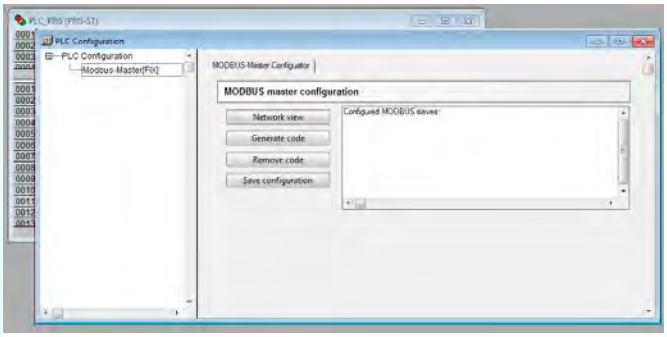
Manual



WAGO Software

MODBUS Master Configurator

for Configuration of MODBUS Networks with WAGO-I/O-PRO (CODESYS)

Version 1.0.0

© 2014 by WAGO Kontakttechnik GmbH & Co. KG
All rights reserved.

WAGO Kontakttechnik GmbH & Co. KG

Hansastraße 27
D-32423 Minden

Phone: +49 (0) 571/8 87 – 0
Fax: +49 (0) 571/8 87 – 1 69

E-Mail: info@wago.com

Web: http://www.wago.com

Technical Support

Phone: +49 (0) 571/8 87 – 5 55
Fax: +49 (0) 571/8 87 – 85 55

E-Mail: support@wago.com

Every conceivable measure has been taken to ensure the accuracy and completeness of this documentation. However, as errors can never be fully excluded, we always appreciate any information or suggestions for improving the documentation.

E-Mail: documentation@wago.com

We wish to point out that the software and hardware terms as well as the trademarks of companies used and/or mentioned in the present manual are generally protected by trademark or patent.

Table of Contents

1 Notes about this Documentation ... 5
1.1 Scope of Validity ... 5
1.2 Copyright ... 5
1.3 Symbols ... 6
1.4 Number Notation ... 8
1.5 Font Conventions ... 8
2 Important Notes ... 9
2.1 Legal Bases ... 9
2.1.1 Subject to Changes ... 9
2.1.2 Personnel Qualification ... 9
2.2 System Requirements ... 9
2.2.1 PC Hardware ... 9
2.2.2 PC Software ... 10
2.3 Hardware Constraints ... 11
2.3.1 Supported MODBUS Master Devices ... 11
2.3.2 Hardware Interfaces ... 11
2.3.3 Acceptable MODBUS Slaves ... 12
2.4 Safety Advice (Precautions) ... 12
3 General ... 13
3.1 Installation ... 13
3.2 Preparation ... 13
3.3 Open MODBUS Master Configurator in WAGO-I/O-*PRO* ... 14
4 The MODBUS Master Configurator ... 15
4.1 Function ... 15
4.2 Configuration Procedure ... 17
4.2.1 Step 1: "Create project" ... 17
4.2.2 Step 2: "Configure MODBUS network" ... 17
4.2.3 Step 3: "Configure network devices" ... 18
4.2.4 Step 4: "Generate code" ... 19
4.2.5 Step 5: "Integrate into application" ... 19
5 User Interface ... 20
5.1 Main View ... 20
5.2 "MODBUS network" Dialog ... 20
5.2.1 Context Menu for the "MODBUS master" Entry ... 22
5.2.1.1 "Parameter code generator" Dialog ... 23
5.2.2 Context Menu for the "Interface (COMx)" Entry ... 24
5.2.2.1 "Settings: Serial interface" Dialog ... 25
5.2.2.2 "MODBUS slave selection" Dialog ... 26
5.2.2.3 "Search device" Dialog ... 26
5.2.3 Context Menu for the "Interface (ETHERNET)" Entry ... 27
5.2.3.1 "MODBUS slave selection" Dialog ... 28
5.2.3.2 "Search device" Dialog" ... 29
5.2.4 Context Menü for the "ModbusSlave" Entry ... 30
5.2.4.1 "Communication settings" – Interface (COMx) Dialog) ... 31
5.2.4.2 "Communication settings" – Interface (ETHERNET Dialog) ... 32

5.2.4.3 "MODBUS slave selection" Dialog ... 34
5.2.5 Context Menu for the "Generic variables" Entry ... 35
5.2.5.1 "WAGO MODBUS register" Dialog ... 36
5.2.6 Context Menu for the "PFC variables" Entry ... 37
5.2.7 Context menu for the "K-Bus" Entry ... 38
5.2.7.1 "Module selection" Dialog ... 39

6 Operation ... 40

6.1 Make Network Presets ... 40
6.2 Add Interface ... 41
6.2.1 Add Serial or ETHERNET Interface ... 41
6.2.2 Configure Serial Interface ... 41
6.3 Insert and Configure MODBUS Slave ... 41
6.3.1 Insert MODBUS Slave Online ... 41
6.3.2 Insert MODBUS Slave Offline ... 42
6.3.3 Configure MODBUS Slave ... 42
6.3.3.1 Configure MODBUS Slave via Serial Interface ... 42
6.3.3.2 Configure MODBUS Slave via ETHERNET Interface ... 43
6.3.4 Configure WAGO MODBUS Slave Hardware Online ... 43
6.3.5 Configure WAGO MODBUS Slave Hardware Offline ... 43
6.4 Insert and Configure Variables ... 44
6.4.1 Add Generic Variables ... 44
6.4.2 Configure Generic Variables ... 44
6.4.3 Add PFC Variables ... 45
6.4.4 Configure PFC Variables ... 45
6.4.5 Configure K-Bus Variables ... 46
6.5 Generate IEC Code ... 46

7 Glossary ... 48

8 Appendix ... 51

8.1 WAGO-I/O-*PRO*-Libraries ... 51
8.1.1 WAGO-I/O-*PRO* Function Blocks ... 52
8.1.2 Directory Structure ... 52
8.1.3 Program Execution of the Generated Code ... 53
8.1.4 Cycle Time of the "MBCFG_MASTER_TASK" ... 54
8.1.5 Data Storage, Persistent ... 54
8.1.6 MODBUS Function Codes ... 55
8.1.7 Access to WAGO Slaves ... 55
8.1.8 Access to the Hardware Area ... 55
8.1.9 Access to the PFC Variable Area and Flags ... 55
8.1.10 Access to the Special Feature Registers ... 56
8.1.11 Optimizing Communication with WAGO MODBUS Slaves ... 56
8.1.12 Access to Generic MODBUS Slaves, Manufacturer-Independent ... 56
8.1.13 Access to MODBUS Data Points from the IEC 61131 Program ... 56

List of Figures ... 58

List of Tables ... 59

1 Notes about this Documentation

Note

Keep this documentation!
The operating instructions are part of the product and shall be kept for the entire lifetime of the product. They shall be transferred to each subsequent user of the product. Care must also be taken to ensure that any supplement to these instructions are included, if applicable.

1.1 Scope of Validity

This documentation applies to the software of the MODBUS Master Configurator.

1.2 Copyright

This Manual, including all figures and illustrations, is copyright-protected. Any further use of this Manual by third parties that violate pertinent copyright provisions is prohibited. Reproduction, translation, electronic and phototechnical filing/archiving (e.g., photocopying) as well as any amendments require the written consent of WAGO Kontakttechnik GmbH & Co. KG, Minden, Germany. Non-observance will involve the right to assert damage claims.

1.3 Symbols

DANGER

Personal Injury!
Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

DANGER

Personal Injury Caused by Electric Current!
Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

WARNING

Personal Injury!
Indicates a moderate-risk, potentially hazardous situation which, if not avoided, could result in death or serious injury.

CAUTION

Personal Injury!
Indicates a low-risk, potentially hazardous situation which, if not avoided, may result in minor or moderate injury.

NOTICE

Damage to Property!
Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.

NOTICE

Damage to Property Caused by Electrostatic Discharge (ESD)!
Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.

Note

Important Note!
Indicates a potential malfunction which, if not avoided, however, will not result in damage to property.

Information

Additional Information:
Refers to additional information which is not an integral part of this documentation (e.g., the Internet).

1.4 Number Notation

Table 1: Number notation

Number code	Example	Note
Decimal	100	Normal notation
Hexadecimal	0x64	C notation
Binary	'100' '0110.0100'	In quotation marks, nibble separated with dots (.)

1.5 Font Conventions

Table 2: Font conventions

Font type	Indicates
italic	Names of paths and data files are marked in italic-type. e.g.: *C:\Programme\WAGO-I/O-CHECK*
Menu	Menu items are marked in bold letters. e.g.: **Save**
>	A greater-than sign between two names means the selection of a menu item from a menu. e.g.: **File** > **New**
Input	Designation of input or optional fields are marked in bold letters, e.g.: **Start of measurement range**
“Value”	Input or selective values are marked in inverted commas. e.g.: Enter the value “4 mA” under **Start of measurement range**.
[Button]	Pushbuttons in dialog boxes are marked with bold letters in square brackets. e.g.: **[Input]**
[Key]	Keys are marked with bold letters in square brackets. e.g.: **[F5]**

2 Important Notes

This section describes the legal principles and system requirements for using the software in compliance with intended purpose, underlying provisions and stated specifications.

2.1 Legal Bases

2.1.1 Subject to Changes

WAGO Kontakttechnik GmbH & Co. KG reserves the right to provide for any alterations or modifications that serve to increase the efficiency of technical progress. WAGO Kontakttechnik GmbH & Co. KG owns all rights arising from the granting of patents or from the legal protection of utility patents. Third-party products are always mentioned without any reference to patent rights. Thus, the existence of such rights cannot be excluded.

2.1.2 Personnel Qualification

Any steps related to the use of WAGO software may only be performed by qualified employees with sufficient knowledge of handling the respective PC system used.

Steps in which files are created or changed on the PC system may only be performed by qualified employees with sufficient knowledge in the administration of the PC system used in addition to the aforementioned.

Steps in which the behavior of the PC system in a network is changed may only be performed by qualified employees with sufficient knowledge in the administration of the network used in addition to the aforementioned.

