See the Big Picture

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

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.

Issue	Date	Revision	Author	Version SW
1.2	10/16/2017	Corrections and amendments	Ko	V6.33.1

Table of Contents

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:

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

1.3 Used symbols

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

⚠ DANGER

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

- ❑ 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):

Register address	Register data (16-bit)
00001	ABB Drives Profile Control
00002	ABB Drives Profile Reference 1
00003	ABB Drives Profile Reference 2
00004	DATA OUT 1
00005	DATA OUT 2
00006	DATA OUT 3
00007	DATA OUT 4
00008	DATA OUT 5
00009	DATA OUT 6
00010	DATA OUT 7
00011	DATA OUT 8
00012	DATA OUT 9
00013	DATA OUT 10
00014	DATA OUT 11
00015	DATA OUT 12

Register setting 51-65 (DATA IN, Parameter group 52 or 55):

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

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.

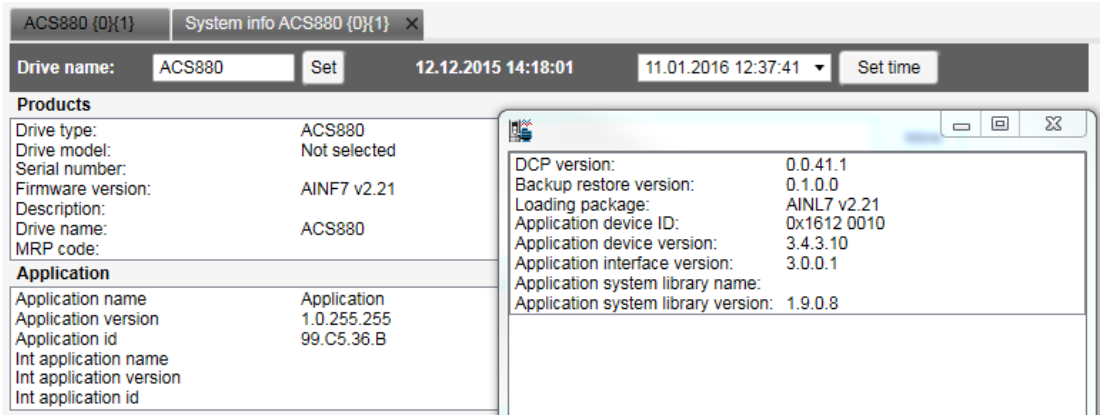
Other documentation

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:

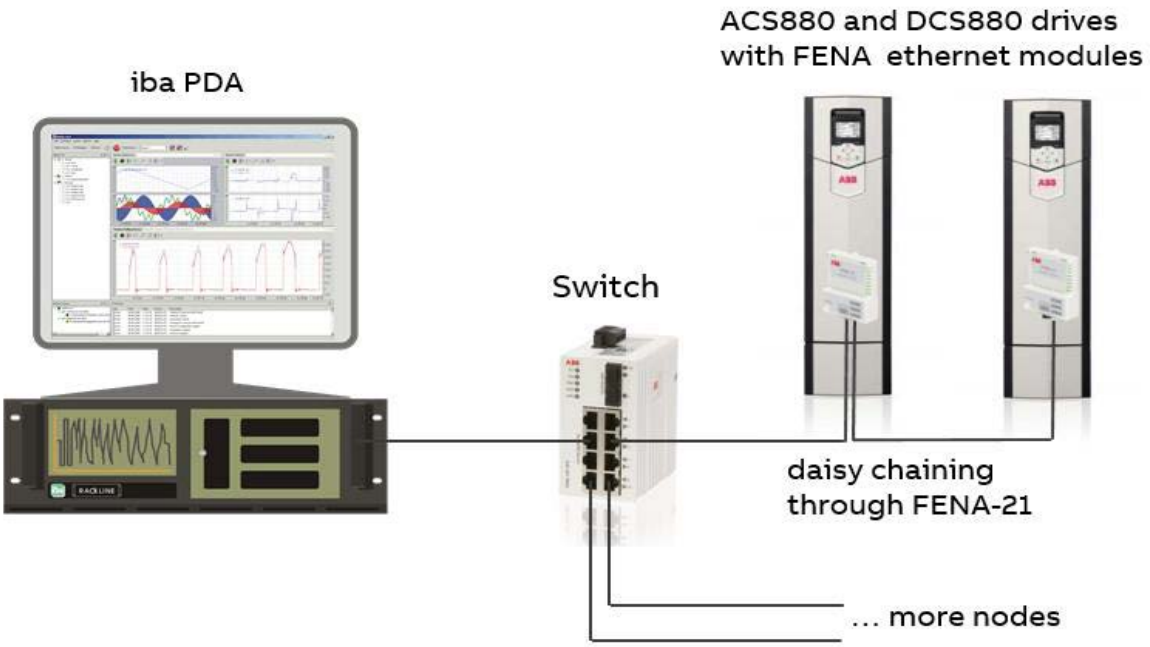
- ❑ IP address of the ACS880: 192.168.50.53
- ❑ Protocol/Profile: "Modbus TCP ABB Classic" and "Modbus TCP ABB Enhanced", respectively; or „Modbus TCP ABB Transparent 16-bit“.
  - Parameter group 50 "Fieldbus adapter (FBA)"

