ibaPDA-Interface-Codesys-Xplorer

PLC-Xplorer Data Interface to CODESYS Systems

Manual

Issue 1.6

Measurement Systems for Industry and Energy

www.iba-ag.com

Manufacturer

iba AG
Koenigswarterstrasse 44
90762 Fuerth
Germany

Contacts

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

The content of this publication has been checked for compliance with the described hardware and software. Nevertheless, discrepancies cannot be ruled out, and we do not provide guarantee for complete conformity. However, the information furnished in this publication is updated regularly. Required corrections are contained in the following regulations or can be downloaded on the Internet.

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

Version	Date	Revision	Author	Version SW
1.6	01-2023	Diagnostic modules	RM/IP	8.1.0

Content

1 About this Manual

This document describes the function and application of the software interface

ibaPDA-Interface-Codesys-Xplorer

This documentation is a supplement to the *ibaPDA* manual. Information about all the other characteristics and functions of *ibaPDA* can be found in the *ibaPDA* manual or in the online help.

1.1 Target group and previous knowledge

This documentation addresses qualified professionals, who are familiar with handling electrical and electronic modules as well as communication and measurement technology. A person is regarded as a professional if he/she is capable of assessing the work assigned to him/her and recognizing possible risks on the basis of his/her specialist training, knowledge and experience and knowledge of the standard regulations.

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

- Windows operating system
- Knowledge of configuration and operation of the relevant control system

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>
Filenames, paths	`Filename`, `Path` Example: `Test.docx`

1.3 Used symbols

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

Danger!

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

- Observe the specified measures.

Warning!

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

- Observe the specified measures.

Caution!

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

- Observe the specified measures

Note

ℹ 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 Codesys-Xplorer data interface.

- *ibaPDA* v8.1.0 or higher
- Basic license for *ibaPDA* + license for *ibaPDA-Interface-PLC-Xplorer* or *ibaPDA-Interface-Codesys-Xplorer*
- With more than 16 connections you need additional *one-step-up-Interface-Codesys-Xplorer* licenses for each additional 16 connections.

Note

The *ibaPDA-Interface-PLC-Xplorer* license contains, among others, the license for this interface.

- Controller (PLC) with CODESYS V2 or CODESYS V3.
  Systems tested so far:
  - 3S CODESYS SP PLC WinNT V2.4
  - 3S CODESYS SP RTE
  - 3S CODESYS Control Win V3
  - ABB AC500 CPU PM554-TP-ETH
  - Danieli HiPAC
  - ELAU Standard CPU
  - ELAU PacDrive C600
  - ifm CR1051
  - Schneider Electric LMC 101C
  - Schneider Electric M258
  - WAGO 750-841
  - WAGO 750-880
  - WAGO 750-881

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

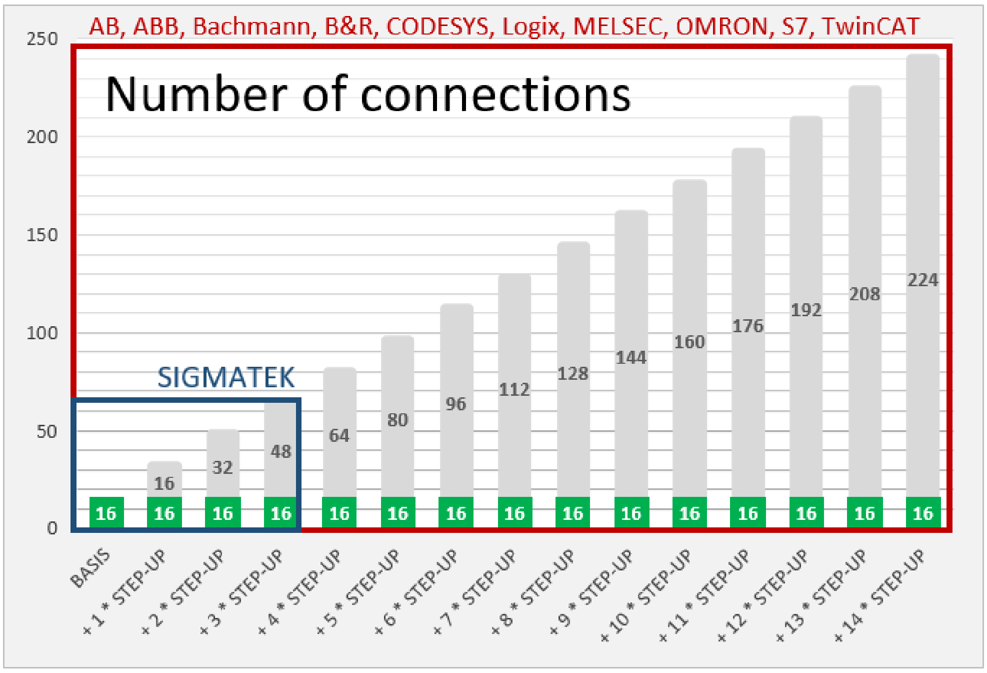
License information

Order no.	Product name	Description
31.001042	ibaPDA-Interface-PLC-Xplorer	Extension license for an ibaPDA system adding available Xplorer data interface (full specification under www.iba-ag.com).
31.000002	ibaPDA-Interface-Codesys-Xplorer	Extension license for an ibaPDA system adding the data interface Codesys-Xplorer (interface to CODESYS)
31.100002	one-step-up-Interface-Codesys-Xplorer	Extension license for 16 further Codesys-Xplorer-connections, a maximum of 14 extension licenses is permissible

Table 1: Available Codesys-Xplorer licenses

Note

If you intend to use more than 16 data links per interface, you can purchase the *one-step-up-...* extension licenses separately for each interface. Up to 16 further connections to PLCs can be established on each *one-step-up*-license. Up to 240 connections can be configured and used per data interface with the multiple purchase or multiple release of these licenses (up to 15 in total). Exception of SIGMATEK: Here, only up to 4 licenses (64 connections) can be activated.



Please note, that a regular *ibaPDA* license (at least *ibaPDA-V7-64*) is the precondition for the upgrades.

You have to take into consideration the limitation of the number of signals by the *ibaPDA* base license.

3 PLC-Xplorer data interface

3.1 General information

The interface *ibaPDA-Interface-Codesys-Xplorer* is suitable for acquiring measured data from a CODESYS controller by TCP/IP via standard NICs. Access thereby is transparent for the controller. A separate configuring and programming of the control system is not necessary.

The selection of the signals to be measured is hereby comfortably carried out by means of symbolic names supported by the *ibaPDA* symbol browser. This enables the access to all defined symbols of the linked CODESYS project.

3.2 System topologies

The connections to the controllers can be established via standard Ethernet interfaces of the computer.

For the operation, no further software is necessary.

Note

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

3.3 Configuration and engineering CODESYS

In general no particular configuration and programming is required on the controller side. In particular, it is not necessary to call any program modules.

The access to the data of the controller is carried out via symbolic addresses. These are imported from the controller or from the project file.

For the creation of the symbols several actions must be carried out.

Refer to examples, ➚ *Appendix*, page 30.

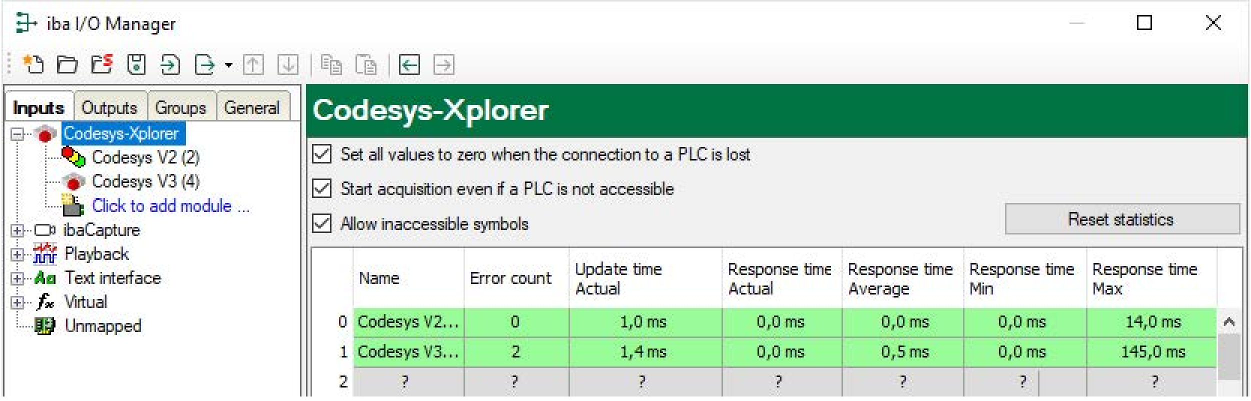
3.4 Configuration and engineering ibaPDA

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

3.4.1 Interface settings

If the Xplorer interface is selected in the tree, you can see an overview of diagnostics information on the configured connections between *ibaPDA* and the controllers.

The interface has the following features and configuration options.



Codesys-Xplorer

☑ Set all values to zero when the connection to a PLC is lost

☑ Start acquisition even if a PLC is not accessible

☑ Allow inaccessible symbols

	Name	Error count	Update time Actual	Response time Actual	Response time Average	Response time Min	Response time Max
0	Codesys V2...	0	1,0 ms	0,0 ms	0,0 ms	0,0 ms	14,0 ms
1	Codesys V3...	2	1,4 ms	0,0 ms	0,5 ms	0,0 ms	145,0 ms
2	?	?	?	?	?	?	?

Set all values to zero when the connection to a PLC is lost
If enabled, all measured values of the PLC are set to zero as soon as the connection is interrupted. If this option is disabled, *ibaPDA* will keep the last valid measured data in memory at the time the connection was interrupted.

Start acquisition even if a PLC is not accessible
If this option is enabled, the acquisition will start even if the CODESYS-CPU is not accessible. A warning is indicated in the validation dialog. If the system has been started without a connection to the CODESYS-CPU, *ibaPDA* will periodically try to connect to the PLC.

Allow inaccessible symbols
Enable this option if you wish to start acquisition even if symbols are not accessible. The inaccessible symbols are issued as warnings in the validation dialog.
This can only occur if the address book is not updated!

Measurement will not start when inaccessible symbols are present if you do not enable this option.

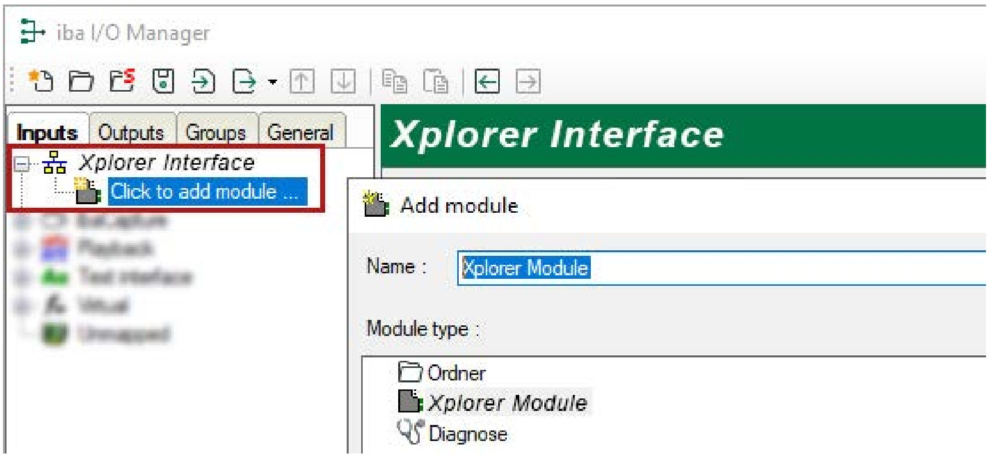
Connection table
For each connection, the table shows the connection status, the current values for the update time (current, real value, average, min. and max.) as well as the data size. In addition, you will find an error counter here for the individual connections during the acquisition.

See ➚ *Connection table*, page 24.

<Reset statistics>
Click this button to reset the calculated times and error counters in the table to 0.

3.4.2 Adding a module

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

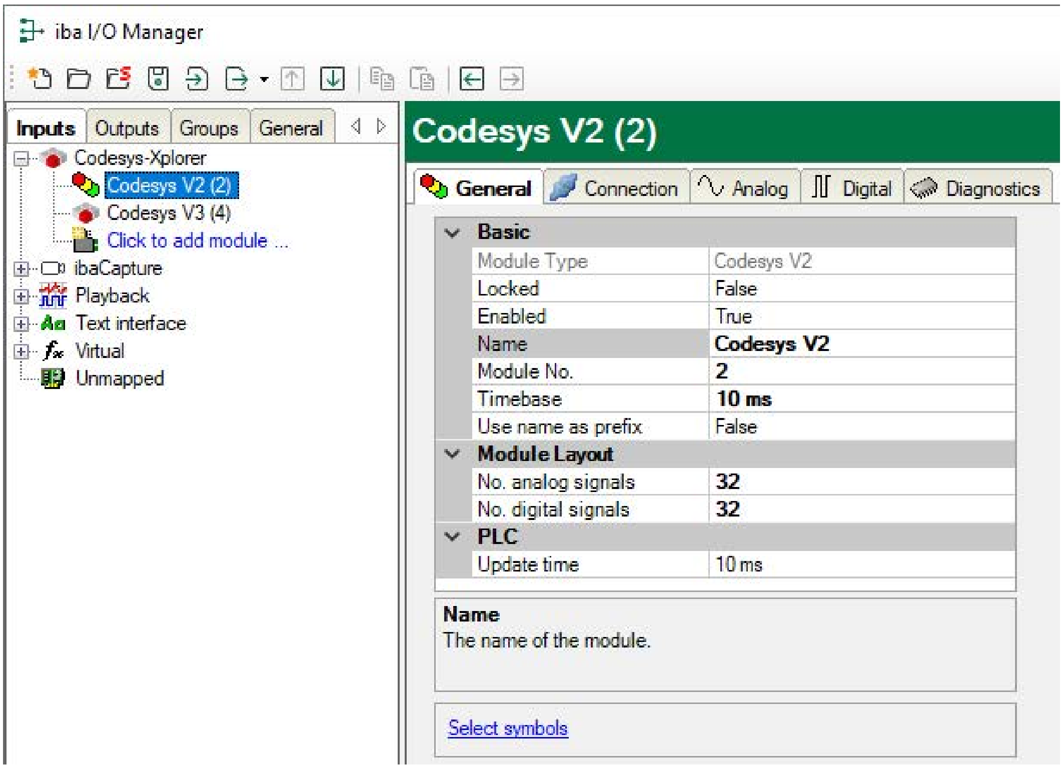


Name : Xplorer Module

3.4.3 General module settings

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

All modules have the following setting options.