2.2 System Requirements

2.2.1 PC Hardware

Table 3: Required PC hardware

Components	Requirements
Operating System	Windows XP (SP3 or higher); Windows 7/8
Memory	Min. 1 GB RAM (recommended: 2 GB RAM or more)
Free hard disk storage	Min. 1.5 MB for the WAGO MODBUS Configurator
Processor	Min. 1 GHz or higher
Other	Installed network card, standard web browser with Java support

2.2.2 PC Software

Table 4: Required software

Components	Source (Item No.)
WAGO-I/O-*PRO* **Version 2.3.9.40 or higher** (includes MODBUS Master Configurator) Customers with older versions should contact Support: support@wago.com	WAGO (759-333) (Manual for WAGO-I/O-*PRO* can be downloaded free at: www.wago.com).

Table 5: Optional software

Components	Source (Item No.)
WAGO-I/O-*CHECK*	WAGO (759-302)
WAGO Ethernet Settings	WAGO (free download at: www.wago.com)

2.3 Hardware Constraints

The hardware constraints are based on the WAGO products from the WAGO-I/O-SYSTEM 750, 758 and 762 Series.
The WAGO devices and interfaces that can be used as MODBUS masters and slaves are listed in the sections below.

2.3.1 Supported MODBUS Master Devices

The MODBUS Master Configurator supports the following WAGO devices that are programmable with CODESYS 2.3:

Table 6: Supported MODBUS master devices

Series / Subgroup	Device Class	Item Mumber
WAGO-I/O-SYSTEM 750 / (P-)FC 750-8xx	16-bit derivatives	750-804, 750-806, 750-819, 750-833, 750-837, 750-842, 750-843
	32-bit derivatives	750-830, 750-831, 750-841, 750-849, 750-871, 750-872, 750-873, 750-880, 750 881, 750-882, 750-884, 750-885, 750-8203, 750-8204, 750-8206
WAGO-I/O-SYSTEM 758 / IPC 758-87x	Compact Industrial PCs	758-874/000-xxx, 758-875/000-xxx 758-876/000-xxx
WAGO PERSPECTO® 762 / CP 762-3xxx/000-001	PERSPECTO® Control-Panel with Target Visualization	762-3035/000-001, 762-3057/000-001, 762-3104/000-001, 762-3121/000-001, 762-3150/000-001

The difference in device classes for the 16-bit and 32-bit derivatives is in access to the ETHERNET interface, in which different CODESYS2 ETHERNET libraries are accessed for ETHERNET access.

The class of devices from WAGO's PERSPECTO® 762 Series differs when accessing serial communication because there is no internal data bus (K-Bus).

2.3.2 Hardware Interfaces

For communication, the MODBUS master uses the ETHERNET or RS-232 hardware interfaces available on the housing of the MODBUS master device or the serial I/O modules plugged in the fieldbus node (see table below).

Table 7: Hardware interfaces

Interfaces	WAGO Device, Item Number
ETHERNET interface	MODBUS master device, see table above
RS-232 interface	MODBUS master device, see table above
RS-232 C serial interface	750-650
RS-485 serial interface	750-653
RS-232 / RS-485 serial interface	750-652

2.3.3 Acceptable MODBUS Slaves

Any device that supports MODBUS ASCII, MODBUS RTU, MODBUS TCP or MODBUS UDP and at least one of the MODBUS function codes (except FC22 and FC23) can be used as a MODBUS slave.

When using MODBUS slaves from third-party manufacturers, include them in the network configuration by creating generic MODBUS slaves.

For WAGO MODBUS slaves, the hardware configuration of the fieldbus node can be manually created in the MODBUS Master Configurator or by simply scanning for devices.

The WAGO devices listed below are acceptable as MODBUS slaves.

Table 8: Possible WAGO MODBUS slaves

MODBUS Slaves	WAGO Devices
MODBUS slaves via ETHERNET (UDP and TCP)	ETHERNET fieldbus coupler from the 750 Series: 750–3xx ETHERNET fieldbus controller from the 750 Series: 750-8xx IPCs der Serie 758: 758-087x PFC200-Derivate: 750-860x
MODBUS slaves with serial interface (MODBUS RTU)	PFC200 derivatives: 750-860x RS232/RS485 coupler (750-312, 750-314, …) RS232/RS485 controller (750-812, 750-814,750-830, …)

2.4 Safety Advice (Precautions)

Note

Use up-to-date security software!
Secure operation of the PC system can be at risk as a result of malware such a viruses and Trojans, as well as related threats such as denial-of-service attacks. Therefore, make sure that the latest security software and definitions are always installed on the PC system.

Note

Disable or uninstall software that is no longer required!
The vulnerability of a PC system against malware and related threats increases with the number of installed or active software components (applications and services). Therefore, uninstall or disable software components that are not needed for the purpose at hand.

3 General

3.1 Installation

The MODBUS Master Configurator is part of the WAGO-I/O-*PRO* software (ItemNo.: 759-333) **2.3.9.40** or higher.

WAGO-I/O-*PRO* is easy to install using a „Setup.exe“ file with a user guided wizard.

3.2 Preparation

When using the MODBUS Master Configurator, it is important that the hardware of your MODBUS master and MODBUS slaves used, as well as the MODBUS network, are set up and working correctly.

A communication link from your PC to your MODBUS master-enabled device must be established for online configuration. This can be achieved via ETHERNET connection or serial interface on the programmable fieldbus controller being used.

In addition, the WAGO-I/O-*PRO* software (version with integrated MODBUS Master Configurator) must be installed on your PC.

Note

Please note the version of WAGO-I/O-*PRO* with integrated MODBUS Master Configurator!
Please note that the MODBUS Master Configurator is integrated in WAGO-I/O-*PRO* version V 2.3.9.40 or higher.
Customers with older versions should contact Support:
support@wago.com

Note

Important Note! -> Durch Hinweisüberschrift ersetzen!
Please note the following when launching WAGO-I/O-*PRO:*

- Do not use an existing project file, e.g., double-click, to launch WAGO-I/O-*PRO*. Doing so prevents code generation.
- You can open only one instance of WAGO-I/O-*PRO* at once.
- You have to use the " remote" command line argument to start WAGO-I/O-*PRO* in order to add modules in WAGO-I/O-*PRO*:
  (Example: "<Install_Path>\WAGO-Software\CODESYS V2.3\CODESYS.exe" –remote).

After creating a new project in WAGO-I/O-*PRO* and selecting the target system according to your MODBUS master, you can open the controller’s MODBUS Master Configurator and begin configuration.

3.3 Open MODBUS Master Configurator in WAGO-I/O-*PRO*

1. Open the MODBUS Master Configurator in the WAGO I/O-PRO programming environment directly.
   First, select the target system in a new project, the required programming language and then the "**Resources**" tab.

2. Double-click "**Controller Configuration**" in the tree structure to open the controller configuration dialog.

3. In this dialog, expand the "**Hardware configuration**" entry in the tree structure.

4. Then click to highlight the „**Modbus-Master[FIX]**“.

   In the right area of the dialog, the configuration dialog for the MODBUS master configuration appears.
   The dialog contains the "MODBUS Master Configurator" tab.

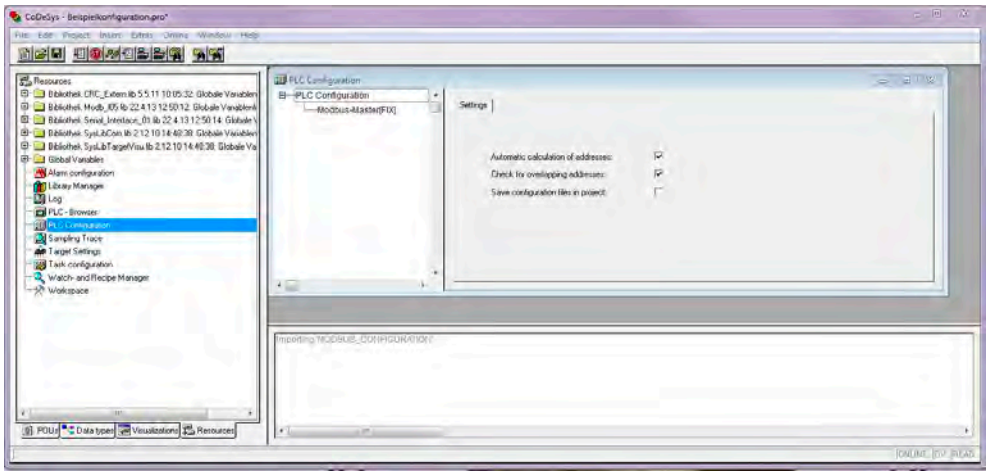


Figure 1: View of the controller configuration with MODBUS Master Configurator

4 The MODBUS Master Configurator

4.1 Function

The MODBUS Master Configurator is an extension application of the WAGO-I/O-*PRO* (CODESYS 2.3) programming environment and is used directly in the programming environment.

The MODBUS Master Configurator simplifies MODBUS network configuration in a WAGO-I/O-*PRO* project that is connected to a WAGO device with MODBUS master support.

The figure below shows an example configuration of a MODBUS network using the MODBUS Master Configurator in CODESYS with a device from WAGO's PERSPECTO® 762 Series as a MODBUS master.

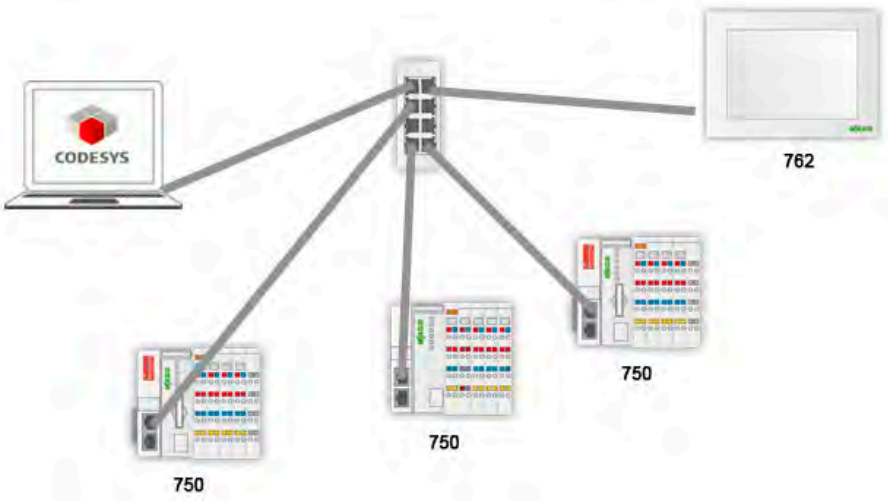


Figure 2: Example configuration for using the MODBUS Master Configurator

The dialog streamlines network creation in the MODBUS Master Configurator. However, scanning for MODBUS-enabled WAGO devices makes creating a network even easier. For any WAGO devices found, a node scan can also read the connected I/O module configuration and determine all data points.