26	FBA A comm supervision f...	0b0000	NoUnit	0b0000	0b1111...	0b0000
31	FBA B enable	Option slot 1	NoUnit			Disable
32	FBA B comm loss func	No action	NoUnit			No action

- Parameter group 54 "FBA B settings"

	54. FBA B settings					
1	FBA B type	**Ethernet**	NoUnit			**None**
2	Protocol/Profile	MB/TCP ABB C	NoUnit			MB/TCP ABB C
3	Commrate	Auto	NoUnit			Auto
4	IP configuration	Static IP	NoUnit			Static IP
5	IP address 1	**192**	NoUnit	0	255	**0**
6	IP address 2	**168**	NoUnit	0	255	**0**
7	IP address 3	**50**	NoUnit	0	255	**0**
8	IP address 4	**53**	NoUnit	0	255	**0**
9	Subnet CIDR	**24**	NoUnit	0	32	**0**
18	FBA B Par18	0	NoUnit	0	65535	0
19	T16 scale	**99**	NoUnit	0	65535	**0**
20	Timeout time	**20**	NoUnit	0	65535	**0**
21	Timeout mode	**Control RW**	NoUnit			**None**
22	Word order	**HiLo**	NoUnit			**LoHi**
23	Address mode	Mode 0	NoUnit			Mode 0
24	FBA B Par24	**128**	NoUnit	0	65535	**0**