Basic	
Module Type	Codesys V2
Locked	False
Enabled	True
Name	**Codesys V2**
Module No.	**2**
Timebase	**10 ms**
Use name as prefix	False
Module Layout	
No. analog signals	**32**
No. digital signals	**32**
PLC	
Update time	10 ms

Name
The name of the module.

Basic settings

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

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

Enabled
Enable the module to record signals.

Name
Here you can enter a name for the module.

Module No.
This internal reference number of the module determines the order of the modules in the signal tree of *ibaPDA* client and *ibaAnalyzer*.

Timebase
All signals of the module are sampled on this time base.

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

Module Layout

No. of analog signals/digital signals

Define the number of configurable analog and digital signals in the signal tables. The default value is 32 for each. The maximum value is 1000. The signal tables are adjusted accordingly.

PLC

Update time
Specifies the reference update time in which the data is requested from the PLC. During measurement, the real current update time may be higher than the specified value if the PLC needs more time to transmit the data. You can check in the connection table how fast the data is actually updated.

<Select symbols>

Click on this link after the connection has been successfully established in order to configure the signals to be measured

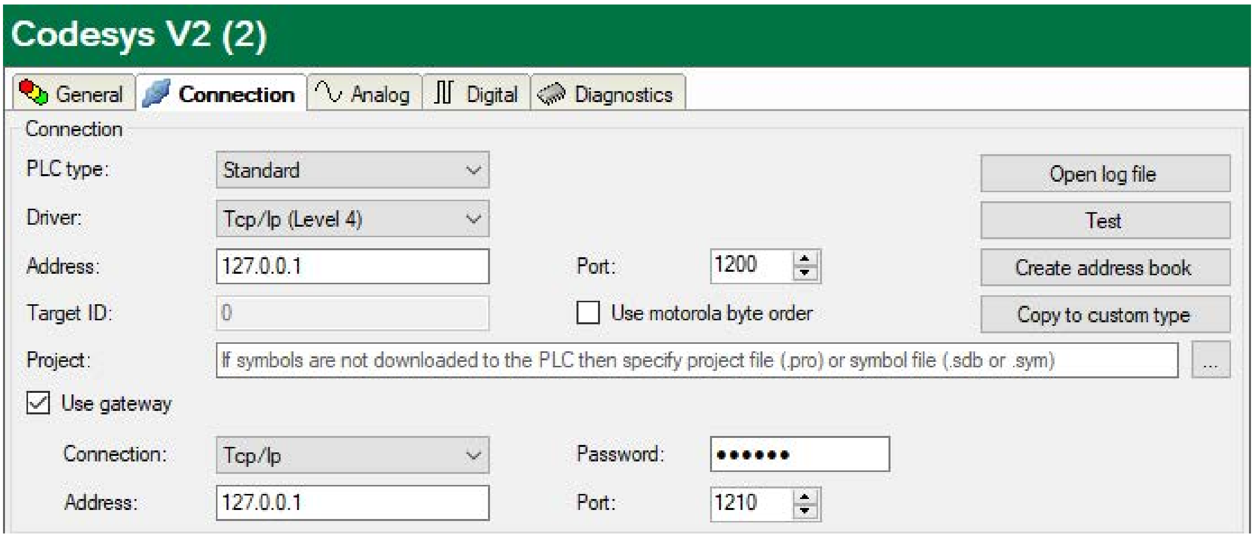
For more information, see ➚ *Signal configuration*, page 17

3.4.4 Connection settings

In the *Connection* tab the connection of the module to the controller is configured.

The connection settings differ according to the CODESYS version.

3.4.4.1 Codesys V2 connection parameters



Connection

PLC type: Standard

Driver: Tcp/Ip (Level 4)

Address: 127.0.0.1 Port: 1200

Target ID: 0 □ Use motorola byte order

Project: If symbols are not downloaded to the PLC then specify project file (.pro) or symbol file (.sdb or .sym)

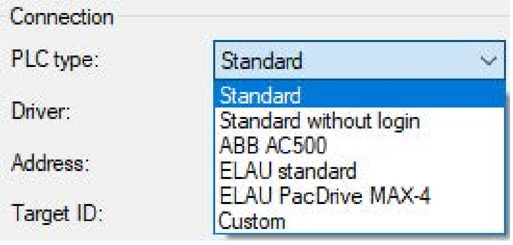
☑ Use gateway

Connection: Tcp/Ip Password: ●●●●●●

Address: 127.0.0.1 Port: 1210

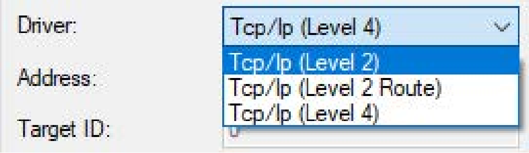
Enter connection parameter

For the connection establishment, the PLC type is to be entered at first and, depending on this, some other parameters. The PLC type must be selected from a dropdown box:



PLC type: Standard

This type can be used for the most of the controllers. It depends on the control system which TCP/IP driver has to be used.



Additionally, enter the IP address, port number (preset is 1200) and the byte order of the controller.
For the driver type "Tcp/Ip (Level 2 Route)" you have to additionally enter the target ID of the controller. Address and port number refer to the control system which functions as router.

PLC type: Standard without application

As standard, except that *ibaPDA* does not try to log in the controller.

PLC type: ABB AC500

Parameter for the connection with the ABB AC500 controller. The set is identical with the standard parameters. Except: the port is preset with 1201 and the Motorola byte order is applied.

PLC type: ELAU standard
Parameter for the ELAU standard controller.
The driver type and the byte order are internally defined.
The port is preset with 5000.

PLC type: ELAU PacDrive MAX-4
As ELAU standard, the difference is that the entry of the hardware version is necessary (preset: 1100).

PLC type: user
This PLC type can be used to test a connection to a new PLC.

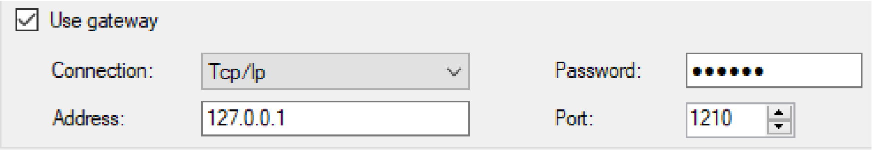
For further information, please see ↗ *Connection setup*, page 16

Use gateway

If the controller and the *ibaPDA* computer are not in the same subnet but are only able to communicate via a gateway, please activate this option. The CODESYS gateway server may be an example of such a gateway which can be accessed locally or remote.

Additional input fields appear:

- Check box "Connection" with the alternatives TCP/IP and local
- In case of “Tcp/Ip”, IP address, port number, and if required, a password of the gateway server have to be entered.
- In case of “Local” a password has to be entered, if required. The fields for address and port are deactivated.



☑ Use gateway

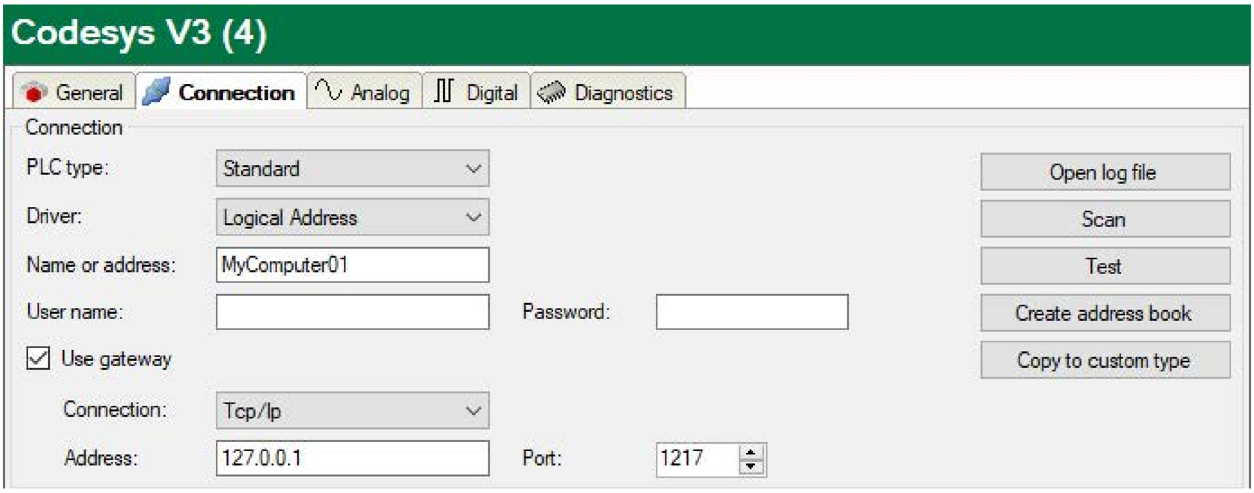
Connection: Tcp/Ip

Password: ••••••

Address: 127.0.0.1

Port: 1210

3.4.4.2 Codesys V3 connection parameters



Codesys V3 (4)

Connection

PLC type: Standard

Driver: Logical Address

Name or address: MyComputer01

User name:

Password:

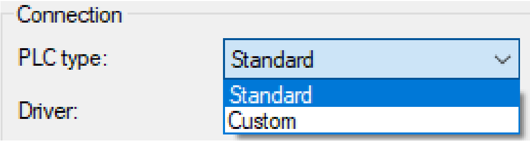
☑ Use gateway

Connection: Tcp/Ip

Address: 127.0.0.1

Port: 1217

For CODESYS V3, only the PLC types “Standard” and “Custom” are available.



PLC type: Standard

This type can be used for the most of the controllers.

For the driver “Logical Address”, you can either enter the device name or the device address.

Note

You can search for all accessible controllers by clicking on the <Scan> button. The controllers with device name and device address are displayed as result.

For the driver “CmpBlkDrvTcp”, you have to enter the IP address of the controller in the usual form, e.g. "192.168.21.121"; it is additionally required to indicate the port number 11740.

PLC type: Custom

The “Custom” type allows to define the connection parameters by itself. The procedure is the same as under CODESYS V2.

For further information, please see ↗ *Connection setup*, page 16

User name/Password:

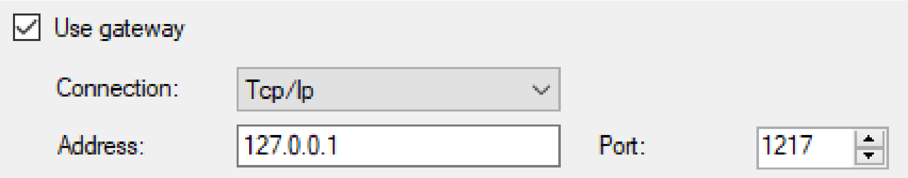
If an user name and a password have been assigned on the controller side, they must be entered here in order to obtain information about the controller and to be able to read out values. Otherwise, no entry is necessary.

Use gateway

If the controller and the *ibaPDA* computer are not in the same network but are only able to communicate via a gateway, please enable this option. The CODESYS gateway server may be an example of such a gateway which can be accessed locally or remote.

Additional input fields appear:

- Check box "Connection" with the alternatives TCP/IP and local
- In case of “Tcp/Ip”, IP address, port number, and if required, a password of the gateway server have to be entered.
- In case of “Local” a password has to be entered, if required. The fields for address and port are deactivated.



☑ Use gateway

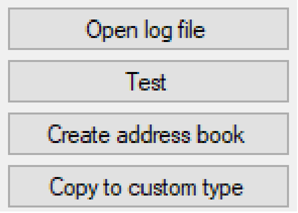
Connection: Tcp/Ip

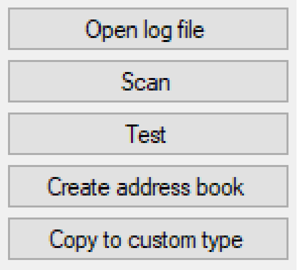
Address: 127.0.0.1

Port: 1217

3.4.4.3 Connection setup

The following buttons are available for establishing and testing of the connection to the controller (left fig.: CODESYS V2; right fig.: CODESYS V3):



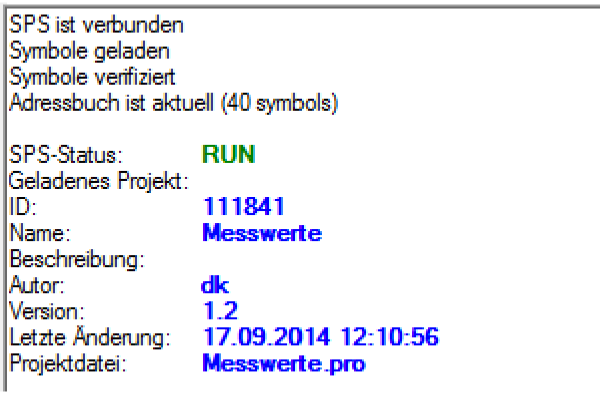


Open log file
The log file entries created during the establishment of the connection are shown in the standard editor.

Scan (only for CODESYS V3)
The network is scanned for installed CODESYS V3 applications.
Some characteristics of the controller are shown. A connection will be established by clicking on the hyperlinks *Device name* or *Device address* (identical with button <Test>).