To scan, the PC must have online access to the WAGO devices via ETHERNET or a serial interface.

From the following configuration settings specified for respective communication processes, the MODBUS Master Configurator can automatically create a corresponding IEC 61131 program code with the click of a mouse. This code is then directly available in the WAGO-I/O-*PRO* (CODESYS 2.3) programming environment.

For data access to the WAGO MODBUS slaves, no manual input of the MODBUS communication parameters is required. The MODBUS function codes (FC) used can be specified via the MODBUS Master Configurator depending on which data ranges (bit/register area) should be accessed.

The MODBUS Master Configurator maximizes support for correctly selecting the MODBUS addresses and function codes.

Additionally, "optimized" or "explicit" communication for individual data channel access can be selected by clicking on the appropriate checkbox. The definitions are included when generating the code and read communication jobs.

To illustrate the types of access, an example is shown below.
If a slave provides three data points, for example, which are defined as "optimized", they are read or written with this MODBUS FC at one time.
If the data points are marked as "explicit", however, the MODBUS master performs three requests to read or write the data points from the slave.
It should be noted that some MODBUS slaves do not allow "optimized" reading because there may be data with more than one word/coil behind a MODBUS address that can only be read as "explicit".

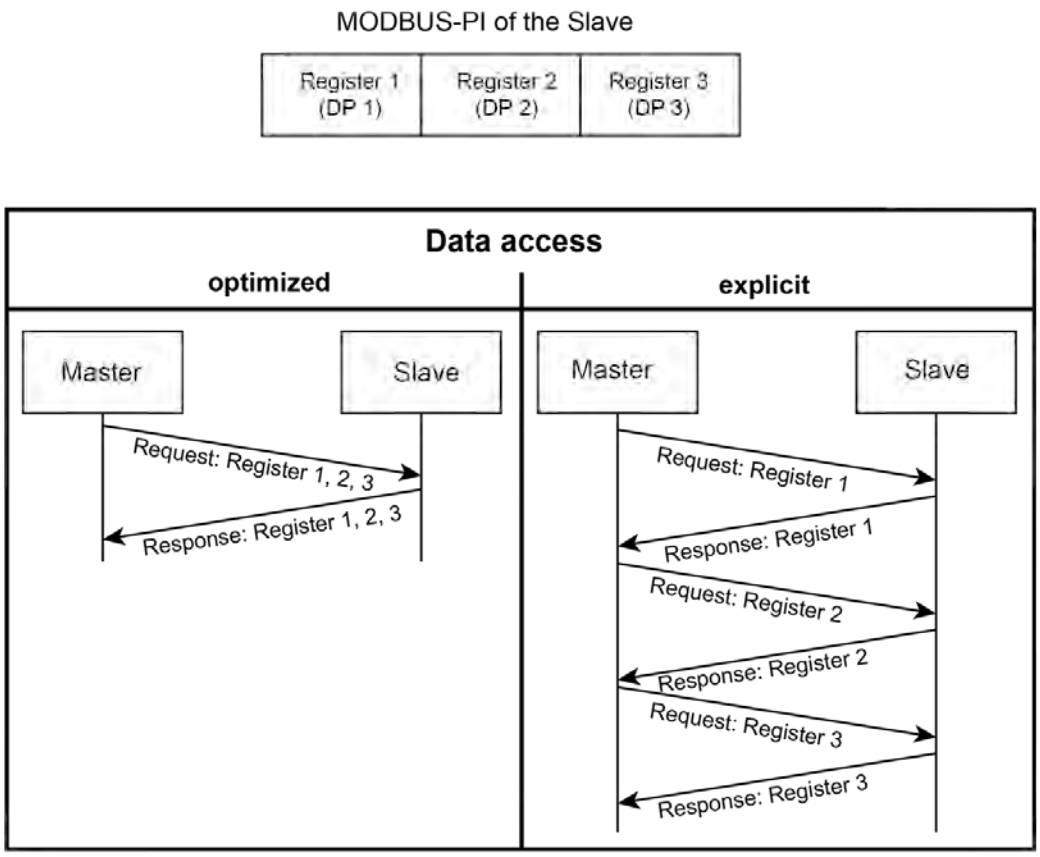


Figure 3: Example of "optimized" and "explicit" data access

4.2 Configuration Procedure

The MODBUS network is configured in five simple steps; steps 1 and 5 in the WAGO-I/O-*PRO* programming environment and steps 2–4 with the MODBUS Master Configurator.

The user project is created in WAGO-I/O-*PRO*, the hardware and software of the network devices configured in the MODBUS Master Configurator and subsequent connection of network devices is then implemented in the WAGO-I/O-*PRO* project.

Table 9: Configuration procedure overview

Tool	Step	
WAGO-I/O-*PRO*	**Step 1:**	Create project
MODBUS Master Configurator	**Step 2:**	Configure MODBUS network
	Step 3:	Configure network devices
	Step 4:	Generate code
WAGO-I/O-*PRO*	**Step 5:**	Integrate into application

The basic process is briefly outlined in the steps shown in the following sections. Step-by-step instructions for this process can be found in the "Operation" section.

Details and information about the software are available in the "MODBUS Master Configurator" and "WAGO-I/O-*PRO* Function Blocks" main sections.

4.2.1 Step 1: "Create project"

Create your project in WAGO-I/O-*PRO*.
Start WAGO-I/O-*PRO* with the "-remote" command line argument and create a new project in WAGO-I/O-*PRO*. Select the target system based on the MODBUS master used.

4.2.2 Step 2: "Configure MODBUS network"

Configure the MODBUS network in the MODBUS Master Configurator.
First, configure the MODBUS network hardware in the controller configuration of WAGO-I/O-*PRO* using the MODBUS Master Configurator.
Once you click the "**MODBUS Master[FIX]**" entry in the tree structure of the controller configuration, the interface of the MODBUS Master Configurator appears.

Click the **[Network View]** button to open the "MODBUS Network" dialog for network configuration.

Within this dialog box, utilize a tree structure to map the structure of your hardware by adding corresponding entries for existing interfaces and connected MODBUS slaves.

The respective settings for the interfaces and MODBUS slaves are made from their corresponding context menus (right-click) for these entries.

Details of the process are briefly described in the table below.

Table 10: General settings from the context menus of tree structure entries

Entry Context Menu	Description
"MODBUS Master"	At the beginning, an entry already exists for your MODBUS master in the representation of the tree structure. First, you can perform settings for the code generator from the context menu of the MODBUS master and then add appropriate interfaces to the master for your existing MODBUS slaves.
"Interface"	You can then automatically or manually add the connected MODBUS slave to each interface from the context menu for the respective interface: Automatically – online from the "*Find devices*" context menu. Offline – via "*Add MODBUS slave*", which provides a list of all possible slaves (WAGO or generic) for manually selecting MODBUS slaves.
"MODBUS Slave"	The communication preferences (transport protocol and IP address) for the respective MODBUS slave are made from the context menu of the corresponding MODBUS slave, via the "*Edit*" menu item / **MODBUS ETHERNET Settings** tab.
"K-Bus"	To select the individual I/O modules on a fieldbus node, you can open the dialog on the tree structure entry "**K-Bus** " via the context menu/*"Add"* menu item and select your physical hardware components offline. Optionally, you can also perform an automatic device scan online from the context menu/menu item "*Scan node*" on the tree structure entry "**K-Bus**".

4.2.3 Step 3: "Configure network devices"

You can also configure individual network devices in the MODBUS Master Configurator.
Once you have configured the structure of the network based on your physical structure, you can rename the displayed variable name and associated comment in the MODBUS network list for each individual data channel and specify the required access (none, read, write) from a selection list. You can also choose whether communication should be "optimized" (default) or "explicit".
In addition, you can add generic variables and — depending on availability — PFC variables, as well as declare their types as required.

4.2.4 Step 4: "Generate code"

Once you have defined all required functionality, clicking the " " button (**[Generate code]**) automatically generates the defined variables and access function codes as IEC function blocks.

4.2.5 Step 5: "Integrate into application"

You can then directly integrate and use the generated IEC function blocks in WAGO-I/O-*PRO* in your project.
To establish a connection between network devices, connect the inputs and outputs of your application, as required, with the configured network variables of the MODBUS network devices.

5 User Interface

5.1 Main View

There are four buttons on the user interface of the MODBUS Master Configurator. MODBUS slaves are listed on the right side next to the buttons in a display window if they have already been configured.
There are no entries in this list prior to MODBUS network configuration.

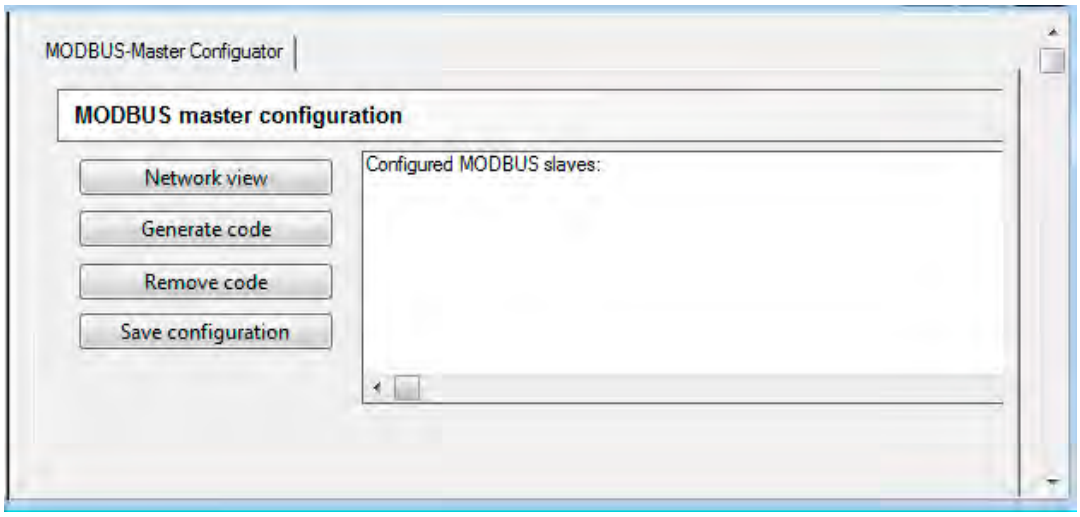


Figure 4: User interface of the MODBUS Master Configurator

Table 11: Legend to figure "User interface of the MODBUS Master Configurator"

Buttons	Description
[Network view]	Click this button to open the "**MODBUS Network**" dialog. Based on the target system selected in the WAGO-IO-*PRO* project, the MODBUS master is already displayed in the dialog list when first opened.
[Generate code]	Click this button to generate the respective IEC code from the configuration settings. Any MODBUS master program previously created is deleted and replaced by a new version.
[Remove code]	Click this button to remove the generated IEC code.
[Save configuration]	Click this button to save the configuration settings and close the "**MODBUS Network**" dialog.