3.2.4 Network configuration



3.3 Configuration ibaPDA

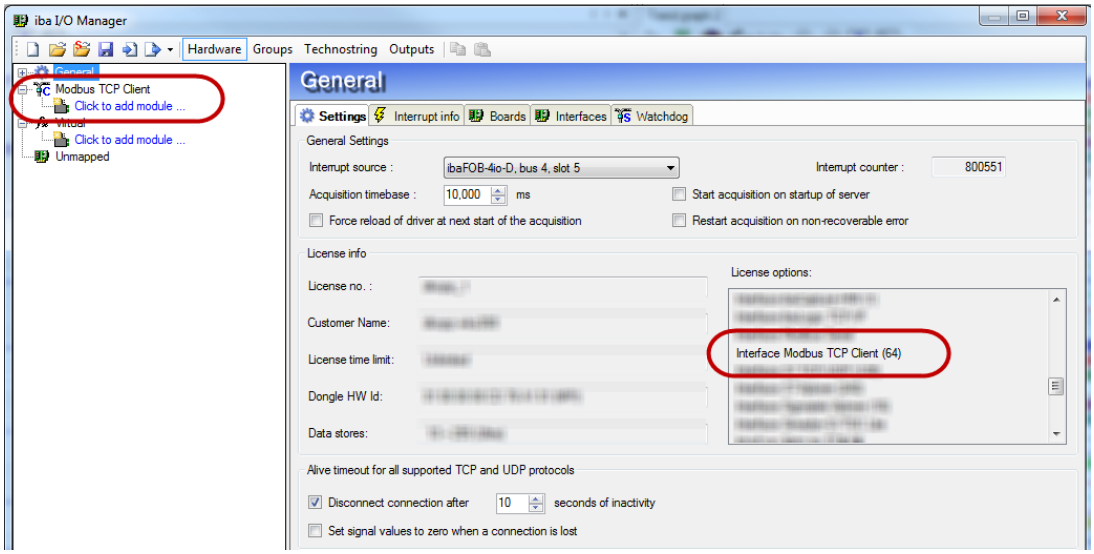
Other documentation

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

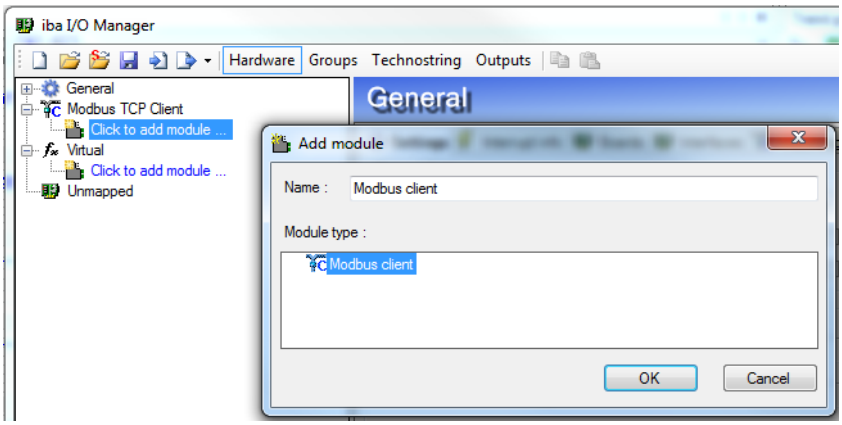
Please consider the settings for the TCP/IP protocol version as described in the appendix.

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.

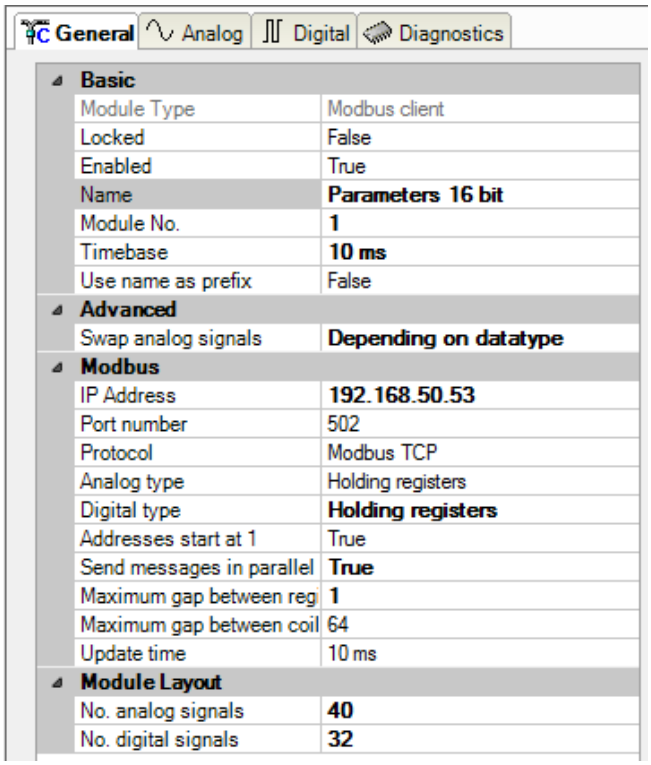


3.3.1 Configuration

❑ 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):

	Name	Unit	Gain	Offset	Register	DataType	Active
0	Par01:****** Actual values **********		1	0	100	INT	☐
1	01.01: Motor speed used	rpm	1	0	101	INT	☑
2	01.02: Motor speed estimated	rpm	1	0	102	INT	☑
3	01.03: Motor speed	%	1	0	103	INT	☑
4	01.06: Output freq	Hz	1	0	106	INT	☑
5	01.07: Motor current	A	1	0	107	INT	☑
6	01.10: Motor torque	%	1	0	110	INT	☑
7	01.11: DC voltage	V	1	0	111	INT	☑
8	01.13: Output voltage	V	1	0	113	INT	☑
9	01.61: Abs motor speed	rpm	1	0	161	INT	☑

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.