Test
With the set link parameters (see below), it will be tried to establish the connection to the controller. In case of success,

- the data of the controller will be shown,
- the link settings will be stored in a file `'modulename'.INI` in the folder "...\iba\ibaPDA\Server\Codesys" ,
- the symbols, if they exist in the controller, are loaded and created as address book in the folder "...\iba\ibaPDA\Server\Codesys\Addressbooks".



```
SPS ist verbunden
Symbole geladen
Symbole verifiziert
Adressbuch ist aktuell (40 symbols)

SPS-Status:          RUN
Geladenes Projekt:
ID:                  111841
Name:                Messwerte
Beschreibung:
Autor:               dk
Version:             1.2
Letzte Änderung:     17.09.2014 12:10:56
Projektdatei:        Messwerte.pro
```

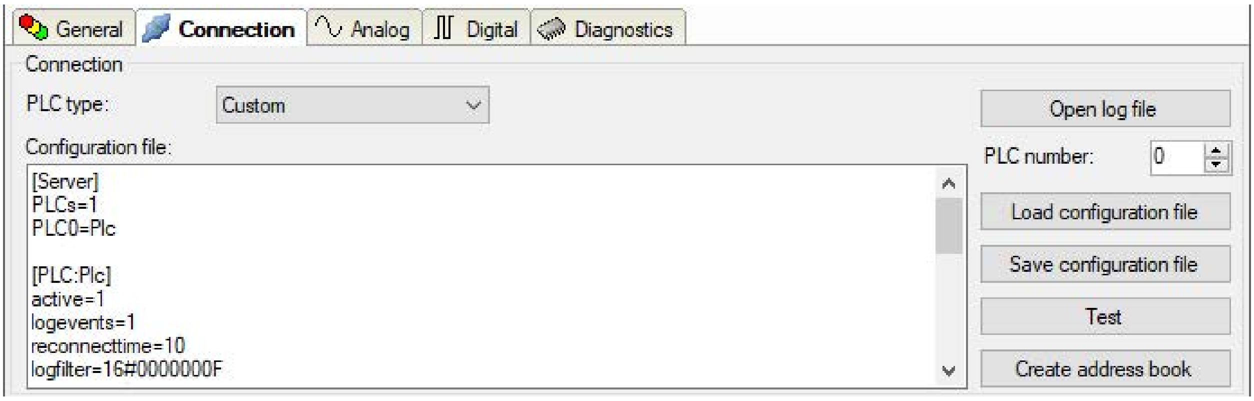
Create address book
The symbols, if they exist in the controller, are loaded and created as address book in the folder "...\iba\ibaPDA\Server\Codesys\Addressbooks".

Only for CODESYS V2:
If the symbols do not exist in the controller, the project file (.pro) or symbol file (.sdb or .sym) must be entered in the field “Project".

Copy to custom type

A copy of the current connection settings is shown as user type. You can then change parameters, to create a connection to a new PLC type.

Using the button <Save configuration file> you can save the modified parameters under any desired name and you can reload it using the button <Load configuration file>.



PLC type: Custom

PLC number: 0

Configuration file:

```
[Server]
PLCs=1
PLC0=Plc

[PLC:Plc]
active=1
logevents=1
reconnecttime=10
logfilter=16#0000000F
```

3.4.5 Signal configuration

In the *Analog* or *Digital* tab you configure the signals to be measured. In the *General* tab under *Module Layout* you define the length of the signal tables or the number of signals per table.

Note

Observe the maximum number of signals permitted by your license.

Note

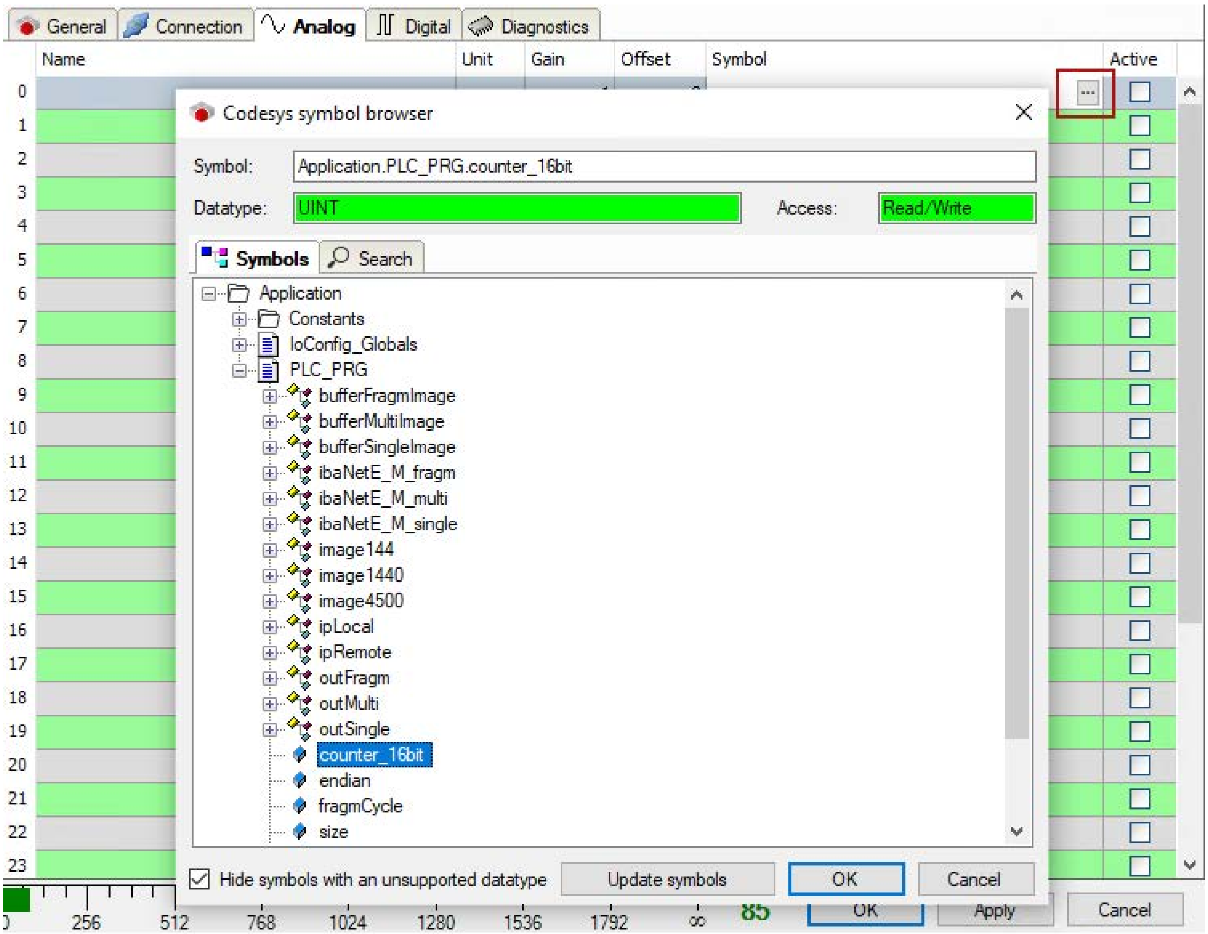
Take into consideration that the number of signals, which are read by a CPU, influences the minimum achievable read cycle. The more signals recorded, the slower the reachable reading cycle.

Selection of the signals to be measured

You have two options to select the signals to be measured:

1. In the *Analog* and/or *Digital* tab in the *Symbol* column:

   By clicking on the *Symbol* field, the icon ⋯ will be shown. A click on the icon opens the symbol browser:



Symbol: Application.PLC_PRG.counter_16bit

Datatype: UINT

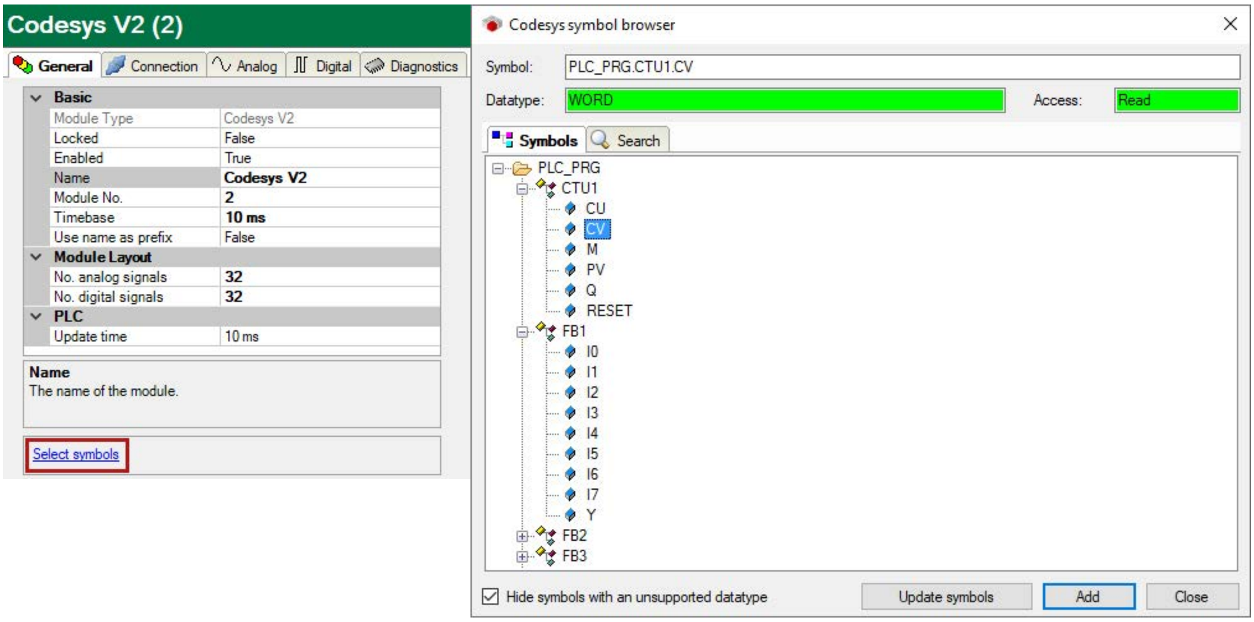
Access: Read/Write

☑ Hide symbols with an unsupported datatype

- Only the analog and/or digital values are visible, depending on in which tab the browser is activated.
- By double-clicking or <OK> you enter the marked variable in the signal table and the browser is closed.

2. By means of the *Select symbols* hyperlink in the tab *General*.

   A click on the hyperlink opens the symbol browser:



Codesys V2 (2)

Basic	
Module Type	Codesys V2
Locked	False
Enabled	True
Name	**Codesys V2**
Module No.	**2**
Timebase	**10 ms**
Use name as prefix	False
Module Layout	
No. analog signals	**32**
No. digital signals	**32**
PLC	
Update time	10 ms

Name
The name of the module.

Codesys symbol browser

Symbol: PLC_PRG.CTU1.CV

Datatype: WORD

Access: Read

- PLC_PRG
  - CTU1
    - CU
    - CV
    - M
    - PV
    - Q
    - RESET
  - FB1
    - I0
    - I1
    - I2
    - I3
    - I4
    - I5
    - I6
    - I7
    - Y
  - FB2
  - FB3

☑ Hide symbols with an unsupported datatype

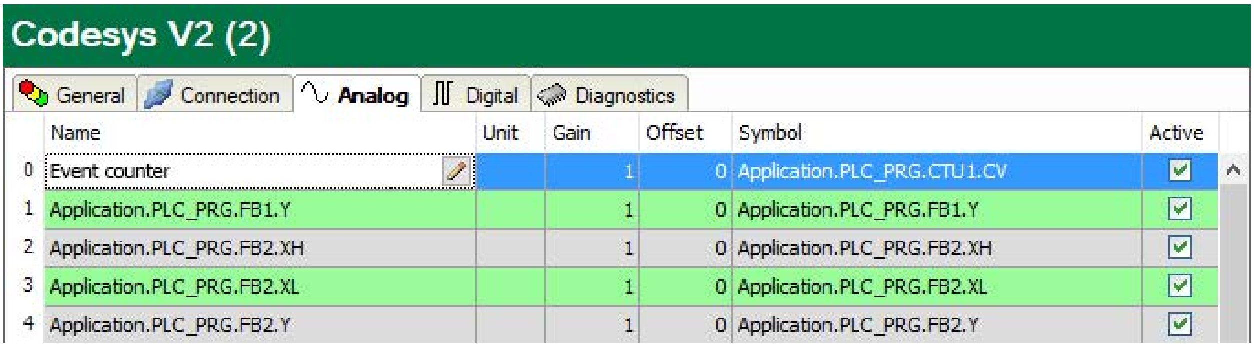
- Analog values as well as digital values are visible.
- By double clicking or <Add> you enter the marked variable in the current line of the analog and/or digital table.
  The marking in the symbol browser goes to the next variable and the marking in the signal table goes to the next line.
- Now, the browser is closed by <Close>.

Note

You can hide all unsupported datatypes by checking the "Hide symbols with an unsupported datatype" checkbox. Not supported are the types LINT and ULINT as well as some non-numerical datatypes, e. g. DATE. The STRING datatype is supported.

Description of the tables

The variables are entered in the *Symbol* column of the signal table with the designation “Application.Program.Module.Connection". Simultaneously, the designation is also entered in the column *Name.* You can edit subsequently and eventually indicate comments, units and conversion factors.



Codesys V2 (2)

	Name	Unit	Gain	Offset	Symbol	Active
0	Event counter		1	0	Application.PLC_PRG.CTU1.CV	☑
1	Application.PLC_PRG.FB1.Y		1	0	Application.PLC_PRG.FB1.Y	☑
2	Application.PLC_PRG.FB2.XH		1	0	Application.PLC_PRG.FB2.XH	☑
3	Application.PLC_PRG.FB2.XL		1	0	Application.PLC_PRG.FB2.XL	☑
4	Application.PLC_PRG.FB2.Y		1	0	Application.PLC_PRG.FB2.Y	☑

You can configure the columns of the table as necessary, e.g. skip the columns *Gain* and *Offset.* See also the description in the *ibaPDA* manual.

Having filled the signal tables, you start the acquisition with <Apply> or <OK>. After a successful start, you can verify the current values within the I/O manager in the *Diagnostics* tab (see ➚ *Module diagnostics*, page 20).

After closing the I/O manager, you can display the current values in different graphical representations, e. g. in trend graphs or in numerical displays.

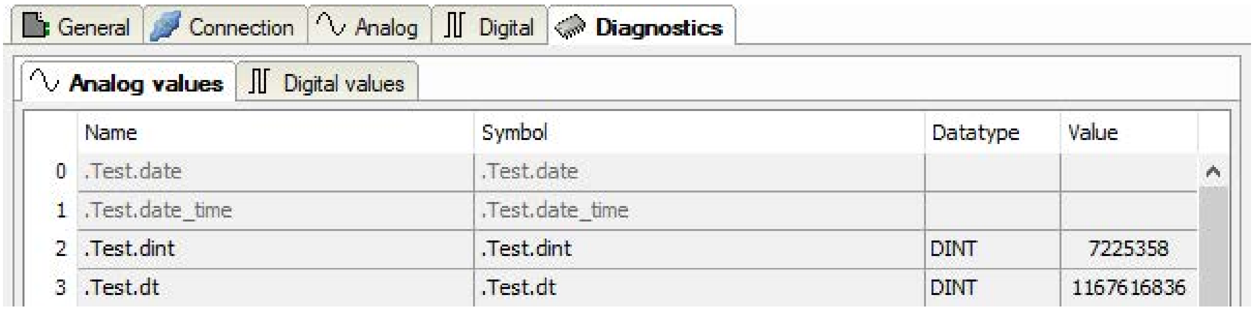
You can find further explanatory notes in the *ibaPDA* manual.