5.2 "MODBUS network" Dialog

Click the **[Network view]** button to open the **MODBUS Network** dialog; here, the network is configured first and then the network devices and individual data points are configured.

When you first open the dialog, only the MODBUS master entry exists.

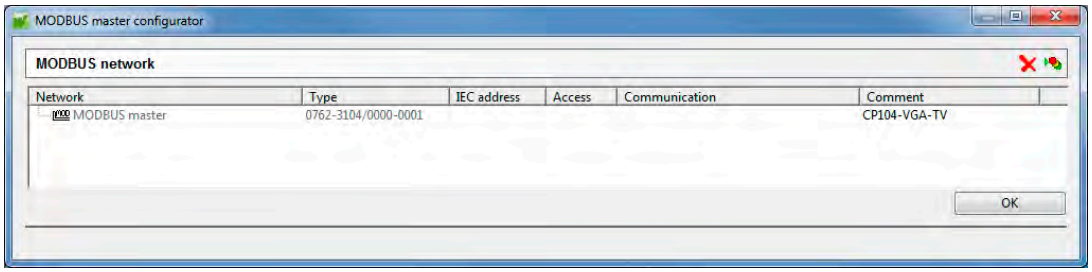


Figure 5: **MODBUS netzwork** dialog (without configuration)

After the network is configured, specific settings can be made for the individual data points of each network device entered.

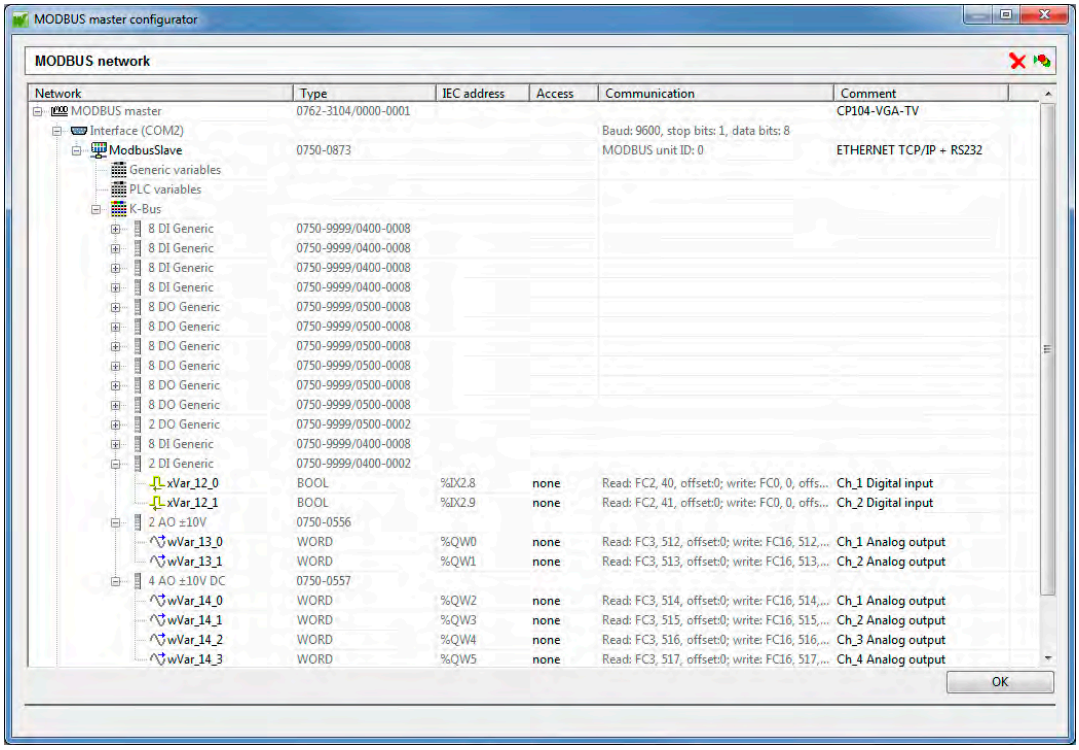


Figure 6: **MODBUS netzwork** dialog (with example configuration)

Table 12: Legend to figure "**MODBUS netzwork** dialog"

Table Entry	Description
Network	Tree structure entry for MODBUS master, interface, MODBUS slave or variable. The "**ModbusSlavex**" entries and the individual variables can be edited by double-clicking. Other settings can be made by right-clicking to call up the context menu for the respective entry.
Type	Specifying the MODBUS slave or variable. Select the variable entries in the selection list by double-clicking. Other settings can be made by right-clicking to call up the context menu for the respective entry.
IEC address	IEC address for data access.
Access	Type of data access (none [default], read, write) can be selected in the selection list by double-clicking. Other settings can be made by right-clicking to call up the context menu for the respective entry.
Communication	Selecting communication type (optimized [default] or explicit) is possible by right-clicking to call up the context menu for the respective entry.
Comment	Can be edited by double-clicking.
Buttons	**Description**
	Click this button to remove the generated code.
	Click this button to generate code from the configuration settings.
[OK]	Click this button to apply all settings and close the dialog.

5.2.1 Context Menu for the "MODBUS master" Entry

Select the "**MODBUS master**" entry and right-click to display the context menu.

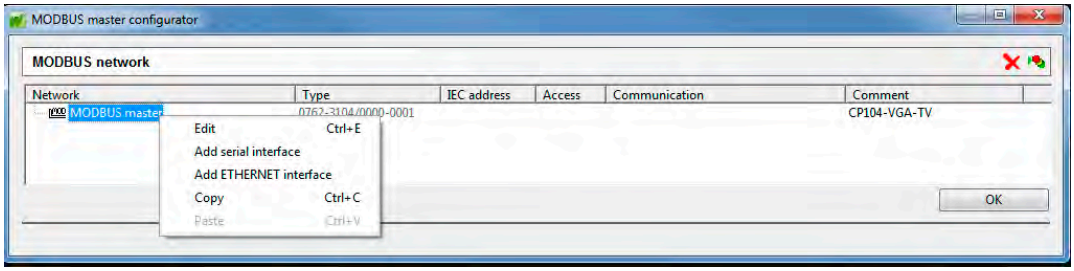


Figure 7: Context menu for the""**MODBUS master"** entry

Table 13: Legende to Figure "Context menu for the "**MODBUS master** entry"

Menu Item	Description
Edit	Opens the "**Parameter code generator**" dialog.
Add serial interface	In the tree structure, the "**Interface (COMx)**" subentry is added to the "**MODBUS Master**" entry.
Add ETHERNET interface	In the tree structure, the "**Interface ETHERNET**" subentry is added to the "**MODBUS Master**" entry.
Copy	No action
Paste	No action
Buttons	
[OK]	Click this button to apply all settings and close the dialog.

5.2.1.1 "Parameter code generator" Dialog

The **Parameter code generator** dialog is opened from the context menu for the MODBUS master, *"Edit"* menu item.

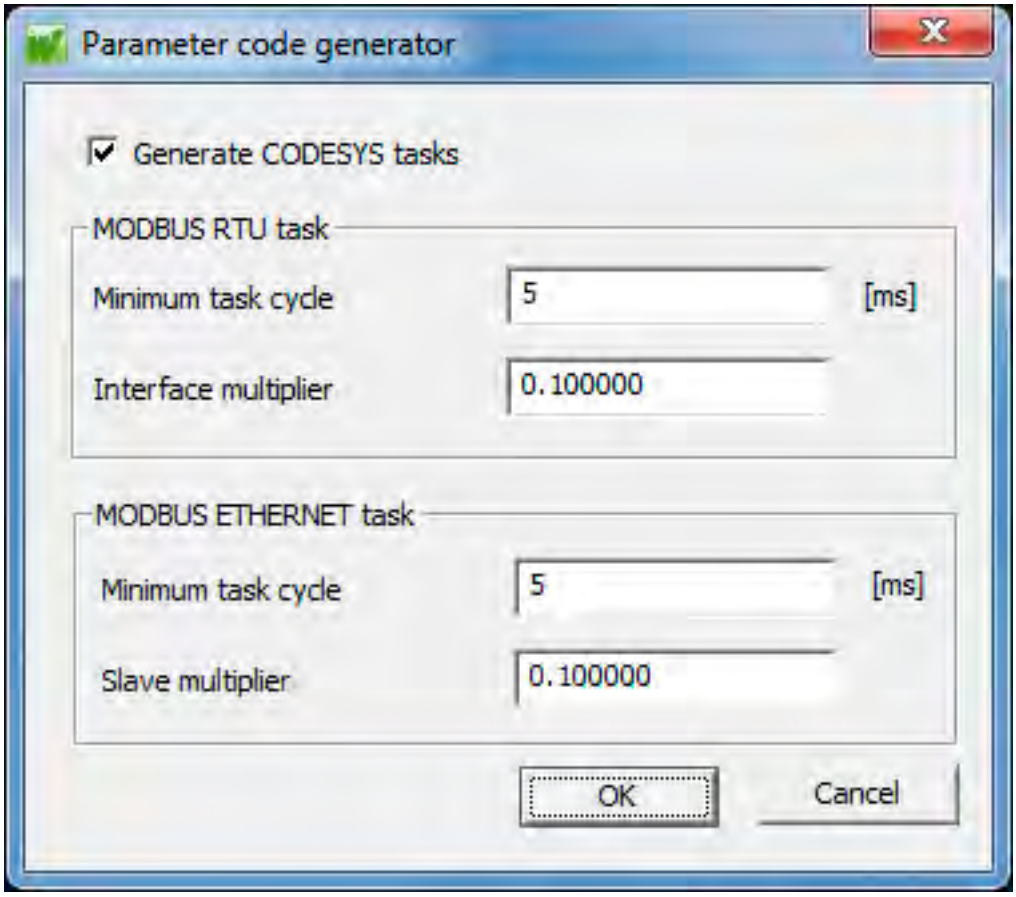


Figure 8: **Parameter code generator** dialog

Table 14: Legende to Figure "**Parameter code generator** dialog"

Entry	Default	Description
Generate CODESYS tasks	☑	☑ Enabled: Explicit tasks are created for the generated CODESYS function blocks. This implies that it is necessary to create a separate task for the PLC_PRG program to call up the function block tasks.
		☐ Disabled: No tasks are created for the generated CODESYS function blocks. CODESYS generates a separate task for the PLC_PRG program by default.
MODBUS RTU task		
Minimum task cycle	5 [ms]	Minimum task cycle setting. Can also be changed later in the task configuration.
Interface multiplier	0.100000	Multiplier of the task cycle time per inserted interface.
MODBUS ETHERNET task		
Minimum task cycle	5 [ms]	Minimum task cycle time setting. Can also be changed later in the task configuration.
Slave	0.100000	Multiplier of the task cycle time per added slave.
Buttons		
[OK]	Click this button to apply all settings and close the dialog.	
[Cancel]	Click this button to discard all settings and close the dialog.	