54. FBA B settings				
1	FBA B type	**Ethernet**	NoUnit	**None**
2	Protocol/Profile	**MB/TCP ABB E**	NoUnit	**MB/TCP ABB C**
3	Commrate	Auto	NoUnit	Auto

- 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.

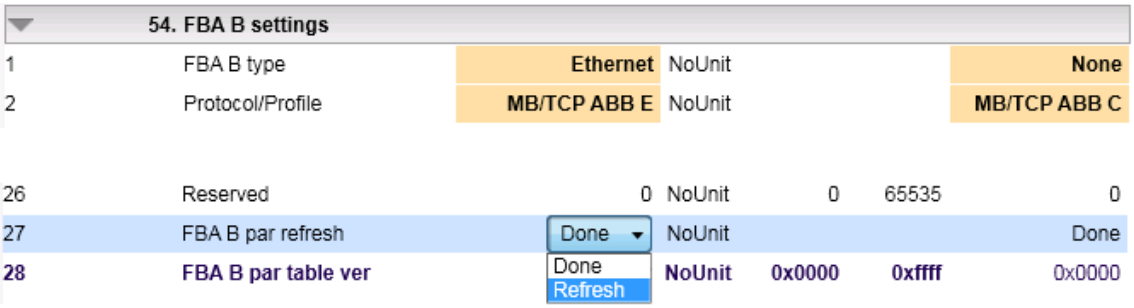
55. FBA B data in				
1	FBA B data in1	**1.1[16]**	NoUnit	**None**
2	FBA B data in2	**1.2[16]**	NoUnit	**None**
3	FBA B data in3	**1.3[16]**	NoUnit	**None**
4	FBA B data in4	**1.6[16]**	NoUnit	**None**

❑ ABB Drive Profile Register setting 51-65:

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

Important note

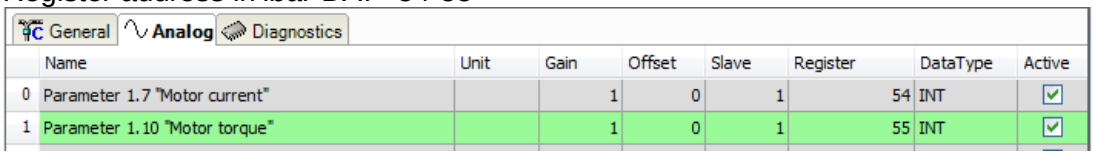
All changes of parameters carried out by Drive Composer will only become effective when parameter 51.27 or 54.27 are set to “Refresh”.



Example:

Reading of drive parameters "Motor current" (Parameter 01.07) and "Motor torque" (Parameter 01.10):

- Register address in *ibaPDA*: 54-55



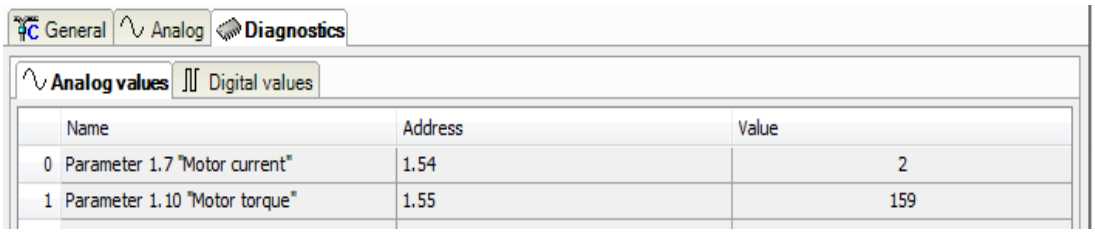
	Name	Unit	Gain	Offset	Slave	Register	DataType	Active
0	Parameter 1.7 "Motor current"		1	0	1	54	INT	☑
1	Parameter 1.10 "Motor torque"		1	0	1	55	INT	☑

- 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".