Note

For further processing of the text signals (datatype STRING) or splitting these into other text signals use a text splitter module under the *Virtual* interface.

3.4.6 Module diagnostics

After applying the configuration the actual values of the analog and digital signals are displayed in the *Diagnostics* tab of the relevant module.



	Name	Symbol	Datatype	Value
0	.Test.date	.Test.date		
1	.Test.date_time	.Test.date_time		
2	.Test.dint	.Test.dint	DINT	7225358
3	.Test.dt	.Test.dt	DINT	1167616836

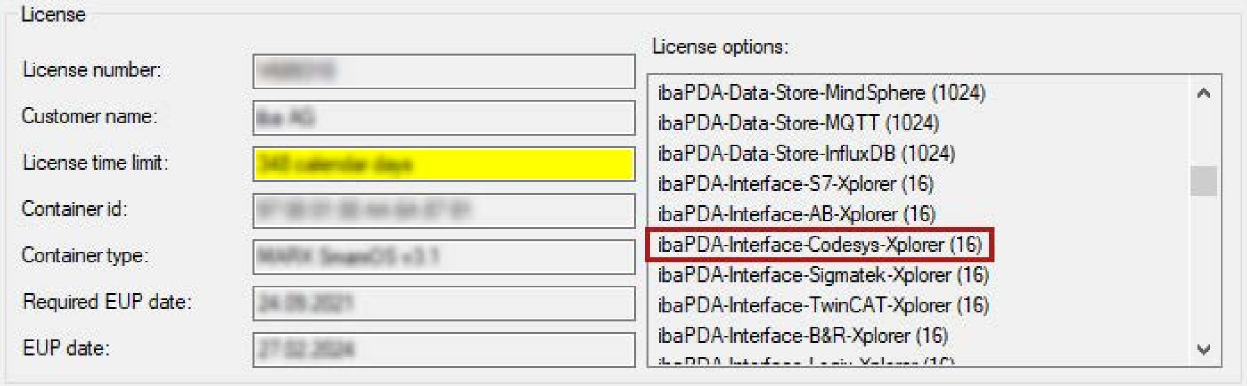
Inactive signals are grayed out.

4 Diagnostics

4.1 License

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

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



License

License number:	[redacted]
Customer name:	[redacted]
License time limit:	[redacted]
Container id:	[redacted]
Container type:	[redacted]
Required EUP date:	[redacted]
EUP date:	[redacted]

License options:

- ibaPDA-Data-Store-MindSphere (1024)
- ibaPDA-Data-Store-MQTT (1024)
- ibaPDA-Data-Store-InfluxDB (1024)
- ibaPDA-Interface-S7-Xplorer (16)
- ibaPDA-Interface-AB-Xplorer (16)
- ibaPDA-Interface-Codesys-Xplorer (16)
- ibaPDA-Interface-Sigmatek-Xplorer (16)
- ibaPDA-Interface-TwinCAT-Xplorer (16)
- ibaPDA-Interface-B&R-Xplorer (16)
- ibaPDA-Interface-Logix-Xplorer (16)

4.2 Visibility of the interface

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

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

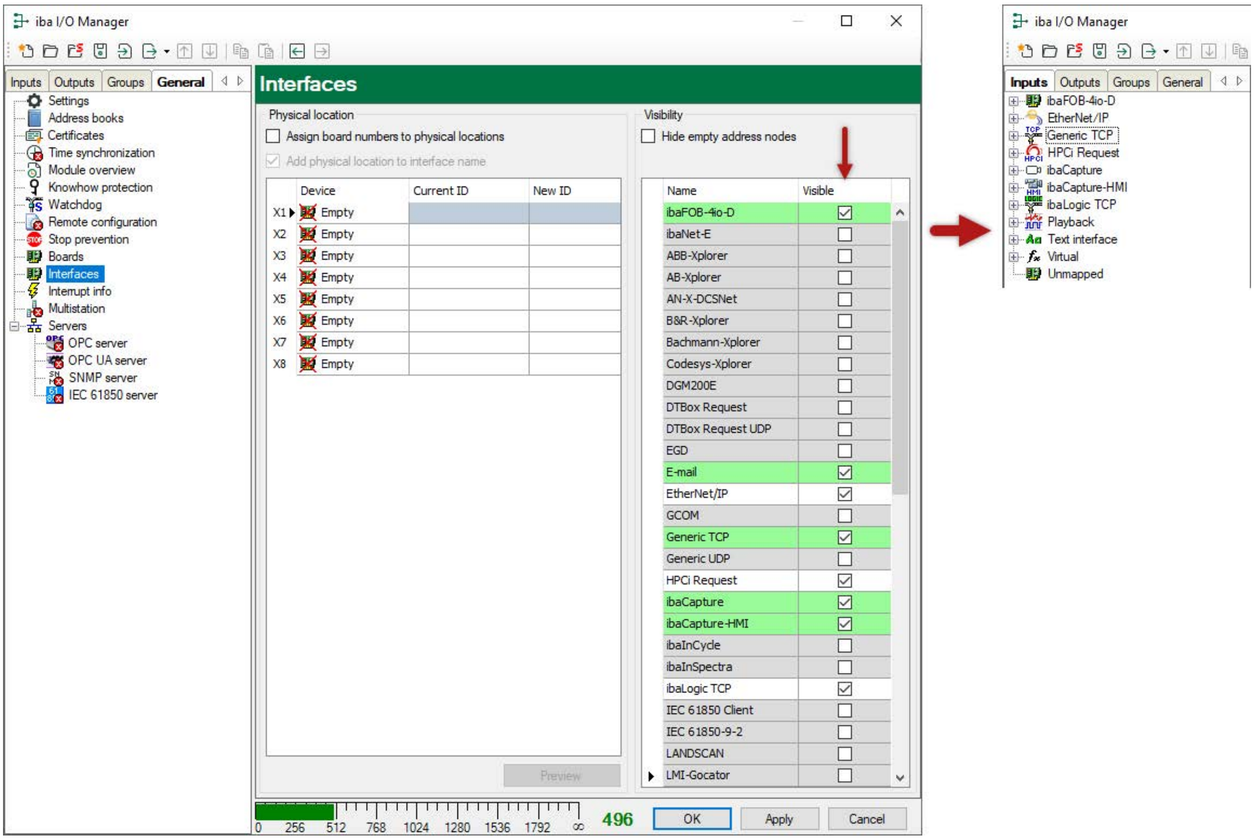
Visibility

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

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

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

Selected interfaces are visible, the others are hidden:



Physical location

□ Assign board numbers to physical locations

☑ Add physical location to interface name

	Device	Current ID	New ID
X1	Empty		
X2	Empty		
X3	Empty		
X4	Empty		
X5	Empty		
X6	Empty		
X7	Empty		
X8	Empty		

Visibility

□ Hide empty address nodes

Name	Visible
ibaFOB-4io-D	☑
ibaNet-E	□
ABB-Xplorer	□
AB-Xplorer	□
AN-X-DCSNet	□
B&R-Xplorer	□
Bachmann-Xplorer	□
Codesys-Xplorer	□
DGM200E	□
DTBox Request	□
DTBox Request UDP	□
EGD	□
E-mail	☑
EtherNet/IP	☑
GCOM	□
Generic TCP	☑
Generic UDP	□
HPCi Request	☑
ibaCapture	☑
ibaCapture-HMI	☑
ibaInCycle	□
ibaInSpectra	□
ibaLogic TCP	☑
IEC 61850 Client	□
IEC 61850-9-2	□
LANDSCAN	□
LMI-Gocator	□

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.

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

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

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

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

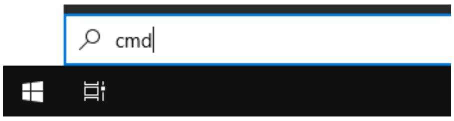
Examples:

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

4.4 Connection diagnostics with PING

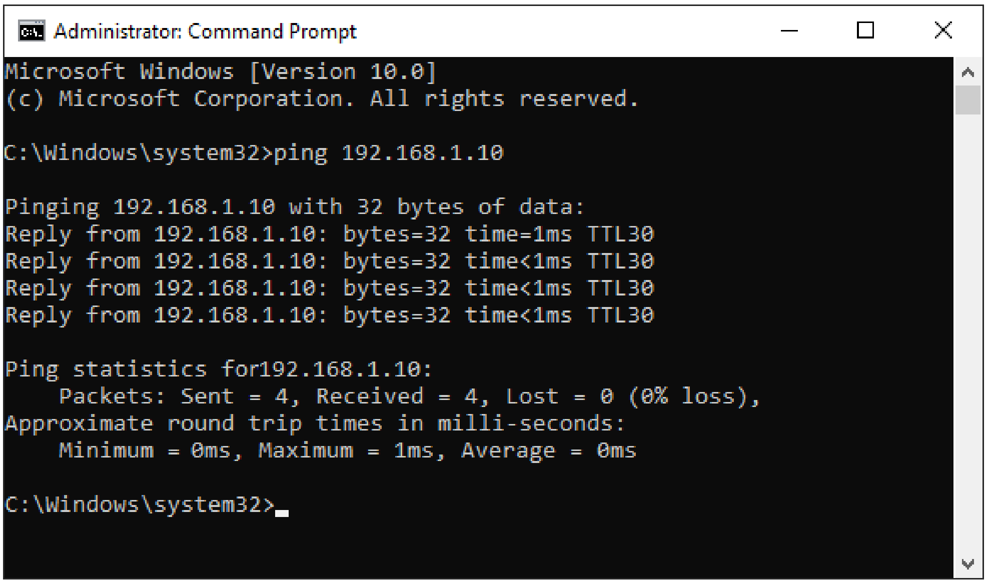
PING is a system command with which you can check if a certain communication partner can be reached in an IP network.

1. Open a Windows command prompt.



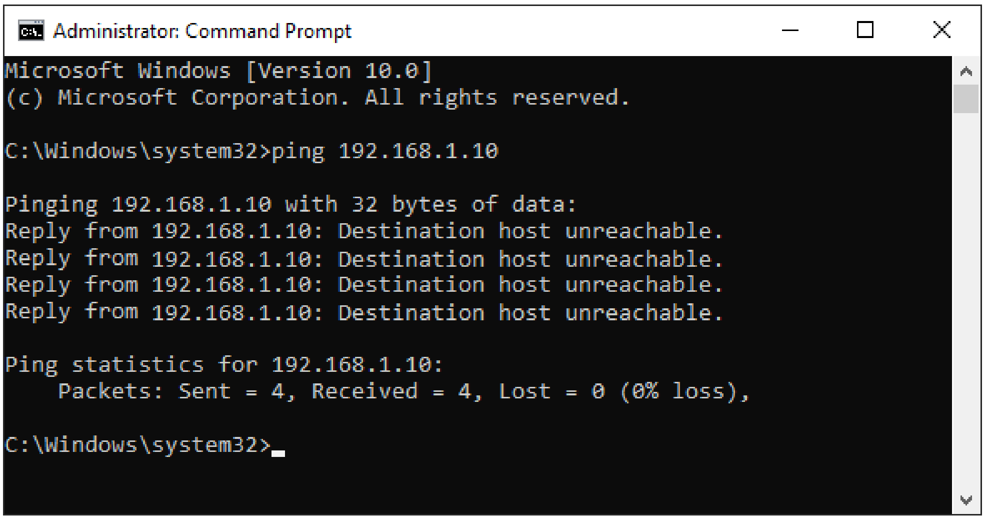
2. Enter the command "ping" followed by the IP address of the communication partner and press <ENTER>.

→ With an existing connection you receive several replies.



```
Microsoft Windows [Version 10.0]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>ping 192.168.1.10

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

Ping statistics for192.168.1.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\Windows\system32>_
```

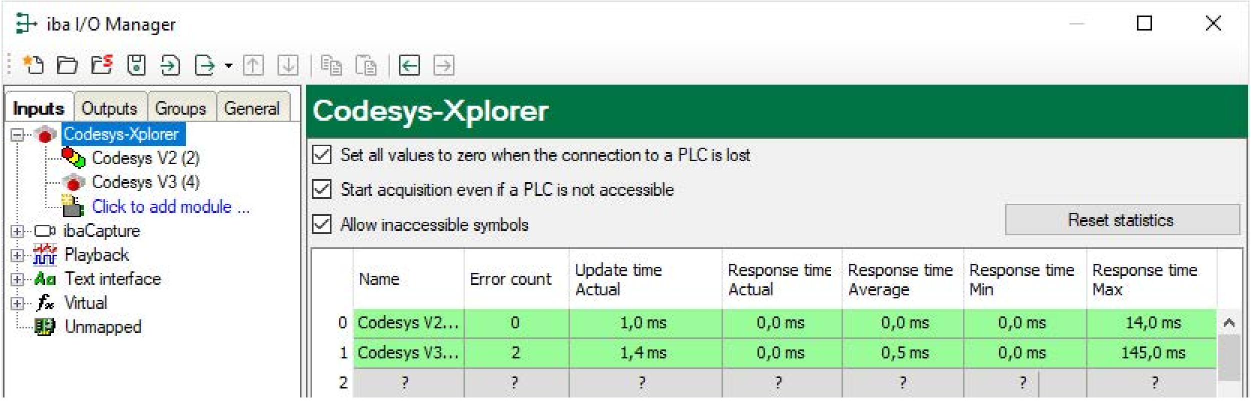
→ With no existing connection you receive error messages.



```
Microsoft Windows [Version 10.0]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>ping 192.168.1.10

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

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

C:\Windows\system32>_
```

4.5 Connection table

For every Ethernet-based interface, there is a table available in the I/O Manager which shows the status of each connection. Each line represents one connection. The following figure shows, as an example, the connection table of the Codesys-Xplorer interface:



Codesys-Xplorer

☑ Set all values to zero when the connection to a PLC is lost

☑ Start acquisition even if a PLC is not accessible

☑ Allow inaccessible symbols

	Name	Error count	Update time Actual	Response time Actual	Response time Average	Response time Min	Response time Max
0	Codesys V2...	0	1,0 ms	0,0 ms	0,0 ms	0,0 ms	14,0 ms
1	Codesys V3...	2	1,4 ms	0,0 ms	0,5 ms	0,0 ms	145,0 ms
2	?	?	?	?	?	?	?