5.2.2 Context Menu for the "Interface (COMx)" Entry

Select the "**Interface (COMx)**" entry and right-click to display the context menu.

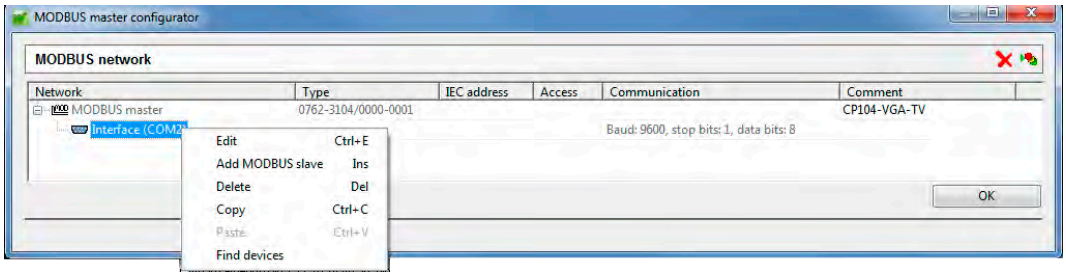


Figure 9: Context menu for the "**Interface (COMx)**" entry

Table 15: Legend to Figure "Context menu for the **Interface (COMx)** entry

Menu Item	Description
Edit	Open the "**Settings: Serial interface**" dialog.
Add MODBUS slave	Opens the "**MODBUS slave selection**" dialog.
Delete	The interface entry is deleted in the tree structure.
Copy	Copies the selected serial interface with all settings. The number of COM ports is automatically increased by one when inserted.
Paste	Inserts the copied serial interface with all settings. The number of COM ports is automatically increased by one when inserted.
Find devices	Opens the "**Search device**" dialog.

5.2.2.1 "Settings: Serial interface" Dialog

The **Settings: Serial interface** dialog is opened from the context menu for the serial interface, *"Edit"* menu item.

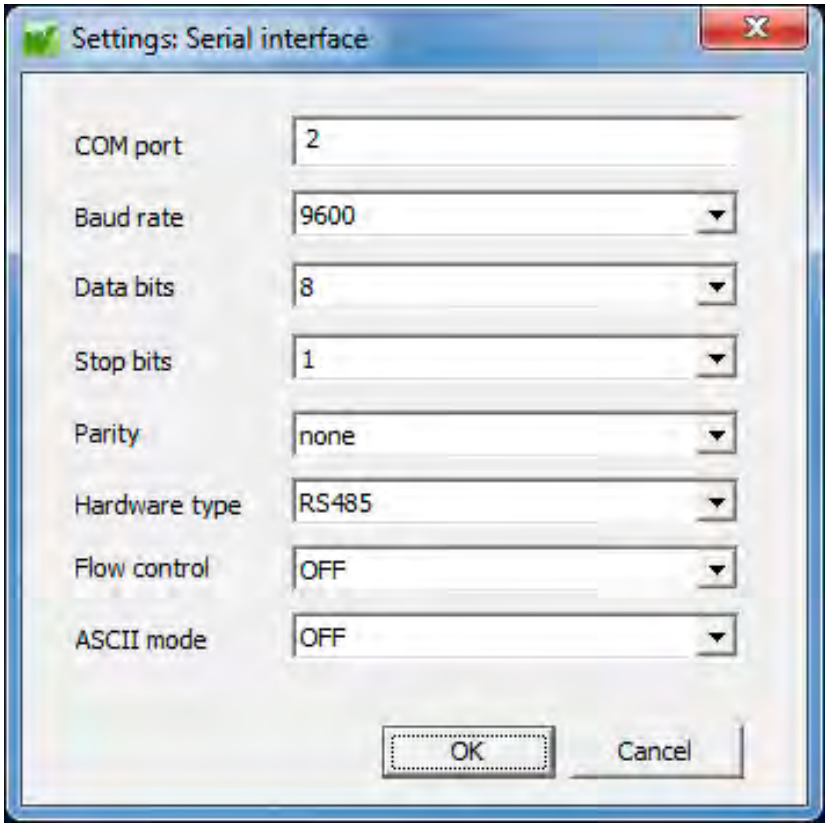


Figure 10: **Settings: Serial interface** dialog

Table 16: Legend to Figure "**Settings: Serial interface** dialog"

Entry	Default	Description
COM port	2	Free number entry of the COM port, automatically increased when inserting multiple serial ports.
Baud rate	9600	Selection list for the baud rate (110–115200).
Data bits	8	Selection list for the number of data bits (7, 8).
Stop bits	1	Selection list for the number of stop bits (1, 2)
Parity	none	Selection list for the parity (even, none, odd)
Hardware type	RS485	Selection list for the type of hardware interface (RS-485, RS-232).
Flow control	OFF	Selection list for the flow control (OFF, RTS/CTS)
ASCII mode	OFF	Selection list for enabling ASCII mode (active, OFF)
Buttons		
[OK]	Click this button to apply all settings and close the dialog.	
[Cancel]	Click this button to discard all settings and close the dialog.	

5.2.2.2 "MODBUS slave selection" Dialog

The **MODBUS slave selection** dialog is opened from the context menu for the "**Interface (COMx)**" or "**Interface (ETHERNET)**" entry, "*Add MODBUS slave*" menu item.
This dialog is also opened from the context menu for the "**ModbusSlave**" entry, "*Replace*" menu item.

The description of this dialog is available in the section "Context menu for the **ModbusSlave** entry" >> "**MODBUS slave selection** dialog".

5.2.2.3 "Search device" Dialog

The **Search device** dialog is opened from the context menu for the "**Interface (COMx)**" entry, "*Find device*" menu item.

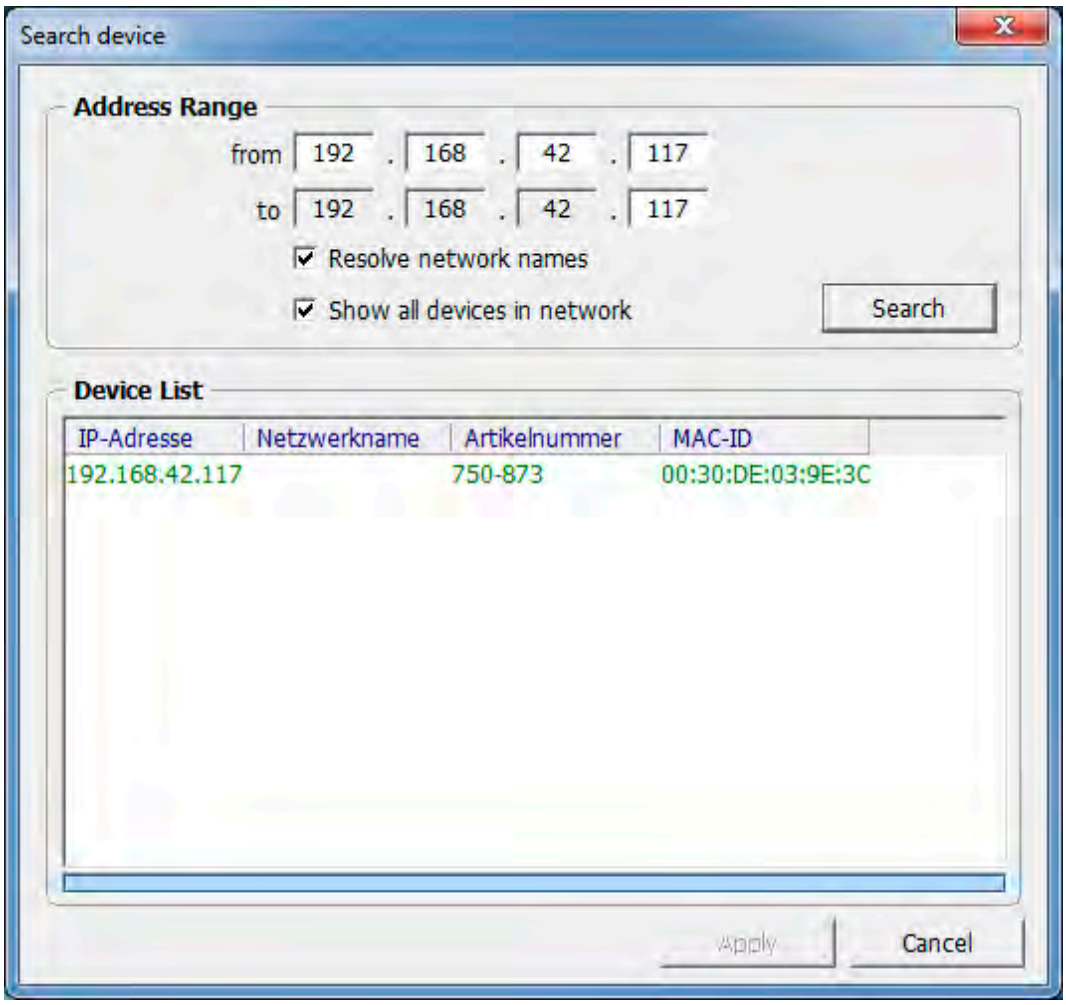


Figure 11: **Search device** dialog

Table 17: Legend to Figure " **Search device** dialog"

Entry	Default		Description
Address range, from … to …			Connected devices that are in the specified range of IP addresses should be searched for and displayed in the device list.
Resolve network names	☑	☑	Display names assigned by DHCP server or WAGO ETHERNET settings (a scan takes longer).
		☐	Names assigned by DHCP server or WAGO ETHERNET settings are not displayed.
Display all devices in the network	☐	☑	All connected devices should be searched and displayed in the device list.
		☐	Only connected devices that are in the specified range of IP addresses should be searched for and displayed in the device list.
[Browse]	Click this button to search for all connected devices, or connected devices within a specified range of IP addresses, and display in the device list.		
Device list	List of all devices found.		
IP address	IP address of the device found.		
Network name	Network name of the device found, if specified.		
Item number	Item number of the device found.		
MAC ID	MAC ID of the device found.		
Buttons			
[Apply]	Click this button to apply to all selected devices and close the dialog.		
[Cancel]	Click this button to discard all settings and close the dialog.		

