for FBA B</th>
</tr>
</thead>
<tbody>
<tr>
<td>00051</td>
<td>ABB Drives Profile Status</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>00052</td>
<td>ABB Drives Profile Actual 1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>00053</td>
<td>ABB Drives Profile Actual 2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>00054</td>
<td>DATA IN 1</td>
<td>52.1</td>
<td>55.1</td>
</tr>
<tr>
<td>00055</td>
<td>DATA IN 2</td>
<td>52.2</td>
<td>55.2</td>
</tr>
<tr>
<td>00056</td>
<td>DATA IN 3</td>
<td>52.3</td>
<td>55.3</td>
</tr>
<tr>
<td>00057</td>
<td>DATA IN 4</td>
<td>52.4</td>
<td>55.4</td>
</tr>
<tr>
<td>00058</td>
<td>DATA IN 5</td>
<td>52.5</td>
<td>55.5</td>
</tr>
<tr>
<td>00059</td>
<td>DATA IN 6</td>
<td>52.6</td>
<td>55.6</td>
</tr>
<tr>
<td>00060</td>
<td>DATA IN 7</td>
<td>52.7</td>
<td>55.7</td>
</tr>
<tr>
<td>00061</td>
<td>DATA IN 8</td>
<td>52.8</td>
<td>55.8</td>
</tr>
<tr>
<td>00062</td>
<td>DATA IN 9</td>
<td>52.9</td>
<td>55.9</td>
</tr>
<tr>
<td>00063</td>
<td>DATA IN 10</td>
<td>52.10</td>
<td>55.10</td>
</tr>
<tr>
<td>00064</td>
<td>DATA IN 11</td>
<td>52.11</td>
<td>55.11</td>
</tr>
<tr>
<td>00065</td>
<td>DATA IN 12</td>
<td>52.12</td>
<td>55.12</td>
</tr>
</tbody>
</table>

**Important note**

All changes of parameters carried out by Drive Composer will only become effective when parameter 51.27 or 54.27 are set to “Refresh”.

<table>
<thead>
<tr>
<th></th>
<th>54. FBA B settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>FBA B type</td>
</tr>
<tr>
<td>2</td>
<td>Protocol/Profile</td>
</tr>
<tr>
<td>26</td>
<td>Reserved</td>
</tr>
<tr>
<td>27</td>
<td>FBA B par refresh</td>
</tr>
<tr>
<td>28</td>
<td>FBA B par table ver</td>
</tr>
</tbody>
</table>
Example:
Reading of drive parameters "Motor current" (Parameter 01.07) and "Motor torque" (Parameter 01.10):

- **Register address in *ibaPDA*: 54-55**

![Image of ibaPDA interface](image)

- **Mapping with Drive Composer:**
  *ibaPDA* reads Modbus register address 54 which equals parameter 55.1; this parameter contains the value of parameter 1.7 "Motor current".

![Image of Drive Composer interface](image)

- **Result in *ibaPDA*:**

![Image of ibaPDA interface](image)
3.3.4 Defining digital signals

- Select the *Digital* tab and set the following parameters (example):

![Modbus-Client](image)

**Description of the columns:**

- **Name**: Assign the signal name, here. For a better orientation, you can enter the parameter number xx.yy and define comment rows.
- **Register**: Enter the parameter number as described above.
- **Bit no.**: Enter the bit number within the control-/status word.
- **Active**: You have to activate the check box in the rows containing valid parameter settings. Make sure that the check box is disabled in the comment rows.

**Tip**

As an option, you can acquire the control-/status word as 16-bit integer value instead of digital signals. You can then break down the control-/status word with the virtual module “16 bit decoder”.

---
**Tip**

You can export the parameters as text file in the Drive Composer (button “Save” with the target format “Text file (*.prn)”).

You can open the resulting text file using an ASCII Editor or MS Excel. You can copy the symbolic designations of the parameters to **ibaPDA** with copy and paste.

In case there are many signals, it might be worth converting the Excel file into an **ibaPDA** module format. Then, you can import this format into **ibaPDA**.

You get the **ibaPDA** module format when exporting a module in the **ibaPDA** I/O manager.

---

### 3.3.5 Starting the acquisition

With the <Apply> or <OK> button, you start the acquisition by applying the I/O configuration to the **ibaPDA** server.

**ibaPDA** establishes the TCP/IP connection to the ACS880 (Modbus server) and requests the variables defined in the list of measurement values.

See chapter 3.4. for checking the connection and the received variables.

---

**Note**

The received analog values are raw values which might need to be scaled.
3.3.6 Scaling signals

- Scaling for 16-bit access to drive parameters
The received analog values for the 16-bit access are raw values. If you want to get the same current values as the values you can see in the Drive Composer, you have to scale these values.

You can calculate the scaling factors from the ACS880 parameter group 46 "Monitoring/Scaling settings".

![Image of ACS880 parameter group 46](image)

The values have to be defined according to the unit; e.g. the 200% speed equals the value 1500 rpm.

**Example for speed values**

Go to the Analog tab of the Modbus client module and click on the field in the “Gain” column on the row “Motor speed used”. By clicking on the button , the scaling dialog box will be opened. Enter “20000” for X1 and “1500” for Y1 (200.00% equals 1500 rpm). Activate the “Symmetrical” check box.

When leaving the box, the scaling factor 0.075 will be computed and then entered.
Scaling for 32-bit access to drive parameters
For the 32-bit access to the parameters (Register 2xxx, see chapter 3.3.3) all values have already been scaled. For getting the physical values, enter the factor 0.01 in the “Gain” column for all values.