The connected target systems (controllers) are identified by their name or IP address in the first (left) column.

Depending on the interface type the table shows error counters, read counters and/or data sizes, as well as the cycle times, refresh times and/or update times of the different connections during the data acquisition. Click the <Reset statistics> button to reset the error counters and the calculation of the response times.

Additional information is provided by the background color of the table rows:

Color	Meaning
Green	The connection is OK and the data are read.
Yellow	The connection is OK, however the data update is slower than the configured update time.
Red	The connection has failed.
Gray	No connection configured.

4.6 Diagnostic modules

Diagnostic modules are available for most Ethernet based interfaces and Xplorer interfaces. Using a diagnostic module, information from the diagnostic displays (e. g. diagnostic tabs and connection tables of an interface) can be acquired as signals.

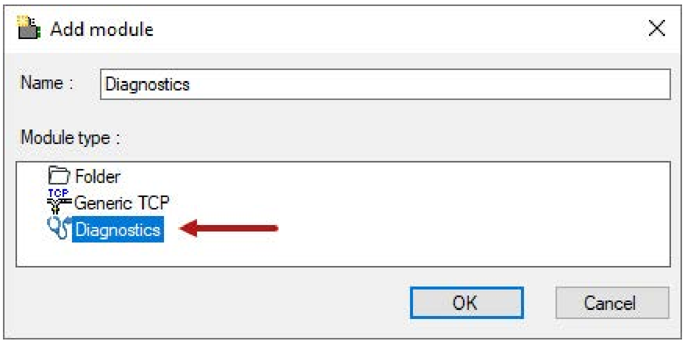
A diagnostic module is always assigned to a data acquisition module of the same interface and supplies its connection information. By using a diagnostic module you can record and analyze the diagnostic information continuously in the *ibaPDA* system.

Diagnostic modules do not consume any license connections, since they do not establish their own connection, but refer to another module.

Example for the use of diagnostic modules:

- A notification can be generated, whenever the error counter of a communication connection exceeds a certain value or the connection gets lost.
- In case of a disturbance, the current response times in the telegram traffic may be documented in an incident report.
- The connection status can be visualized in *ibaQPanel.*
- You can forward diagnostic information via the SNMP server integrated in *ibaPDA* or via OPC DA/UA server to superordinate monitoring systems like network management tools.

In case the diagnostic module is available for an interface, a "Diagnostics" module type is shown in the "Add module" dialog (example: Generic TCP).



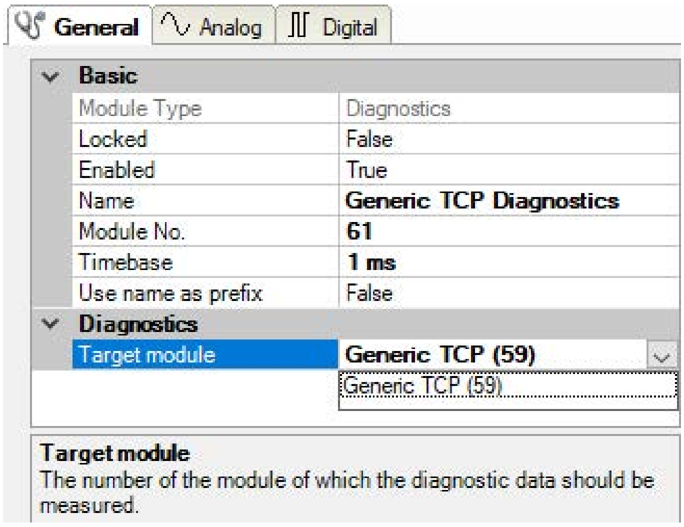
Add module

Name : Diagnostics

Module type :

Module settings diagnostic module

For a diagnostic module, you can make the following settings (example: Generic TCP):



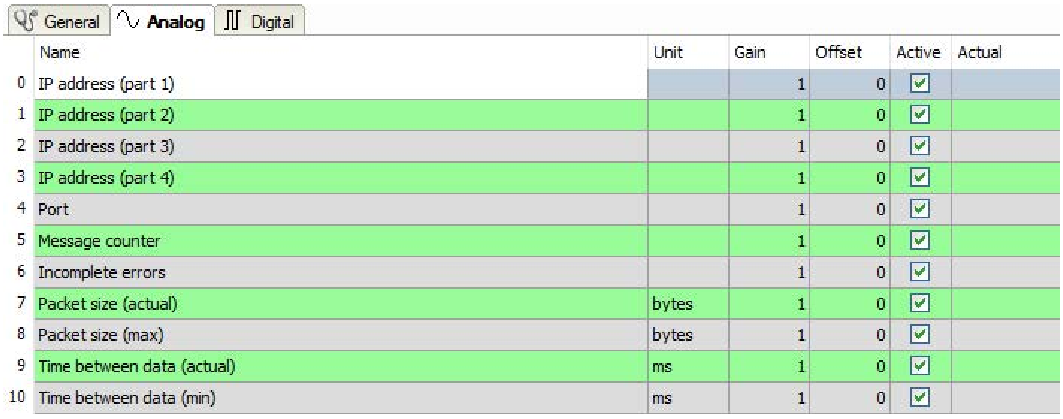
Basic	
Module Type	Diagnostics
Locked	False
Enabled	True
Name	**Generic TCP Diagnostics**
Module No.	**61**
Timebase	**1 ms**
Use name as prefix	False
Diagnostics	
Target module	**Generic TCP (59)**
	Generic TCP (59)

Target module
The number of the module of which the diagnostic data should be measured.

The basic settings of a diagnostic module equal those of other modules.

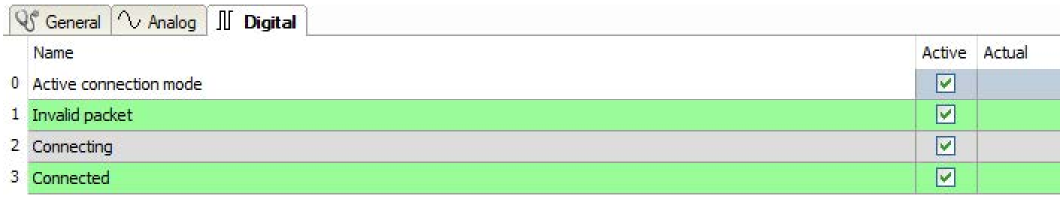
There is only one setting which is specific for the diagnostic module: the target module.

By selecting the target module, you assign the diagnostic module to the module on which you want to acquire information about the connection. You can select the supported modules of this interface in the drop down list of the setting. You can assign exactly one data acquisition module to each diagnostic module. When having selected a module, the available diagnostic signals are immediately added to the *Analog* and *Digital* tabs. It depends on the type of interface, which signals exactly are added. The following example lists the analog values of a diagnostic module for a Generic TCP module.



	Name	Unit	Gain	Offset	Active	Actual
0	IP address (part 1)		1	0	☑	
1	IP address (part 2)		1	0	☑	
2	IP address (part 3)		1	0	☑	
3	IP address (part 4)		1	0	☑	
4	Port		1	0	☑	
5	Message counter		1	0	☑	
6	Incomplete errors		1	0	☑	
7	Packet size (actual)	bytes	1	0	☑	
8	Packet size (max)	bytes	1	0	☑	
9	Time between data (actual)	ms	1	0	☑	
10	Time between data (min)	ms	1	0	☑	

For example, the IP (v4) address of a Generic TCP module (see fig. above) will always be split into 4 parts derived from the dot-decimal notation, for better reading. Also other values are being determined, as there are port number, counters for telegrams and errors, data sizes and telegram cycle times. The following example lists the digital values of a diagnostic module for a Generic TCP module.



	Name	Active	Actual
0	Active connection mode	☑	
1	Invalid packet	☑	
2	Connecting	☑	
3	Connected	☑	

Diagnostic signals

Depending on the interface type, the following signals are available:

Signal name	Description
Buffer file size (actual/avg/max)	Size of the file for buffering statements
Buffer memory size (actual/avg/max)	Size of the memory used by buffered statements
Buffered statements	Number of unprocessed statements in the buffer
Buffered statements lost	Number of buffered but unprocessed and lost statements
Connected	Connection is established
Connected (in)	A valid data connection for the reception (in) is available
Connected (out)	A valid data connection for sending (out) is available
Connecting	Connection being established
Connection attempts (in)	Number of attempts to establish the receive connection (in)
Connection attempts (out)	Number of attempts to establish the send connection (out)
Connection ID O->T	ID of the connection for output data (from the target system to *ibaPDA*). Corresponds to the assembly instance number
Connection ID T->O	ID of the connection for input data (from *ibaPDA* to target system). Corresponds to the assembly instance number
Connection phase (in)	Status of the ibaNet-E data connection for reception (in)
Connection phase (out)	Status of the ibaNet-E data connection for sending (out)
Connections established (in)	Number of currently valid data connections for reception (in)
Connections established (out)	Number of currently valid data connections for sending (out)
Data length	Length of the data message in bytes
Data length O->T	Size of the output message in byte
Data length T->O	Size of the input message in byte
Destination IP address (part 1-4) O->T	4 octets of the IP address of the target system Output data (from target system to *ibaPDA*)
Destination IP address (part 1-4) T->O	4 octets of the IP address of the target system Input data (from *ibaPDA* to target system)
Disconnects (in)	Number of currently interrupted data connections for reception (in)
Disconnects (out)	Number of currently interrupted data connections for sending (out)
Error counter	Communication error counter
Exchange ID	ID of the data exchange
Incomplete errors	Number of incomplete messages
Incorrect message type	Number of received messages with wrong message type
Input data length	Length of data messages with input signals in bytes (*ibaPDA* receives)
Invalid packet	Invalid data packet detected

Signal name	Description
IP address (part 1-4)	4 octets of the IP address of the target system
Keepalive counter	Number of KeepAlive messages received by the OPC UA Server
Lost images	Number of lost images (in) that were not received even after a retransmission
Lost Profiles	Number of incomplete/incorrect profiles
Message counter	Number of messages received
Messages per cycle	Number of messages in the cycle of the update time
Messages received since configuration	Number of received data telegrams (in) since start of acquisition
Messages received since connection start	Number of received data telegrams (in) since the start of the last connection setup. Reset with each connection loss.
Messages sent since configuration	Number of sent data telegrams (out) since start of acquisition
Messages sent since connection start	Number of sent data telegrams (out) since the start of the last connection setup. Reset with each connection loss.
Multicast join error	Number of multicast login errors
Number of request commands	Counter for request messages from *ibaPDA* to the PLC/CPU
Output data length	Length of the data messages with output signals in bytes (*ibaPDA* sends)
Packet size (actual)	Size of the currently received message
Packet size (max)	Size of the largest received message
Ping time (actual)	Response time for a ping telegram
Port	Port number for communication
Producer ID (part 1-4)	Producer ID as 4 byte unsigned integer
Profile Count	Number of completely recorded profiles
Read counter	Number of read accesses/data requests
Receive counter	Number of messages received
Response time (actual/average/max/min)	Response time is the time between measured value request from *ibaPDA* and response from the PLC or reception of the data. Actual: current value Average/max/min: static values of the update time since the last start of the acquisition or reset of the counters.
Retransmission requests	Number of data messages requested again if lost or delayed
Rows (last)	Number of resulting rows by the last SQL query (within the configured range of result rows)
Rows (maximum)	Maximum number of resulting rows by any SQL query since the last start of acquisition (possible maximum equals the configured number of result rows)

Signal name	Description
Send counter	Number of send messages
Sequence errors	Number of sequence errors
Source IP address (part 1-4) O->T	4 octets of the IP address of the target system Output data (from target system to *ibaPDA*)
Source IP address (part 1-4) T->O	4 octets of the IP address of the target system Input data (from *ibaPDA* to target system)
Statements processed	Number of executed statements since last start of acquisition
Synchronization	Device is synchronized for isochronous acquisition
Time between data (actual/ max/min)	Time between two correctly received messages Actual: between the last two messages Max/min: statistical values since start of acquisition or reset of counters
Time offset (actual)	Measured time difference of synchronicity between *ibaPDA* and the ibaNet-E device
Topics Defined	Number of defined topics
Topics Updated	Number of updated topics
Unknown sensor	Number of unknown sensors
Update time (actual/average/ configured/max/min)	Specifies the update time in which the data is to be retrieved from the PLC, the CPU or from the server (configured). Default is equal to the parameter "Timebase". During the measurement the real actual update time (actual) can be higher than the set value, if the PLC needs more time to transfer the data. How fast the data is really updated, you can check in the connection table. The minimum achievable update time is influenced by the number of signals. The more signals are acquired, the greater the update time becomes. Average/max/min: static values of the update time since the last start of the acquisition or reset of the counters.
Write counter	Number of successful write accesses
Write lost counter	Number of failed write accesses

5 Appendix

5.1 Link parameter Overview

The following table shows the necessary connection parameters for the to date tested controllers:

Controller	CODESYS version	PLC type, Driver	Port:	Note
3S CODESYS Control Win V3	V3	Standard, Logical Address		
3S CODESYS SP PLCWinNT V2.4	V2.3	Standard TCP/IP (Level 4)	1200	
3S CODESYS SP RTE	V2.3	Standard TCP/IP (Level 4)	1200	
ABB AC500 PM554-TP-ETH	V2.3	ABB AC500 TCP/IP (Level 4)	1201	
Bosch-Rexroth L40	V2.3	Standard TCP/IP (Level 2 Route)	1200	
Bosch-Rexroth Indramat MLC L75	V3	CmpBlkDrvTcp	11740	
Danieli HiPAC	V2.3	Standard TCP/IP (Level 4)	1200	
Eckelmann ENC66c	V2.3	Standard TCP/IP (Level 2)	1200	
ELAU PacDrive C600	V2.3	ELAU standard	5000	PLC Status can’t be read out
ifm CR1051	V2.3	Standard without application TCP/IP (Level 4)	1200	Symbols were not stored in PLC but read directly from SDB file.
Schneider Electric LMC 101C	V3	Standard Logical Address		
Schneider Electric M258	V3	Standard Logical Address		
WAGO 750-841	V2.3	Standard TCP/IP (Level 2)	2455	Symbols were not stored in PLC but read directly from SDB file.
WAGO 750-880	V2.3	Standard TCP/IP (Level 2)	2455	Symbols were not stored in PLC but read directly from SDB file.