5.2.3 Context Menu for the "Interface (ETHERNET)" Entry

Select the "**Interface (ETHERNET)**" entry and right-click to display the context menu.

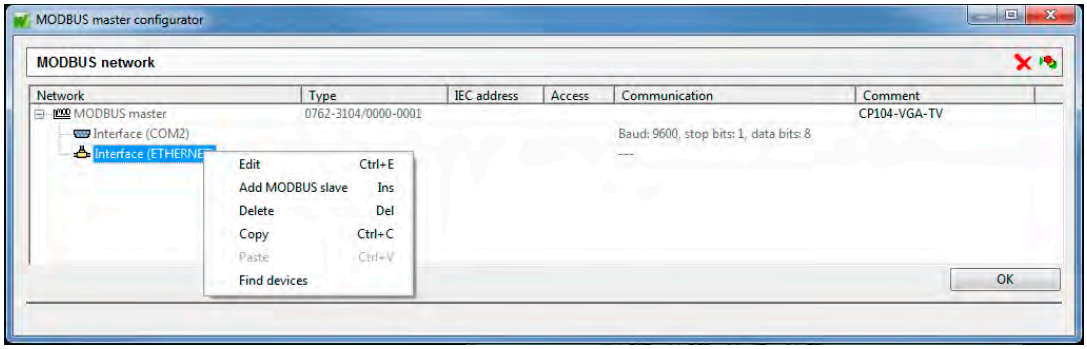


Figure 12: Context menu for the "**Interface (ETHERNET)**" entry

Table 18: Legend to Figure "Context menu for the **Interface (ETHERNET)** entry"

Menu Item	Description
Edit	ETHERNET interface has no settings.
Add MODBUS slave	Opens the "**MODBUS slave selection**" dialog.
Delete	The interface entry is deleted in the tree structure.
Copy	No action
Paste	No action
Find devices	Opens the "**Search device**" dialog.

5.2.3.1 "MODBUS slave selection" Dialog

The **MODBUS slave selection** dialog is opened from the context menu for the "**Interface (COMx)**" or "**Interface (ETHERNET)**" entry, "*Add MODBUS slave*" menu item.
This dialog is also opened from the context menu for the "**ModbusSlave**" entry, "*Replace*" menu item.

The description of this dialog is available in the section "Context menu for the **ModbusSlave** entry" >> "**MODBUS slave selection** dialog".

5.2.3.2 "Search device" Dialog"

The **Search Device** dialog is opened from the context menu for the "**Interface (ETHERNET)**" entry, "*Find device*" menu item.

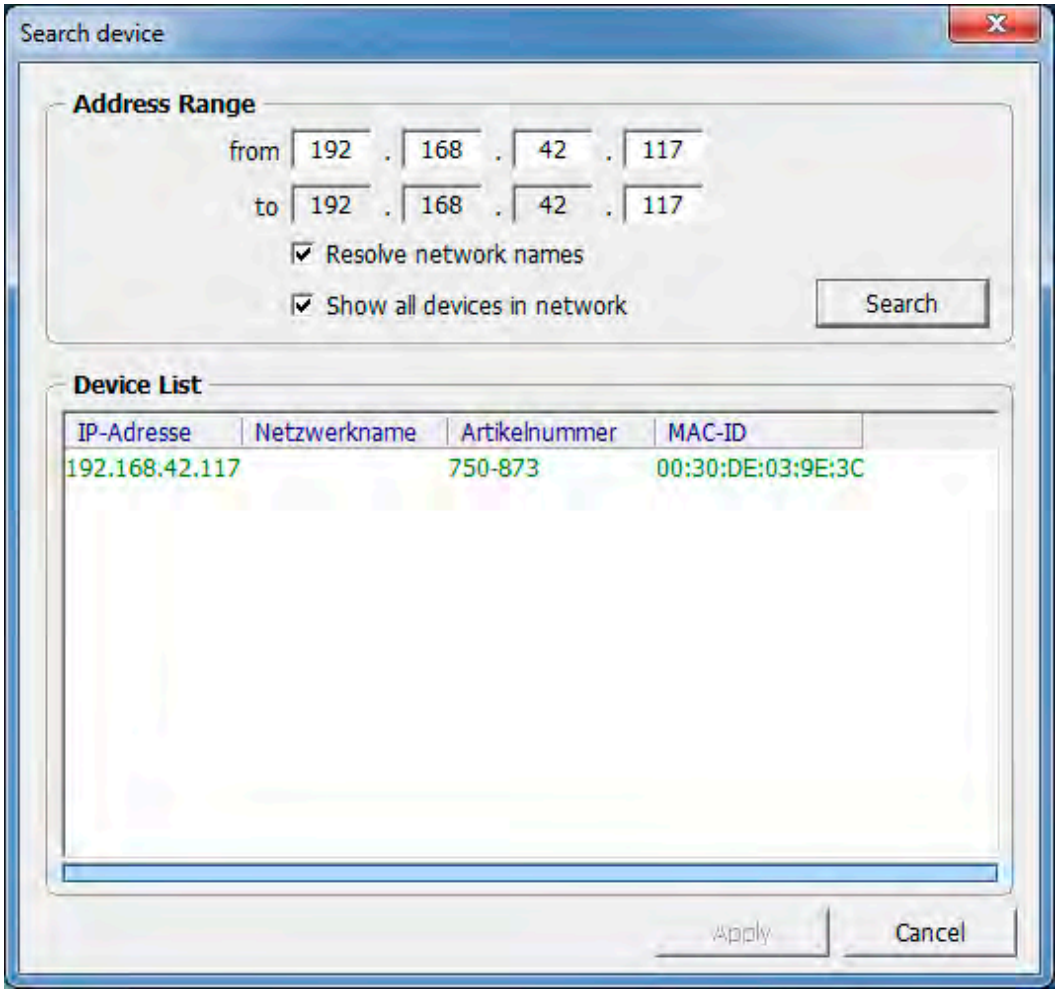


Figure 13: **Search device** dialog

Table 19: Legend to Figure "**Search device** dialog"

Entry	Default	Description
Address range, from … to …		Connected devices that are in the specified range of IP addresses should be searched for and displayed in the device list.
Resolve network names	☑	☑ Display names assigned by DHCP server or WAGO ETHERNET settings (a scan takes longer).
		☐ Names assigned by DHCP server or WAGO ETHERNET settings are not displayed.
Display all devices in the network	☐	☑ All connected devices should be searched and displayed in the device list.
		☐ Only connected devices that are in the specified range of IP addresses should be searched for and displayed in the device list.
[Browse]	Click this button to search for all connected devices, or connected devices within a specified range of IP addresses, and display in the device list.	
Device list	List of all devices found.	
IP address	IP address of the device found.	
Network name	Network name of the device found, if specified.	
Item number	Item number of the device found.	
MAC ID	MAC ID of the device found.	
Buttons		
[Apply]	Click this button to apply to all selected devices and close the dialog.	
[Cancel]	Click this button to discard all settings and close the dialog.	

5.2.4 Context Menü for the "ModbusSlave" Entry

Select the "**ModbusSlave**" entry and right-click to display the context menu.

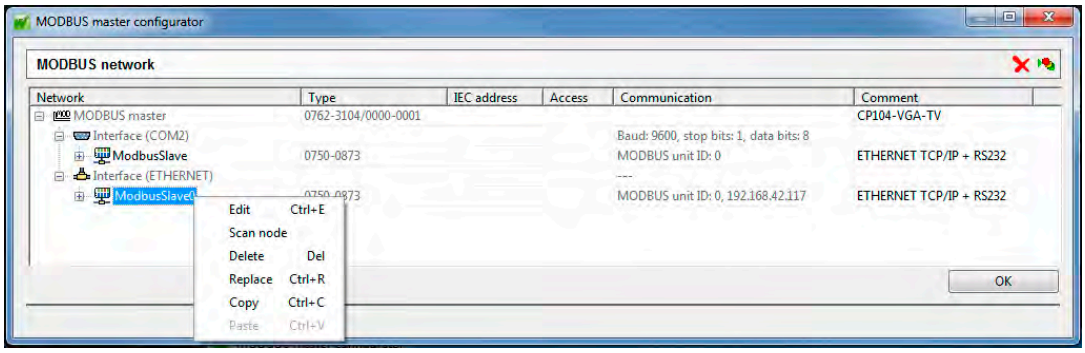


Figure 14: Context menu for the "**ModbusSlave**" entry

Table 20: Legend to Figure "Context menu for the **ModbusSlave** entry"

Menu Item	Description
Edit	Opens the "**Communication settings**" dialog.
Scan node	If the MODBUS slave can be accessed online, the components and process data variables of the connected fieldbus node are scanned and listed in the tree structure under the "**ModbusSlave**" entry → "**K-Bus**" subentry.
Delete	The MODBUS slave entry is deleted from the tree structure.
Replace	Opens the "**MODBUS slave selection**" dialog.
Copy	Copies the selected MODBUS slave and all its settings. The number of MODBUS slaves is automatically increased by one when inserted.
Paste	Inserts the copied MODBUS slave and all its settings. The number of MODBUS slaves is automatically increased by one when inserted.

5.2.4.1 "Communication settings" – Interface (COMx) Dialog)

The **Communication settings** dialog is opened from the context menu for the "**ModbusSlave**" entry "*Edit*" menu item.

The "**MODBUS RTU settings**" tab provides the settings for access from the connected MODBUS master to the MODBUS slave via the serial interface.

The "**Service settings**" tab provides the settings for access during maintenance from a connected service PC to the MODBUS slave.