55. FBA B data in				
1	FBA B data in1	1.7[16]	NoUnit	None
2	FBA B data in2	1.10[16]	NoUnit	None

1. Actual values						
1	Motor speed used	0,00	rpm	-30000,00	30000,00	0,00
2	Motor speed estimated	0,00	rpm	-30000,00	30000,00	0,00
6	Output frequency	0,00	Hz	-500,00	500,00	0,00
7	Motor current	0,00	A	0,00	30000,00	0,00
10	Motor torque	0,0	%	-1600,0	1600,0	0,0

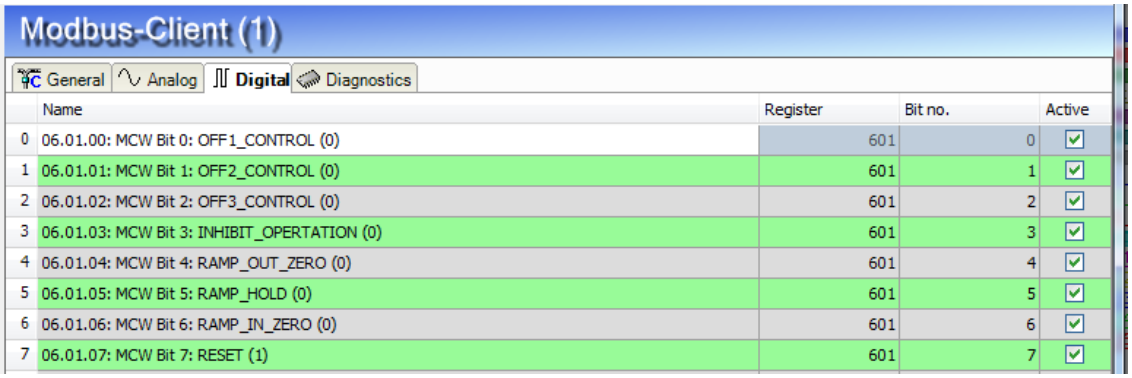
- Result in *ibaPDA*:



	Name	Address	Value
0	Parameter 1.7 "Motor current"	1.54	2
1	Parameter 1.10 "Motor torque"	1.55	159

3.3.4 Defining digital signals

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



Description of the columns:

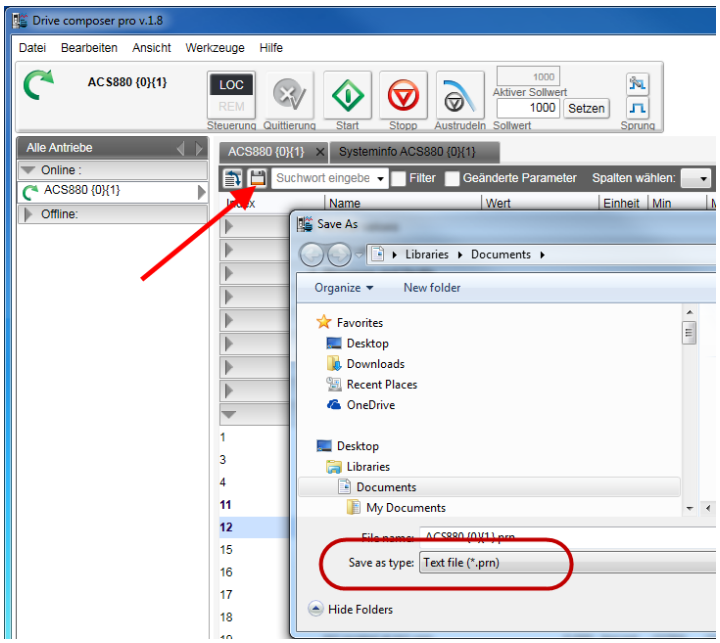
- Name — Assign the signal name, here. For a better orientation, you can enter the parameter number xx.yy and define comment rows.
- Register: — Enter the parameter number as described above.
- Bit no.: — Enter the bit number within the control-/status word.
- Active: — You have to activate the check 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".