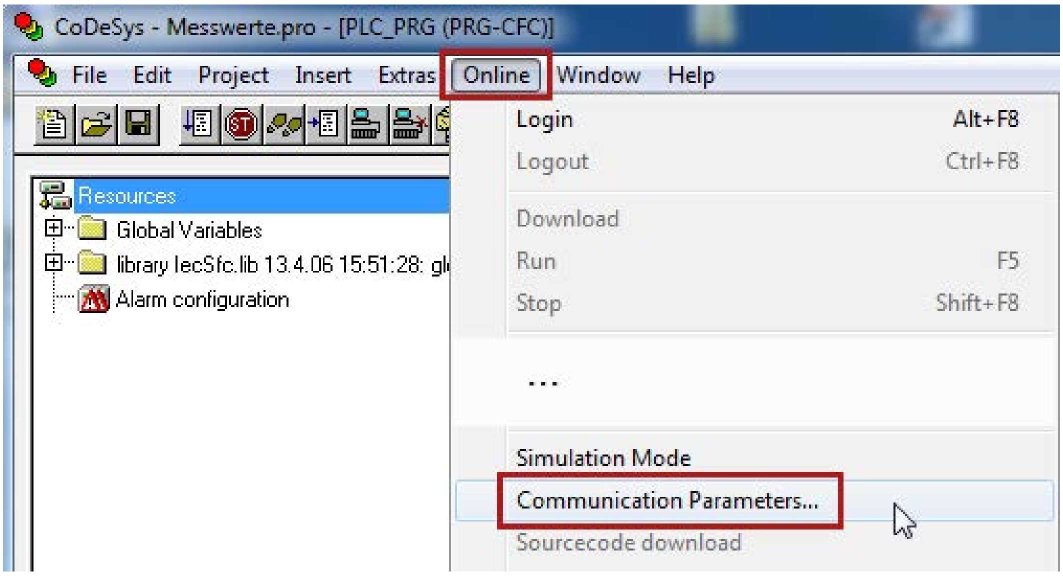
Controller	CODESYS version	PLC type, Driver	Port:	Note
WAGO 750-881	V2.3	Standard TCP/IP (Level 2)	2455	Symbols were not stored in PLC but read directly from SDB file.

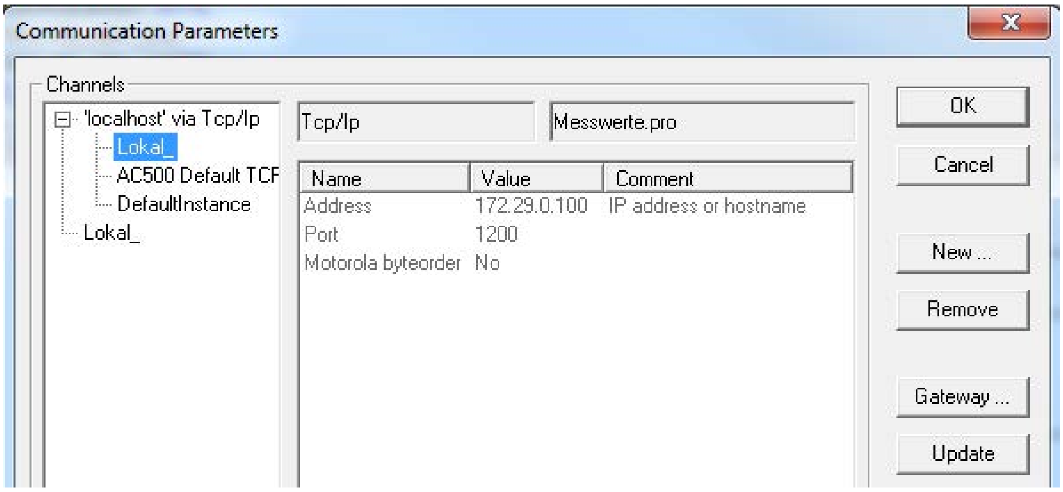
5.2 Example CODESYS PLCWinNT V2.4

The access to the controllers with CODESYS V2 is carried out via Ethernet. Therefore, you have to define the communication interface by means of your CODESYS software.

5.2.1 Communication setting

You will find the settings in the CODESYS software under *Online - Communication Parameters...*


CoDeSys - Messwerte.pro - [PLC_PRG (PRG-CFC)]
File Edit Project Insert Extras Online Window Help
Resources
Global Variables
library IecSfc.lib 13.4.06 15:51:28: gl
Alarm configuration
Login Alt+F8
Logout Ctrl+F8
Download
Run F5
Stop Shift+F8
...
Simulation Mode
Communication Parameters...
Sourcecode download



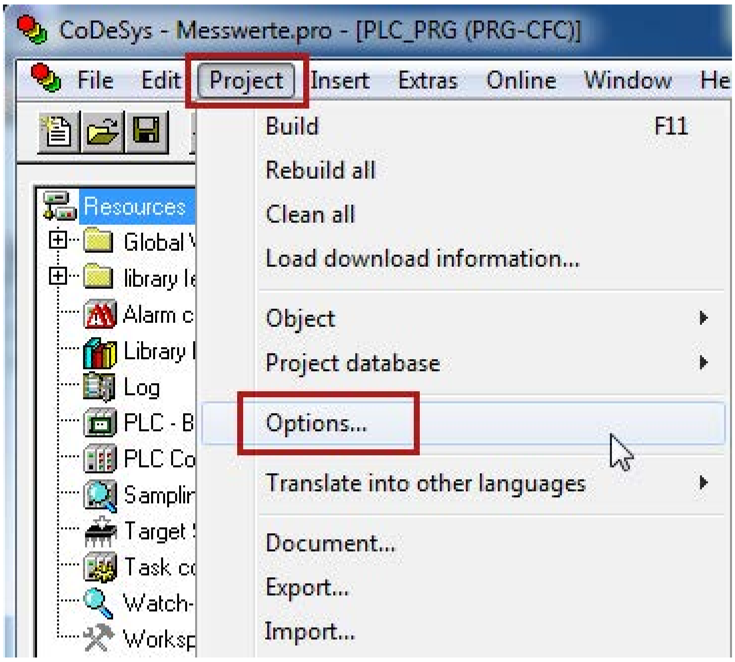
Tcp/Ip | Messwerte.pro

Name	Value	Comment
Address	172.29.0.100	IP address or hostname
Port	1200	
Motorola byteorder	No	

5.2.2 Create symbols

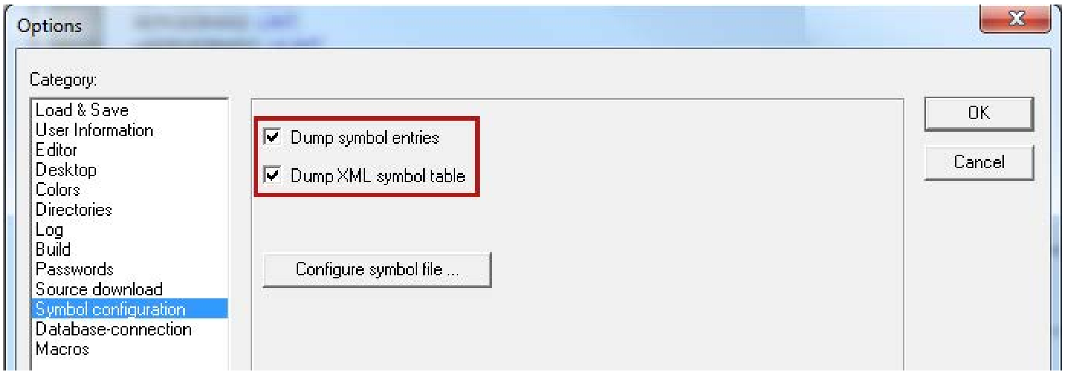
For creating the symbols on the controller the following steps are necessary:

1. Select *Project – Options...* in the opened project.



2. In the category *Symbol configuration* activate the options:
   - *Dump symbol entries*
   - *Dump XML symbol table*

   Then click on the button <Configure symbol file...>.

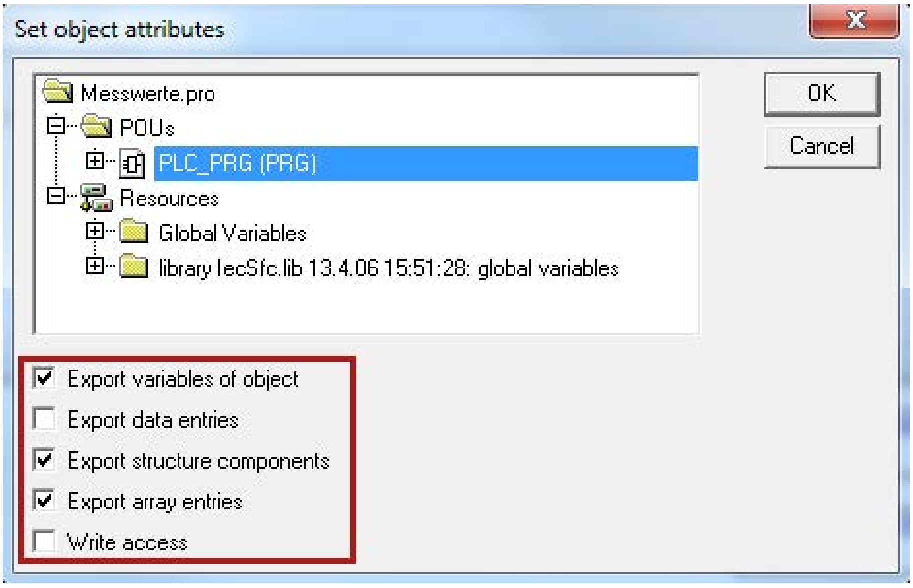


☑ Dump symbol entries
☑ Dump XML symbol table

3. There you mark the areas from which you want to create the symbols.

   Normally, it is sufficient selecting the user programs (POU) and the global variables. The following checkboxes have to be set for each selected object or object group in the tree as indicated below:

   - *Export variables of object*
   - *Export structure components*
   - *Export array entries*



Set object attributes

- Messwerte.pro
  - POUs
    - PLC_PRG (PRG)
  - Resources
    - Global Variables
    - library IecSfc.lib 13.4.06 15:51:28: global variables

☑ Export variables of object

□ Export data entries

☑ Export structure components

☑ Export array entries

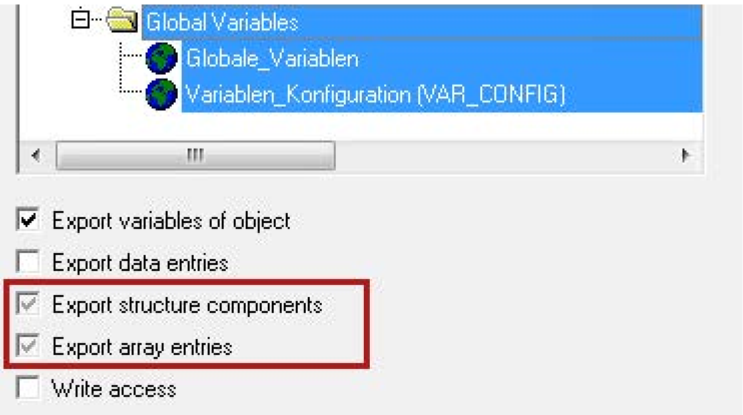
□ Write access

Note

Best Practice:

Select the required individual items or groups one after each other in the object tree (marked by a blue background color) and be sure to select the correct Export settings as indicated above.

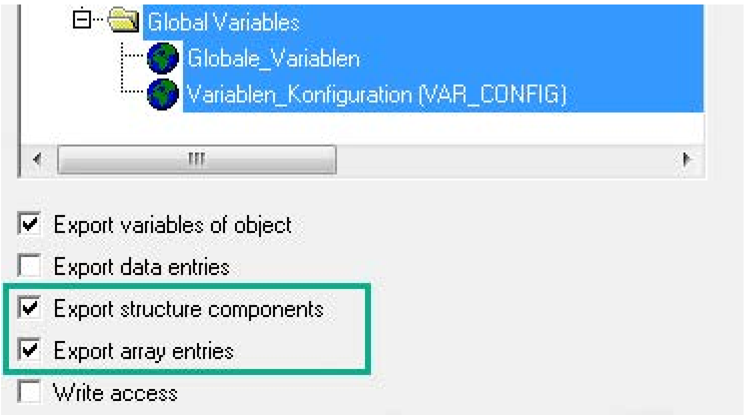
When selecting a group of items, be sure the checkboxes are fully checked and are **NOT in the indeterminate state (greyed)** as in the example below.



Global Variables
Globale_Variablen
Variablen_Konfiguration (VAR_CONFIG)

☑ Export variables of object
□ Export data entries
☑ Export structure components
☑ Export array entries
□ Write access

If this is the case, first clear the checkbox and then check it again.



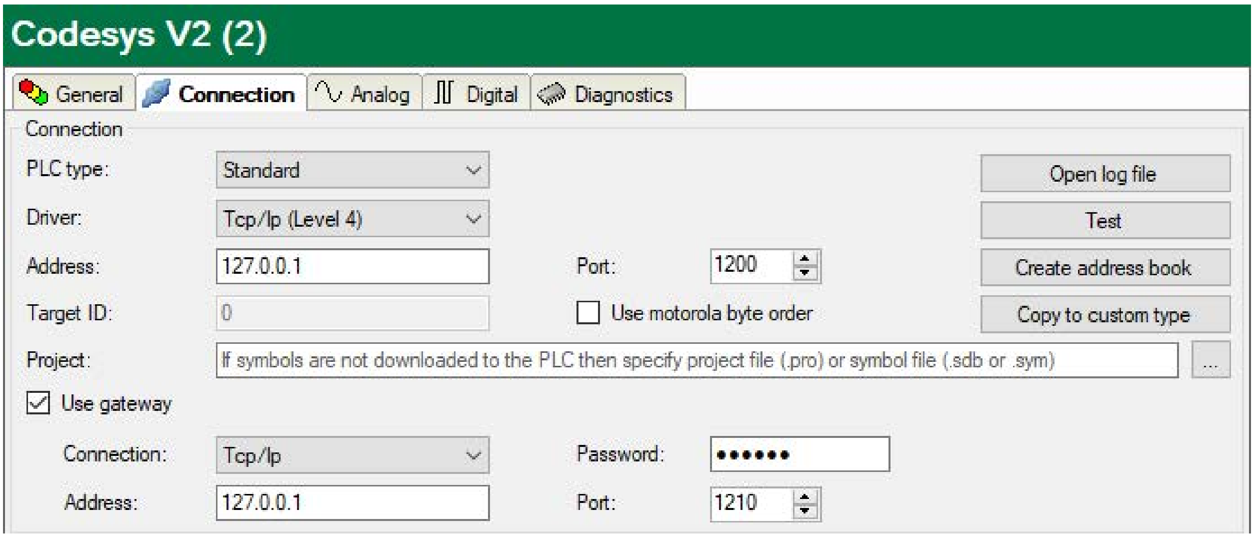
Global Variables
Globale_Variablen
Variablen_Konfiguration (VAR_CONFIG)

☑ Export variables of object
□ Export data entries
☑ Export structure components
☑ Export array entries
□ Write access

4. Close the dialog, completely compile the project and download the project into the controller.

5.2.3 Connection establishment in ibaPDA

You will find the connection data in the *Connection* tab, if you mark the module *CODESYS V2* in the interface tree view.