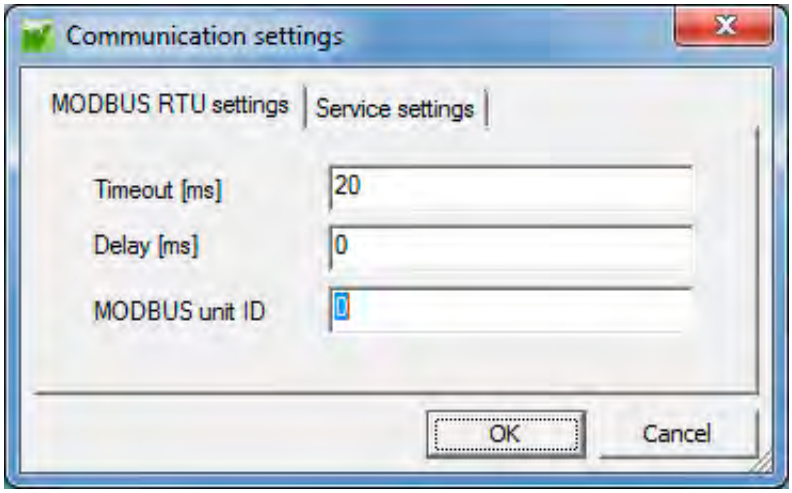


Figure 15: **Communication settings** dialog [for Interface (COMx)]

Table 21: Legende to Figure "**Communication settings** dialog [for Interface (COMx)]**)**"

"MODBUS RTU settings" Tab		
Entry	**Default**	**Description**
Time out [ms]	20	Time span in which the response is expected.
Delay [ms]	0	Delay in [ms] between two MODBUS requests. Prevents overloading of the slaves. When '0', the next request is made immediately after receiving the response.
MODBUS unit ID	0	The MODBUS unit ID of the slave must be unique for all devices in the line.
"Service settings" Tab		
Entry	**Default**	**Description**
ETHERNET	○	◉ Access enabled from the service PC to the MODBUS slave via the ETHERNET interface.
		○ Access enabled from the service PC to the MODBUS slave via the serial interface.
Service Interface	◉	◉ Access enabled from the service PC to the MODBUS slave via the serial interface; select interface via selection list.
		○ Access disabled from the service PC to the MODBUS slave via the serial interface.
Buttons		
[OK]	Click this button to apply to all selected devices and close the dialog.	
[Cancel]	Click this button to discard all settings and close the dialog.	

5.2.4.2 "Communication settings" – Interface (ETHERNET Dialog)

The **Communication settings** dialog is opened from the context menu for the "**ModbusSlave**" entry,"*Edit*" menu item.

The "**MODBUS ETHERNET settings**" tab provides the settings for access from the connected MODBUS master to the MODBUS slave via the ETHERNET interface.

The "**Service settings**" tab provides the settings for access during maintenance from a connected service PC to the MODBUS slave.

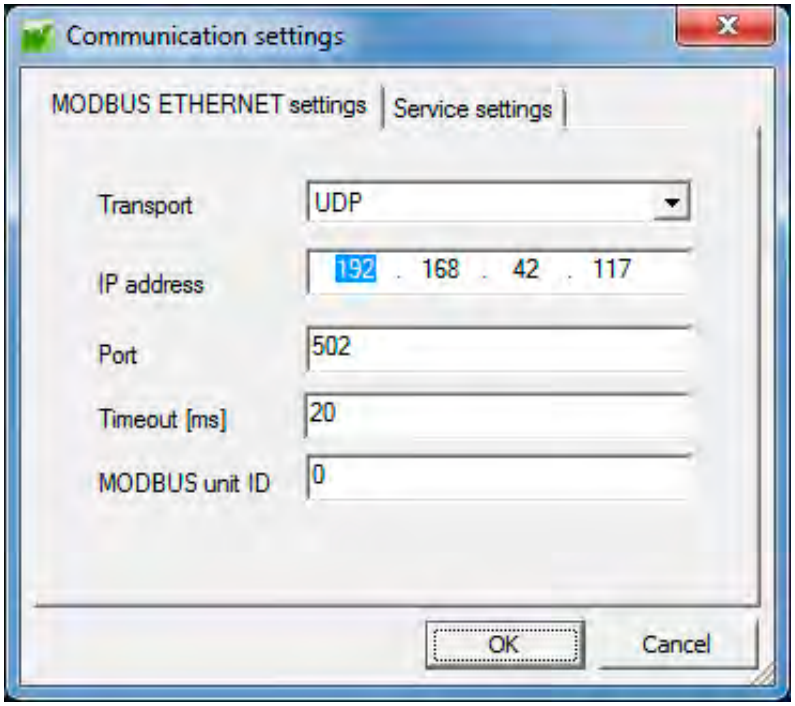


Figure 16: **Communication settings** dialog [for Interface (ETHERNET)]

Table 22: Legend to Figure " **Communication settings** dialog [for Interface (ETHERNET)]"

"MODBUS RTU settings" Tab		
Entry	**Default**	**Description**
Transport	UDP	The UDP protocol is used for transport via the ETHERNET interface.
		The TCPP protocol is used for transport via the ETHERNET interface.
IP address	0.0.0.0	IP address of the slave
Port	502	Interface port
Time out [ms]	20	Time span to wait for the response.
MODBUS unit ID	0	The MODBUS unit ID of the slave must be unique for all devices in the line.
"Service settings" Tab		
Entry	**Default**	**Description**
ETHERNET	◉	◉ Access enabled from the service PC to the MODBUS slave via the ETHERNET interface.
		○ Access enabled from the service PC to the MODBUS slave via the serial interface.
Service Interface	○	◉ Access enabled from the service PC to the MODBUS slave via the serial interface; select interface via selection list.
		○ Access disabled from the service PC to the MODBUS slave via the serial interface.
Buttons		
[Apply]	Click this button to apply to all selected devices and close the dialog.	
[Cancel]	Click this button to discard all settings and close the dialog.	

5.2.4.3 "MODBUS slave selection" Dialog

The **MODBUS slave selection** dialog is opened from the context menu for the "**Interface (COMx)**" or "**Interface (ETHERNET)**" entry, "*Add MODBUS slave*" menu item.
This dialog is also opened from the context menu for the "**ModbusSlave**" entry, "*Replace*" menu item.

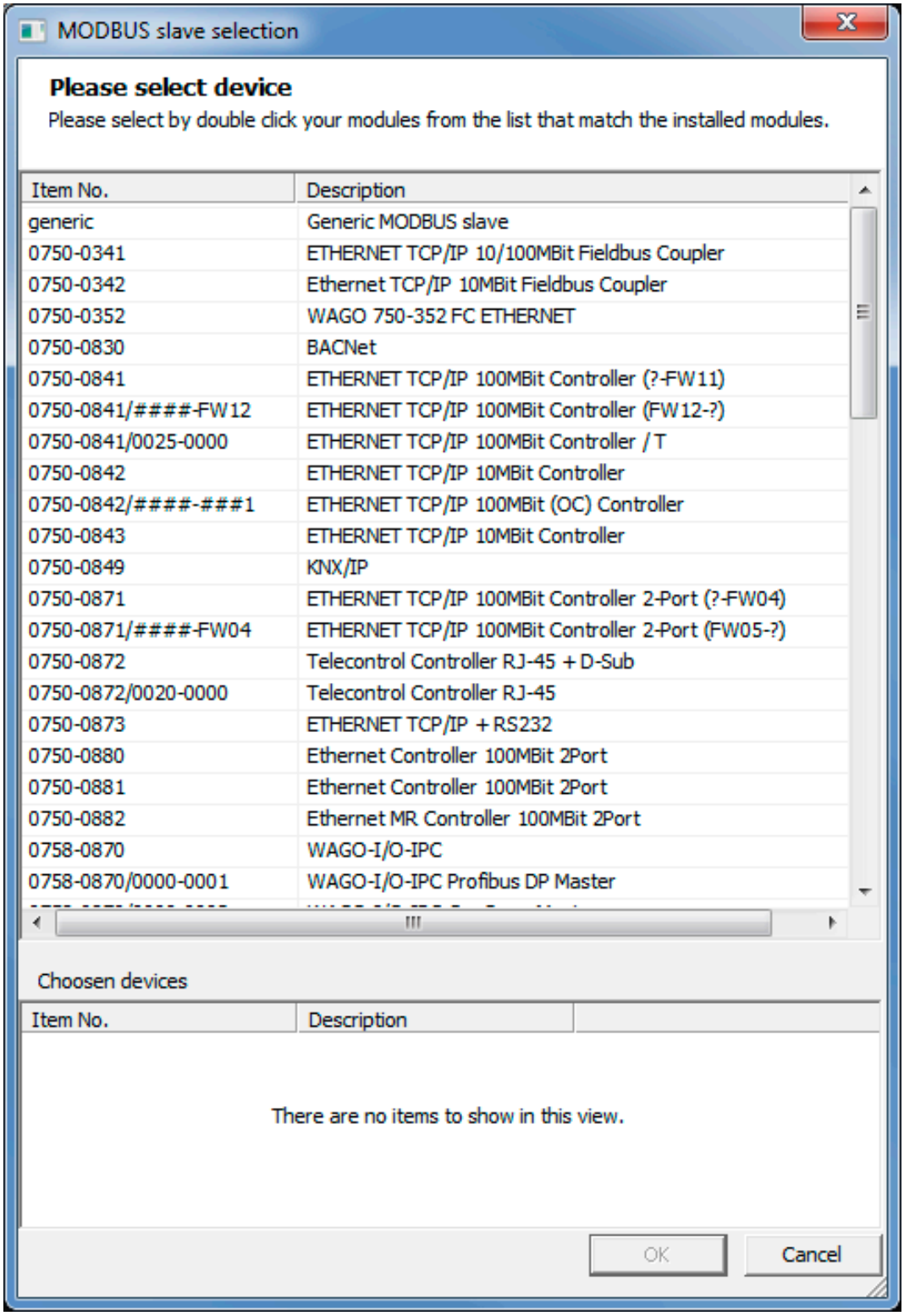


Figure 17: **MODBUS slave selection** Dialog

Table 23: Legend to Figure " **MODBUS slave selection** Dialog"

Entry	Description
Item No. **generic** **075x-xxx/xxx**	Item number of the MODBUS slave available Generic MODBUS slave to integrate the MODBUS slave from a third-party manufacturer. Item number of the WAGO MODBUS slave
Description	Item description of the MODBUS slave
Buttons	
[OK]	Click this button to apply to all selected devices and close the dialog.
[Cancel]	Click this button to discard all settings and close the dialog.