3.3.7 Displaying signals
After having started the acquisition, the signals can be displayed in many different ways:
- as trend graph
- as numerical view
- as graphical objects (only with ibaQPanel license)

As an example, you can select the trend graph by double clicking on the Icon .
The numerical view will be displayed with a double-click on .
Then, you can draw the measured values from the signal tree to the desired view.
3.4 Diagnostics

3.4.1 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.

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

With an existing connection, you receive several replies.

![PING Command Output](image)

With not existing connection, you receive error messages.

3.4.2 Checking the connection
The connections to the drives are established after having accepted the configuration.

If you want to see the connection list, click on the “Modbus TCP Client” interface in the tree structure.

![Modbus TCP Client](image)

Here, you can have a look at the error counters and the response times.
3.4.3  Checking the data

Click on the **Diagnostics** tab in the I/O manager data module.

Here, the current analog and digital values are displayed in the tables.

![Image of I/O manager data module](image)

**Note**

The unscaled raw values are displayed in this table.

3.4.4  Response times

The response times to the variables (drive parameters) essentially depend on the following values:

- the number of variables
- the number of Modbus TCP messages per sample

The number of messages depends on the distribution of the parameters. You can only request successive parameters in a message, as the access to non-existent parameters might result in errors. In case of gaps in the parameter sequence, the access is split into several messages.

**Note**

This is why you have to set the value “Maximum gap between registers” in the module parameters to “1”; see chapter 3.3.1.

**Example:**

In the following sample project, the drive captures 22 analog values and 32 digital values. The distribution of the parameters results in 15 messages with an average response time per cycle of 83 ms.

<table>
<thead>
<tr>
<th>IP Address</th>
<th>Error count</th>
<th>Messages per cycle</th>
<th>Update time</th>
<th>Response time Actual</th>
<th>Response time Average</th>
<th>Response time Min</th>
<th>Response time Max</th>
<th>Response time Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.30.53</td>
<td>0</td>
<td>15</td>
<td>81 ms</td>
<td>81 ms</td>
<td>83 ms</td>
<td>79 ms</td>
<td>129 ms</td>
<td></td>
</tr>
</tbody>
</table>
Calculation:
We can grossly suppose the following response times:
- per message: approx. 3.3 ms
- per 16-bit value: approx. 1.6 ms

Hence, for 10 successive variables, the response time is approx. 20 ms. For 10 individual variables, it is approx. 45 ms.

Examples for response times:

<table>
<thead>
<tr>
<th>Number of variables</th>
<th>Number of messages</th>
<th>Average Response time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>20</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>22</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>27</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>32</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>33</td>
</tr>
<tr>
<td>13</td>
<td>4</td>
<td>35</td>
</tr>
<tr>
<td>14</td>
<td>5</td>
<td>40</td>
</tr>
<tr>
<td>15</td>
<td>5</td>
<td>42</td>
</tr>
<tr>
<td>16</td>
<td>6</td>
<td>47</td>
</tr>
<tr>
<td>17</td>
<td>7</td>
<td>52</td>
</tr>
<tr>
<td>18</td>
<td>8</td>
<td>57</td>
</tr>
<tr>
<td>19</td>
<td>9</td>
<td>62</td>
</tr>
<tr>
<td>20</td>
<td>10</td>
<td>67</td>
</tr>
</tbody>
</table>

Response times for access to the ABB Drives Profile Register:

<table>
<thead>
<tr>
<th>IP Address</th>
<th>Error count</th>
<th>Messages per cycle</th>
<th>Update time</th>
<th>Response time Actual</th>
<th>Response time Average</th>
<th>Response time Min</th>
<th>Response time Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.192.168.30.53</td>
<td>0</td>
<td>7</td>
<td>40 ms</td>
<td>40 ms</td>
<td>40 ms</td>
<td>36 ms</td>
<td>50 ms</td>
</tr>
<tr>
<td>1.192.168.30.53</td>
<td>0</td>
<td>1</td>
<td>1 ms</td>
<td>1 ms</td>
<td>1 ms</td>
<td>1 ms</td>
<td>2 ms</td>
</tr>
</tbody>
</table>

1st line: Direct access to drive parameter: 32 analog values (16-bit)
7 single messages, average response time: 40 ms

2nd line: Access to the ABB Drives Profile Register: 15 analog values (16-bit)
1 message, average response time: 1 ms.

Important note
Since accesses to the drives are always carried out in parallel, these response times are valid for all drives independent of the number of drives connected.
4 Appendix

4.1 TCP/IP protocol variants

Restriction:

*ibaPDA* measurements of automation devices using TCP/IP (SIMATIC S7 - CP443 and CP343, SIMATIC TDC - CP5100 and CP51M1, or other) sometimes do not work with cycle times < 200 ms.

**Error in *ibaPDA***:
Sequence error 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:**
Either: Switch off "fast acknowledge" on the controller if possible. This, however, can entail problems, as frequently there are also running connections to other partners.

Or: Switch off "delayed acknowledge" in Windows.

This is set by a parameter in the Windows Registry:

- Under Windows 2000:
  Parameter "TcpDelAckTicks": REG_DWORD = 0;

  Parameter "TcpAckFrequency": REG_DWORD = 1;

The parameters are absent by default and have to be entered in the path:

"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{InterfaceGUID}"
You have to select the correct interface. Which interface is the correct one can be recognized for example at the currently set IP addresses.
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.

Contact

Headquarters

iba AG
Koenigswarterstr. 44
90762 Fuerth
Germany

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

Regional and worldwide

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