ACS880 {0}{1}

Enter keyword | Filter | Not at default | Select columns: | Enable updating

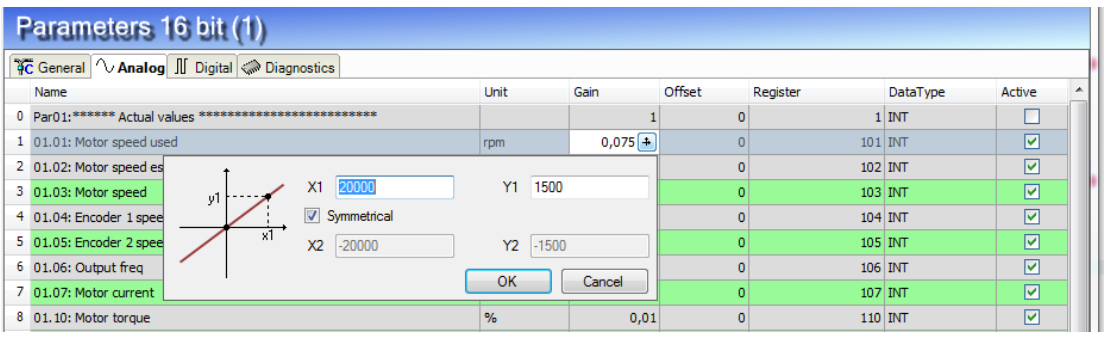
Index	Name	Value	Unit	Min	Max	Default
	45. Energy efficiency					
	46. Monitoring/scaling settings					
1	Speed scaling	1500,00	rpm	0,10	30000,00	1500,00
2	Frequency scaling	50,00	Hz	0,10	1000,00	50,00
3	Torque scaling	100,0	%	0,1	1000,0	100,0
4	Power scaling	1000,00	kW	0,10	30000,00	1000,00
5	Current scaling	10000	A	0	30000	10000
6	Speed ref zero scaling	0,00	rpm	0,00	30000,00	0,00
7	Frequency ref zero scaling	0,00	Hz	0,00	1000,00	0,00
11	Filter time motor speed	500	ms	0	20000	500
12	Filter time output frequency	500	ms	0	20000	500
13	Filter time motor torque	100	ms	0	20000	100
14	Filter time power out	100	ms	0	20000	100
21	At speed hysteresis	100,00	rpm	0,00	30000,00	100,00
22	At frequency hysteresis	10,00	Hz	0,00	1000,00	10,00

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 2xxxx, 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.

Parameters 32 bit (4)

General | Analog | Digital | Diagnostics

	Name	Unit	Gain	Offset	Register	DataType	Active
0	Par01:****** Actual values **********		0,01	0	20200	DINT	☐
1	01.01: Motor speed used	rpm	0,01	0	20202	DINT	☑
2	01.02: Motor speed estimated	rpm	0,01	0	20204	DINT	☑
3	01.03: Motor speed	%	0,01	0	20206	DINT	☑
4	01.04: Encoder 1 speed filtered	rpm	0,01	0	20208	DINT	☑
5	01.05: Encoder 2 speed filtered	rpm	0,01	0	20210	DINT	☑

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.