Connection

PLC type: Standard

Driver: Tcp/Ip (Level 4)

Address: 127.0.0.1 Port: 1200

Target ID: 0 □ Use motorola byte order

Project: If symbols are not downloaded to the PLC then specify project file (.pro) or symbol file (.sdb or .sym)

☑ Use gateway

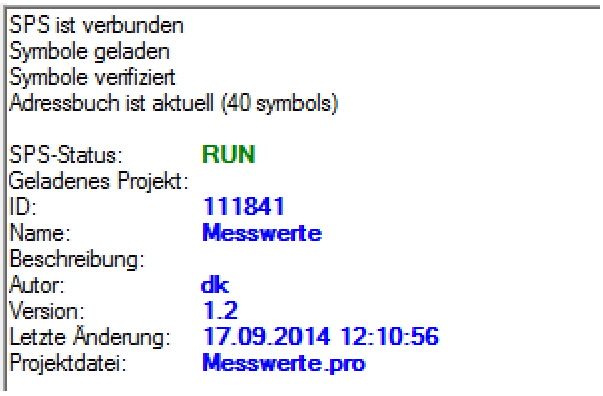
Connection: Tcp/Ip Password: ••••••

Address: 127.0.0.1 Port: 1210

For the connection to the runtime system CODESYS PLCWinNT V2.4 you have to set the following data:

- PLC type: Standard
- Driver: TCP/IP (Level 4)
- Address: IP address of the computer where the PLCWinNT runs
- Port: 1200
- Target-ID: inactive
- Motorola byte order: Disabled

By clicking on the button <Test>, the connection to the controller will be established and the project data will be displayed:



```
SPS ist verbunden
Symbole geladen
Symbole verifiziert
Adressbuch ist aktuell (40 symbols)

SPS-Status:        RUN
Geladenes Projekt:
ID:                111841
Name:              Messwerte
Beschreibung:
Autor:             dk
Version:           1.2
Letzte Änderung:   17.09.2014 12:10:56
Projektdatei:      Messwerte.pro
```

Other documentation

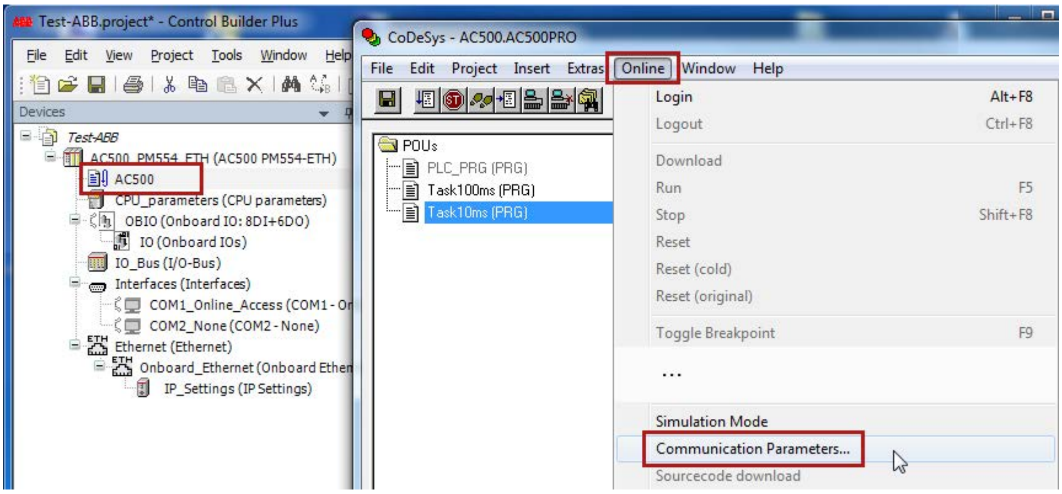
The example is based on the project *Measured values.pro* on the DVD “iba Software & Manuals”.

Path: “\04_Libraries_and_Examples\71_ibaPDA-Interface-CODESYS-Xplorer\Codesys_PLCWinNT_V2_4\”

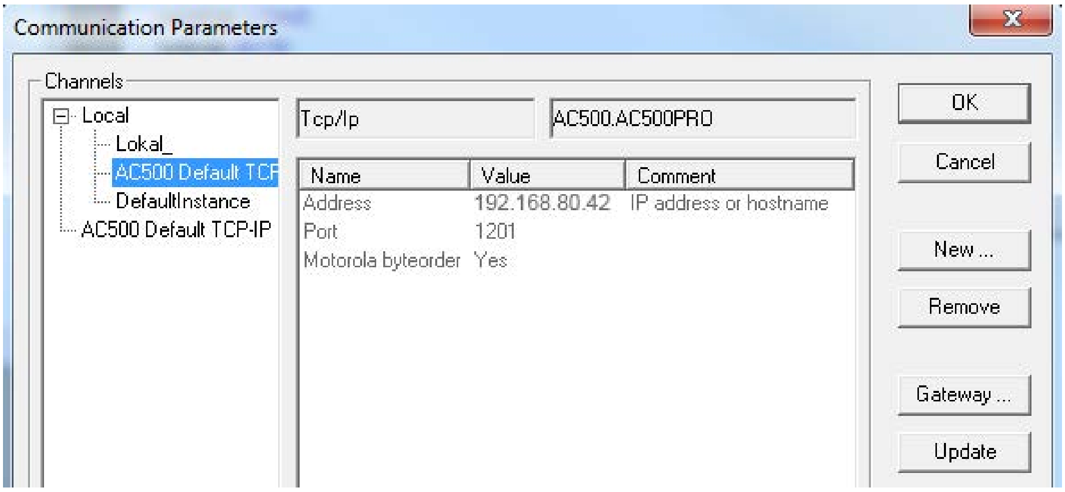
5.3 Example ABB AC500 (CPU PM551-TP-ETH)

5.3.1 Communication setting

Open the CODESYS project via the ABB Control Builder Plus. You will find the settings in the CODESYS software under *Online - Communication Parameters...*


Test-ABB.project* - Control Builder Plus
File Edit View Project Tools Window Help
Devices
Test-ABB
AC500_PM554_ETH (AC500 PM554-ETH)
AC500
CPU_parameters (CPU parameters)
OBIO (Onboard IO: 8DI+6DO)
IO (Onboard IOs)
IO_Bus (I/O-Bus)
Interfaces (Interfaces)
COM1_Online_Access (COM1 - On
COM2_None (COM2 - None)
Ethernet (Ethernet)
Onboard_Ethernet (Onboard Ether
IP_Settings (IP Settings)
CoDeSys - AC500.AC500PRO
File Edit Project Insert Extras Online Window Help
POUs
PLC_PRG (PRG)
Task100ms (PRG)
Task10ms (PRG)
Login Alt+F8
Logout Ctrl+F8
Download
Run F5
Stop Shift+F8
Reset
Reset (cold)
Reset (original)
Toggle Breakpoint F9
...
Simulation Mode
Communication Parameters...
Sourcecode download

Enter the IP address of the controller under channel AC500:


Communication Parameters

Channels

- Local
  - Lokal_
  - AC500 Default TCP
  - DefaultInstance
- AC500 Default TCP-IP

Tcp/Ip AC500.AC500PRO

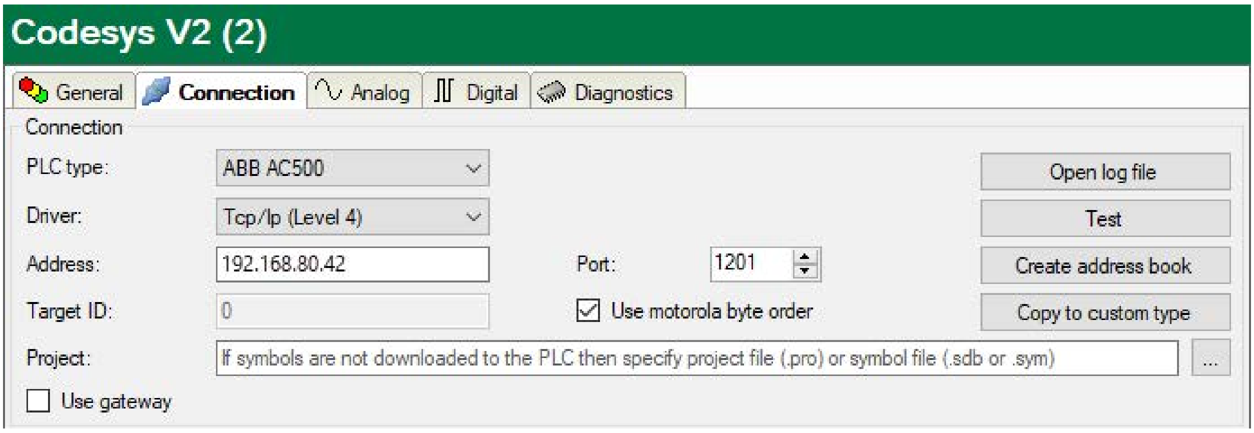
Name	Value	Comment
Address	192.168.80.42	IP address or hostname
Port	1201	
Motorola byteorder	Yes	

5.3.2 Create symbols

See chapter ➚ *Create symbols*, page 32

5.3.3 Connection establishment in ibaPDA

You will find the connection data in the *Connection* tab, if you mark the module *CODESYS V2* in the interface tree view.



Connection

PLC type: ABB AC500

Driver: Tcp/Ip (Level 4)

Address: 192.168.80.42

Port: 1201

Target ID: 0

☑ Use motorola byte order

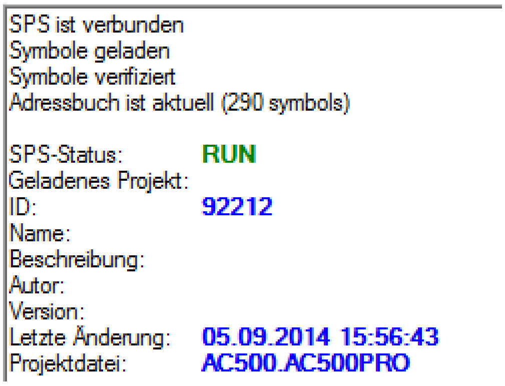
Project: If symbols are not downloaded to the PLC then specify project file (.pro) or symbol file (.sdb or .sym)

☐ Use gateway

For the connection to the runtime system ABB ACS500 (PM554-ETH) you have to set the following data:

- PLC type: ABB AC500
- Driver: TCP/IP (Level 4)
- Address: IP address of the controller, e.g. 192.168.21.125, if *ibaPDA* is in the same subnet as ACS500.
  If ACS500 is in another network segment, enable “Use gateway" and there, enter the IP address of the gateway server.
- Port: 1201
- Target-ID: inactive
- Motorola byte order: enabled

By clicking on the button <Test> the connection to the controller will be established and the project data will be displayed:



```
SPS ist verbunden
Symbole geladen
Symbole verifiziert
Adressbuch ist aktuell (290 symbols)

SPS-Status:         RUN
Geladenes Projekt:
ID:                 92212
Name:
Beschreibung:
Autor:
Version:
Letzte Änderung:    05.09.2014 15:56:43
Projektdatei:       AC500.AC500PRO
```

Other documentation

The example is based on the project *Test ABB.pro* on the DVD “iba Software & Manuals”.

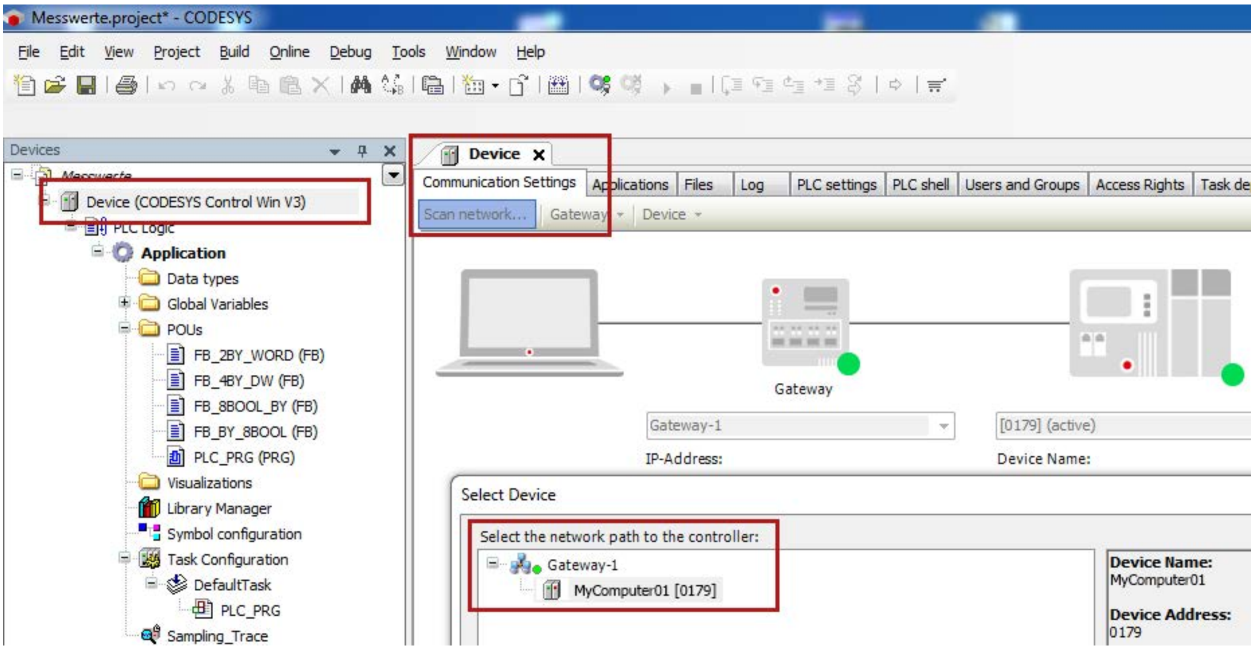
Path: “\04_Libraries_and_Examples\71_ibaPDA-Interface-CODESYS-Xplorer\ABB_AC500\”

5.4 Example CODESYS Control Win V3

The access to the controllers with CODESYS V3 is carried out via Ethernet. Therefore, you have to define the communication interface by means fo your CODESYS software.