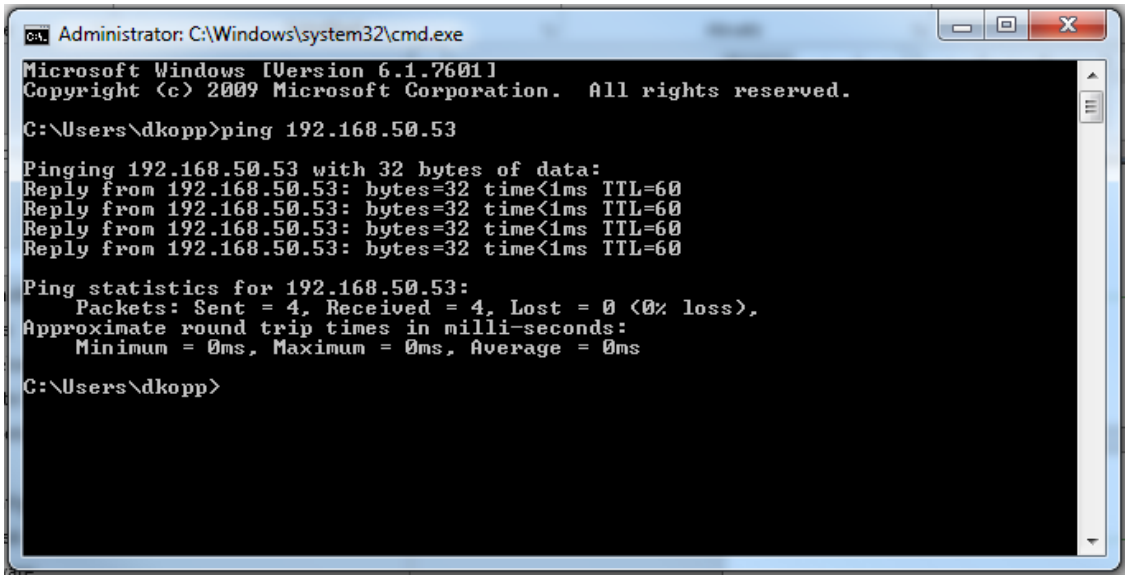
3.4 Diagnostics

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.

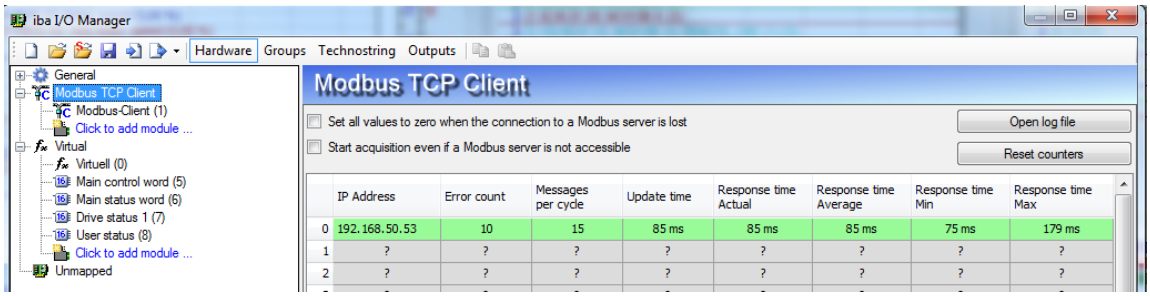


With not existing connection, you receive error messages.

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.

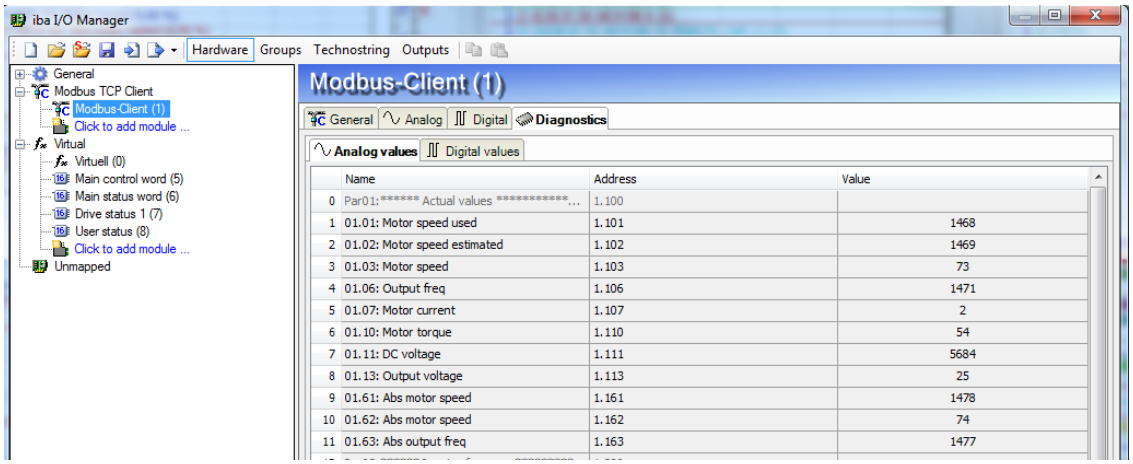


Here, you can have a look at the error counters and the response times.