5.4.1 Communication setting

In the Register *Device - Communication Settings - Scan networks* you can scan the network for CODESYS V3 runtime systems. The selected device will be loaded by double click.



Gateway-1

IP-Address:

[0179] (active)

Device Name:

Device Name:
MyComputer01

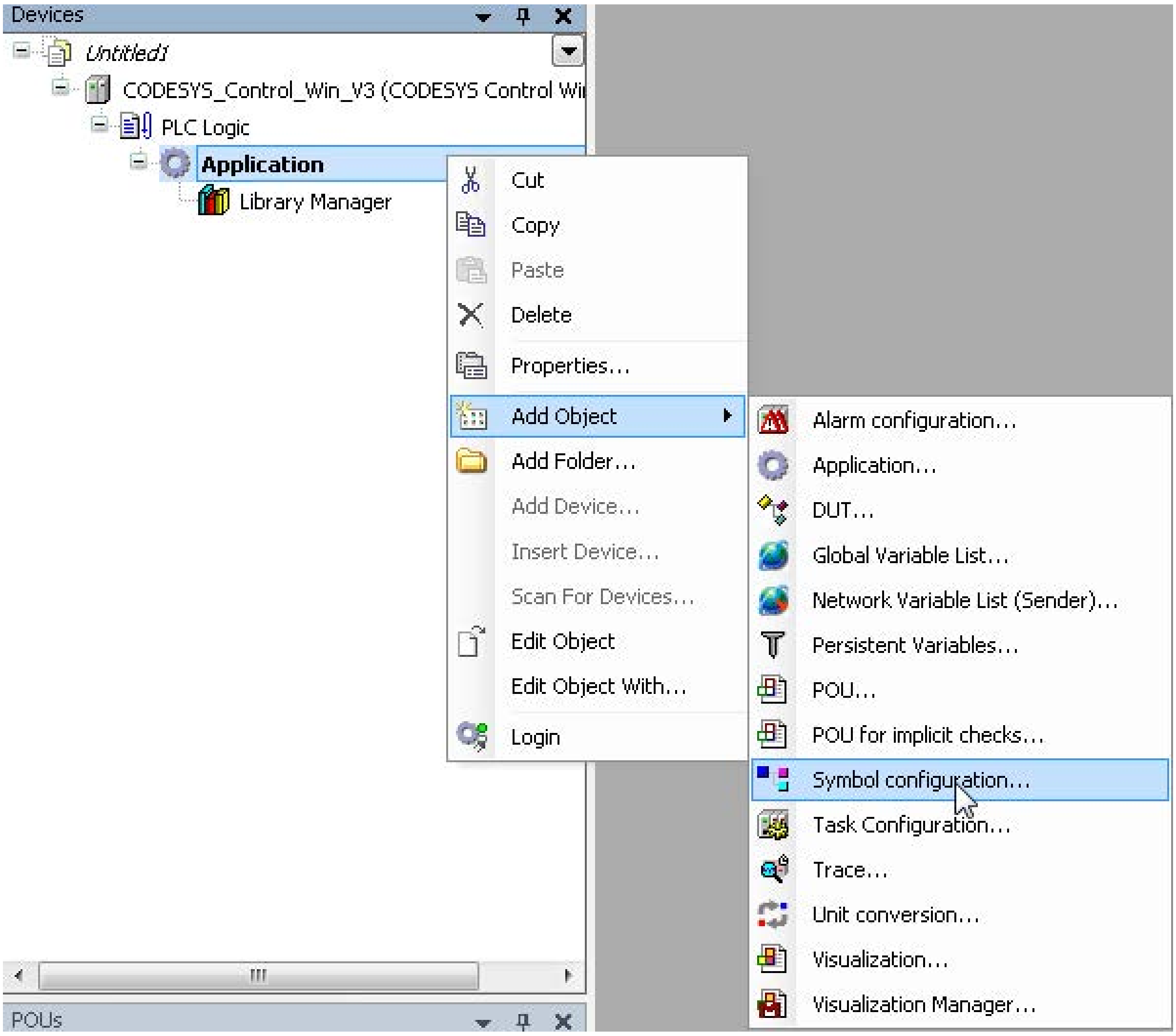
Device Address:
0179

5.4.2 Symbols in CODESYS V3

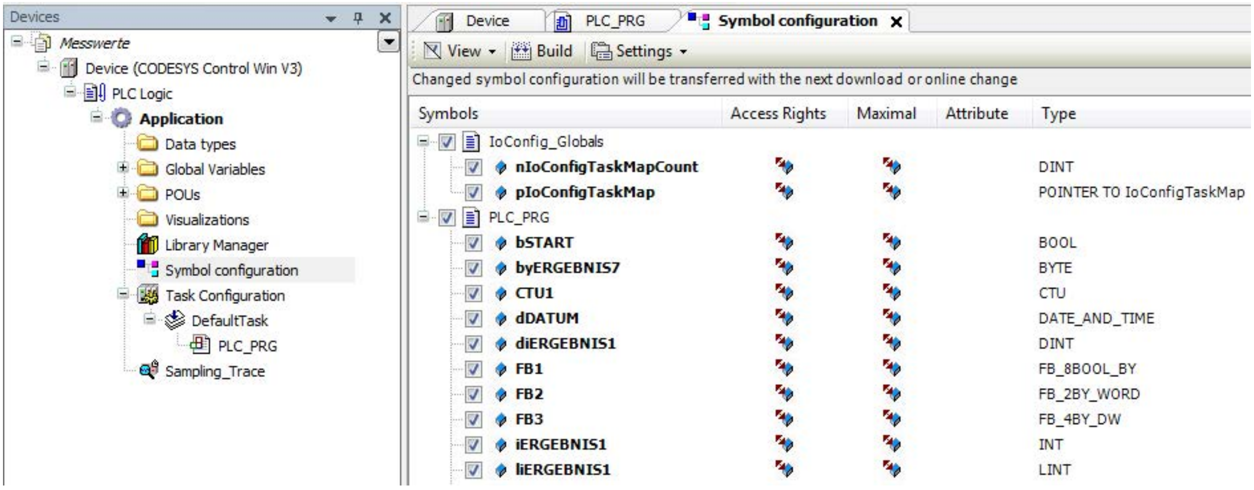
The access to the data of the controller is carried out via symbolic addresses. These are imported from the controller.

For creating the symbols the following steps are necessary:

1. Add the object *Symbol configuration* to your application.



2. Select the variables, you want to measure.



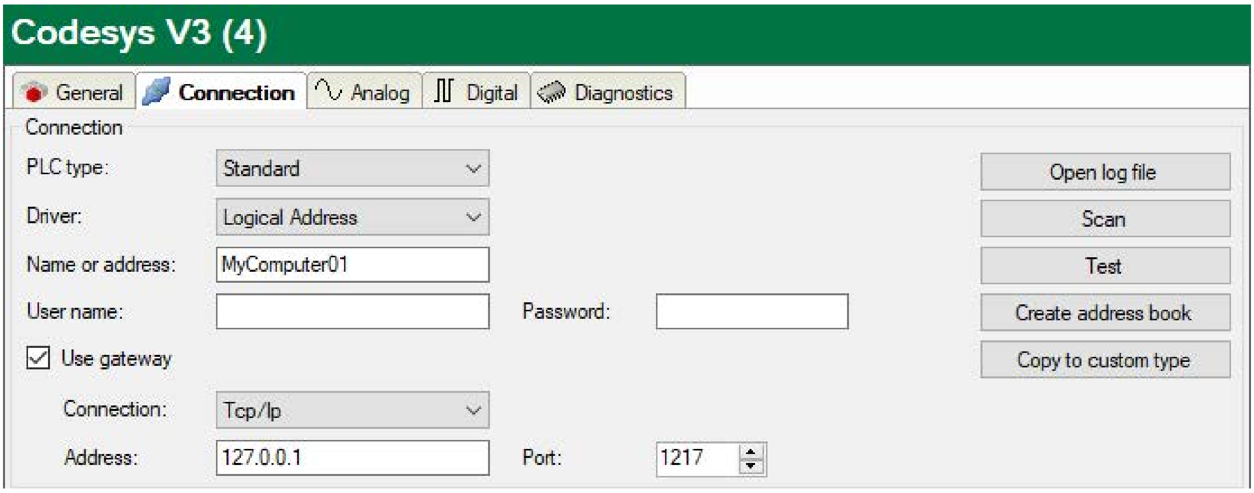
Changed symbol configuration will be transferred with the next download or online change

Symbols	Access Rights	Maximal	Attribute	Type
☑ IoConfig_Globals				
☑ nIoConfigTaskMapCount	⇄	⇄		DINT
☑ pIoConfigTaskMap	⇄	⇄		POINTER TO IoConfigTaskMap
☑ PLC_PRG				
☑ bSTART	⇄	⇄		BOOL
☑ byERGEBNIS7	⇄	⇄		BYTE
☑ CTU1	⇄	⇄		CTU
☑ dDATUM	⇄	⇄		DATE_AND_TIME
☑ diERGEBNIS1	⇄	⇄		DINT
☑ FB1	⇄	⇄		FB_8BOOL_BY
☑ FB2	⇄	⇄		FB_2BY_WORD
☑ FB3	⇄	⇄		FB_4BY_DW
☑ iERGEBNIS1	⇄	⇄		INT
☑ liERGEBNIS1	⇄	⇄		LINT

3. Close the dialogs, compile the project completely and download the project to the controller.

5.4.3 Connection establishment in ibaPDA

You will find the connection data in the *Connection* tab, if you mark the module *CODESYS V3* in the interface tree view.



Codesys V3 (4)

Connection

PLC type: Standard

Driver: Logical Address

Name or address: MyComputer01

User name:

Password:

☑ Use gateway

Connection: Tcp/Ip

Address: 127.0.0.1

Port: 1217

For a connection to the runtime system CODESYS Control Win V3 you have to set the following data:

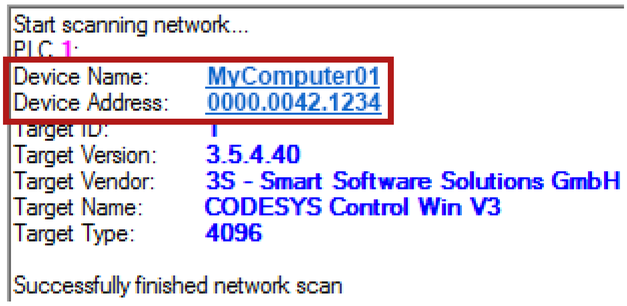
- PLC type: Standard
- Driver: Logical Address

By clicking on the button <Scan> the network will be scanned for a CODESYS V3 runtime system.

Tip

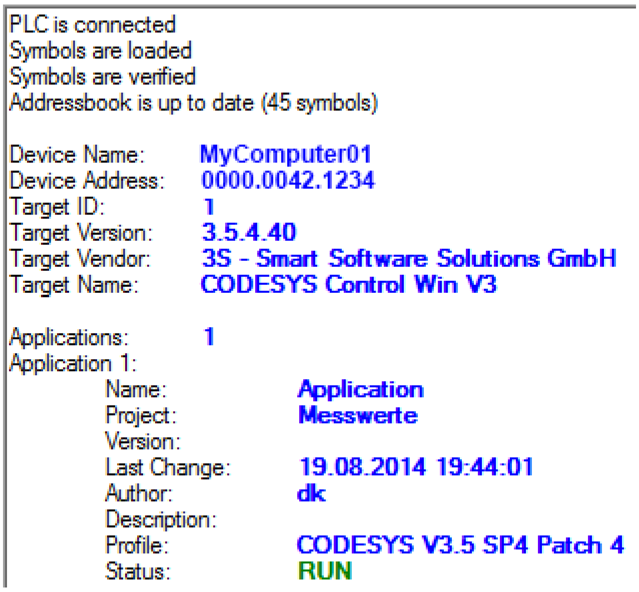
If no controller can be found, the controller is eventually in another network segment. Enable the option “Use gateway”, enter the IP address of the gateway server and click once again on <Scan>.

In case of success the found controllers are listed:



```
Start scanning network...
PLC 1:
Device Name:      MyComputer01
Device Address:   0000.0042.1234
Target ID:        1
Target Version:   3.5.4.40
Target Vendor:    3S - Smart Software Solutions GmbH
Target Name:      CODESYS Control Win V3
Target Type:      4096

Successfully finished network scan
```

By clicking on the hyperlink under *Device name* or *Device Address* the connection to the controller is established and the following result is shown:



```
PLC is connected
Symbols are loaded
Symbols are verified
Addressbook is up to date (45 symbols)

Device Name:      MyComputer01
Device Address:   0000.0042.1234
Target ID:        1
Target Version:   3.5.4.40
Target Vendor:    3S - Smart Software Solutions GmbH
Target Name:      CODESYS Control Win V3

Applications:     1
Application 1:
           Name:          Application
           Project:       Messwerte
           Version:
           Last Change:   19.08.2014 19:44:01
           Author:        dk
           Description:
           Profile:       CODESYS V3.5 SP4 Patch 4
           Status:        RUN
```

Other documentation

The example is based on the project *Measured values.project* on the DVD “iba Software & Manuals”.

Path: “\04_Libraries_and_Examples\71_ibaPDA-Interface-CODESYS-Xplorer\Control_Win_V3\”

6 Support and contact

Support

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

Note

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

Contact

Headquarters

iba AG
Koenigswarterstrasse 44
90762 Fuerth
Germany

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

Mailing address

iba AG
Postbox 1828
D-90708 Fuerth, Germany

Delivery address

iba AG
Gebhardtstrasse 10
90762 Fuerth, Germany

Regional and Worldwide

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

www.iba-ag.com.