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.



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.

	IP Address	Error count	Messages per cycle	Update time	Response time Actual	Response time Average	Response time Min	Response time Max
0	192.168.50.53	0	15	81 ms	81 ms	83 ms	79 ms	129 ms
1	?	?	?	?	?	?	?	?

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:

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

Response times for access to the ABB Drives Profile Register:

	IP Address	Error count	Messages per cycle	Update time	Response time Actual	Response time Average	Response time Min	Response time Max
0	192.168.50.53	0	7	40 ms	40 ms	40 ms	36 ms	50 ms
1	192.168.50.53	0	1	1 ms	1 ms	1 ms	1 ms	2 ms
2	?	?	?	?	?	?	?	?

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.

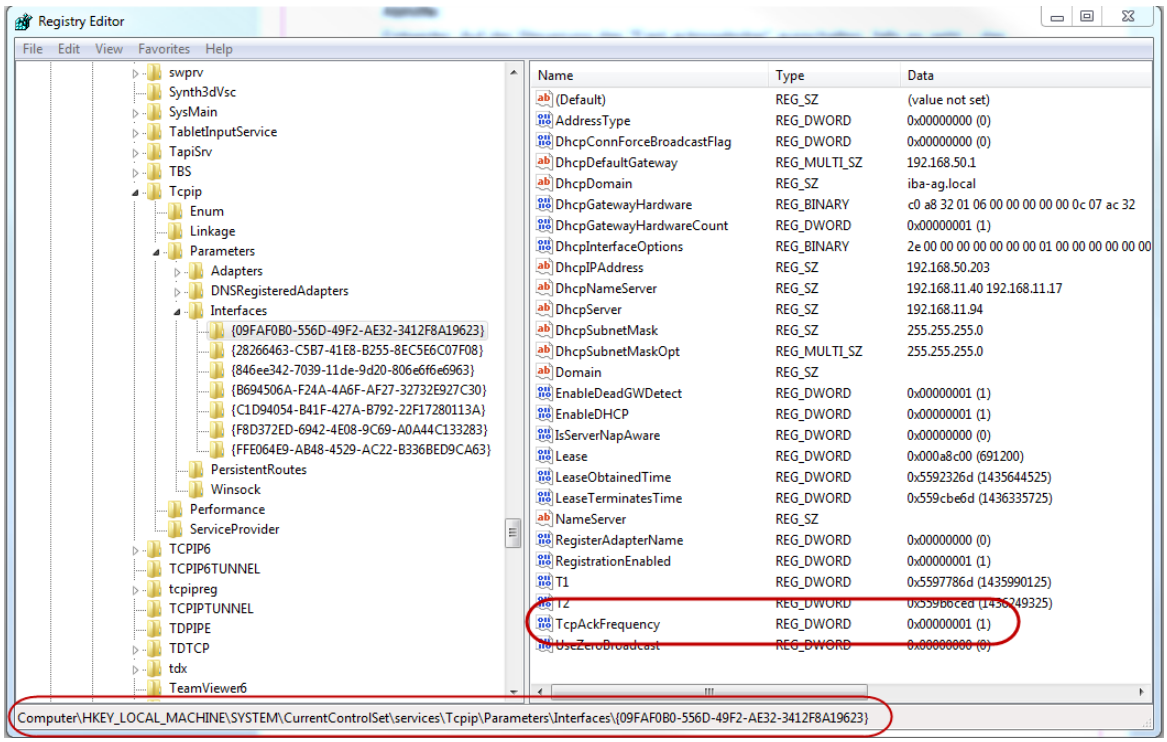
This is set by a parameter in the Windows Registry:

- Under Windows 2000:
  Parameter "TcpDelAckTicks": REG_DWORD = 0;
- Under Windows XP, Windows Server 2003, Windows 7, Windows 8, Windows 10, Windows Server 2012, Windows Server 2016:
  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

www.iba-ag.com.