5.2.5 Context Menu for the "Generic variables" Entry

Select the "**Generic Variables**" entry and right-click to display the context menu.

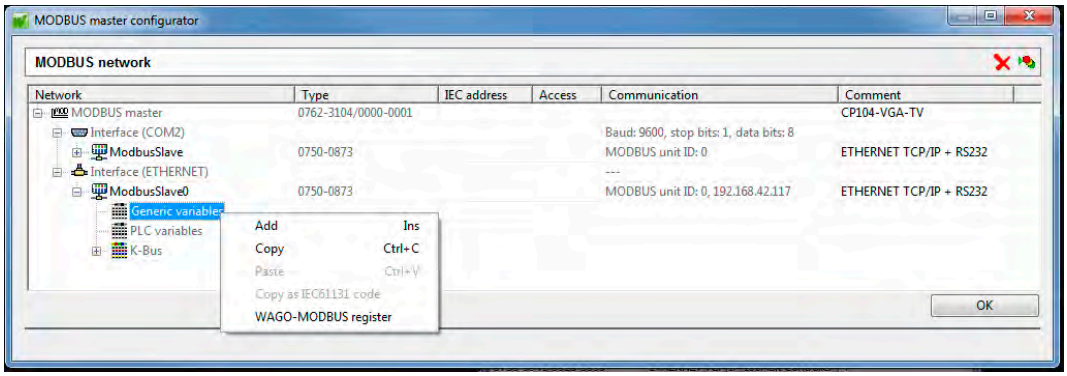


Figure 18: Context menu for the "**Generic variables**" entry

Table 24: Legend to Figure " Context menu for the **Generic variables** entry "

Menu Item	Description
Add	Adds a generic variable as a subentry with the following table entries that can be changed by double-clicking: - Network: "xNewVar" (editable) - Type: "Bool" (default) (selection list) - Access: "none" (default), (selection list: "read/write") - Communication: "Read:FC0, 0, Offset:0; Write:FC0, 0, Offset:0".
Copy	Copies the selected generic variables with all settings.
Paste	Inserts the copied generic variables with all settings.
WAGO MODBUS register	Opens the "**WAGO MODBUS register**" dialog.

5.2.5.1 "WAGO MODBUS register" Dialog

The **WAGO MODBUS register** dialog is opened from the context menu for the "**Generic variables**" entry, "*WAGO MODBUS register*" menu item.

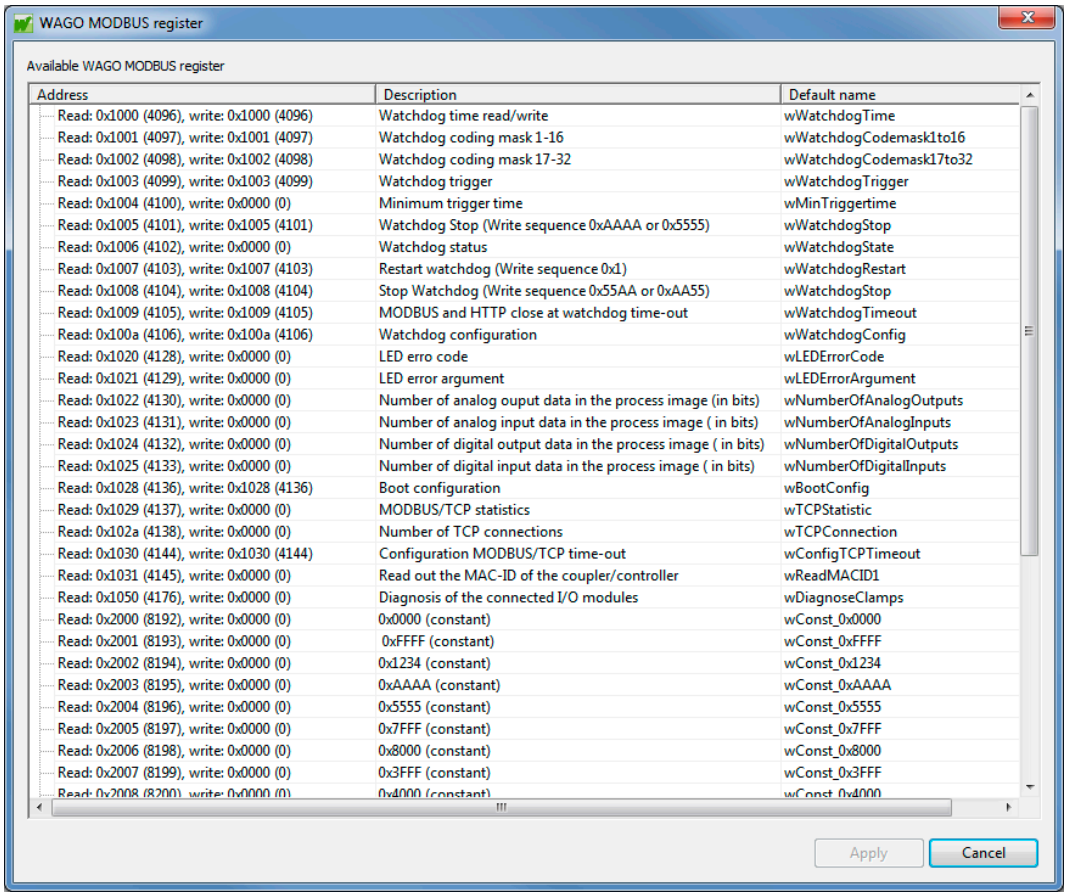


Figure 19: **WAGO MODBUS register** dialog

Table 25: Legend to Figure "**WAGO MODBUS register**" dialog

Entry	**Description**
Address	Register address for read and write range of the WAGO MODBUS register entries.
Description	Description of the register entry
Default name	Default name of the variables for the corresponding register
Buttons	
[Apply]	Click this button to apply all selected MODBUS registers as generic variables and close the dialog.
[Cancel]	Click this button to discard all settings and close the dialog.

5.2.6 Context Menu for the "PFC variables" Entry

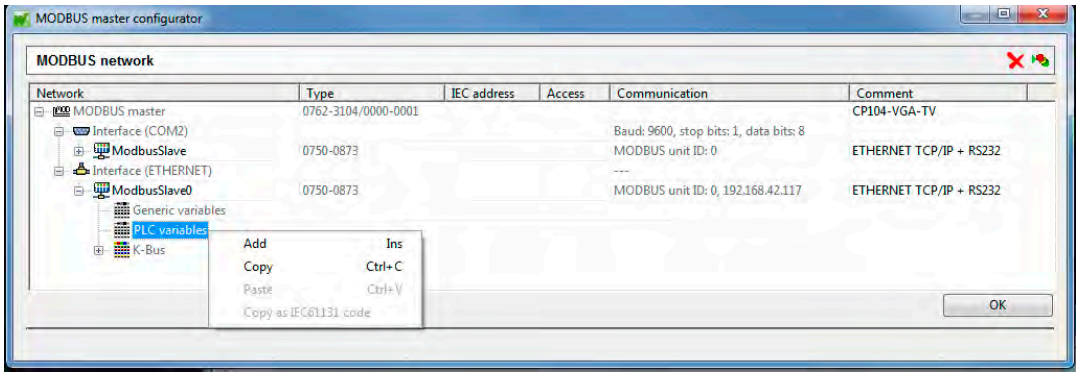


Figure 20: Context menu for the "**PFC variables**" entry

Table 26: Legend to Figure " Context menu for the "**PFC variables**" entry"

Menu Item	Description
Add	Adds a PFC variable as a subentry with the following table entries that can be changed by double-clicking: - Network: "xNewVar" (editable) - Type: "Bool" (default) (selection list) - IEC address: "%MX0.0" (default), (type-dependent) - Access: "none" (default) (selection list: "read/write") - Communication: "Read:FC1, 12288, Offset:0; Write:FC15,12288, Offset:0". (default), (FC depends on the variable type)
Copy	Copies the selected PFC variables with all of the settings.
Paste	Inserts the copied PFC variables with all of the settings.
Copy as IEC61131 code	Copies the selected entries as IEC 61131 code to the clipboard, enabling slave variable definitions to be transferred to the slave CODESYS program.
	Note **You can also copy IEC code from CODESYS!** To do so, select the variable definitions in CODESYS program, copy them to the clipboard and insert them in the PFC variable list via the *"Insert"* menu item. The Master Configurator can then generate corresponding entries in the tree structure. You can also paste the variable definitions from the clipboard into another instance of the MODBUS Master Configurator.

5.2.7 Context menu for the "K-Bus" Entry

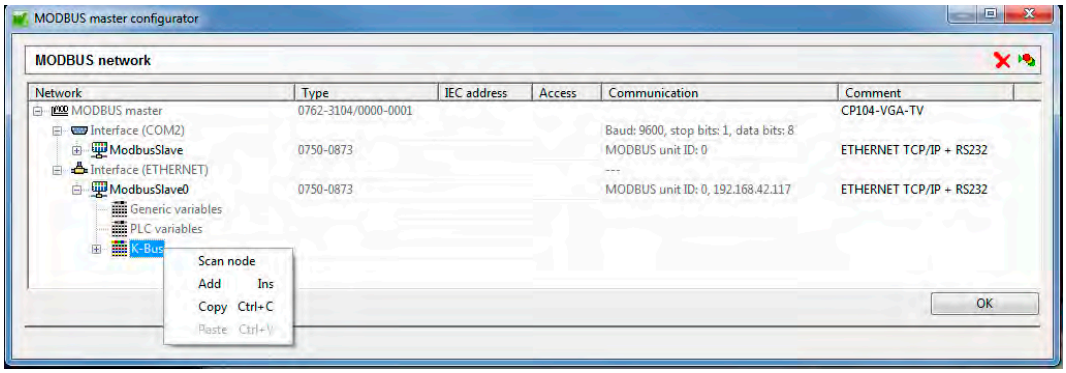


Figure 21: Context menu for the "**K-Bus**" entry

Table 27: Legend to Figure "Context menu for the **K-Bus** entry"

Menu Item	Description
Scan node	If the MODBUS slave can be accessed online, the components and process data variables of the connected fieldbus node are scanned and their configurations listed as subentries.
Add	The "**Module selection**" dialog appears.
Copy	Copies all configured I/O modules to the marked internal data bus with all settings.
Paste	Inserts the copied I/O module configuration with all settings in the selected internal data bus.

5.2.7.1 "Module selection" Dialog

The **Module selection** dialog is opened from the "**K-Bus** " context menu, *"Add"* menu item.

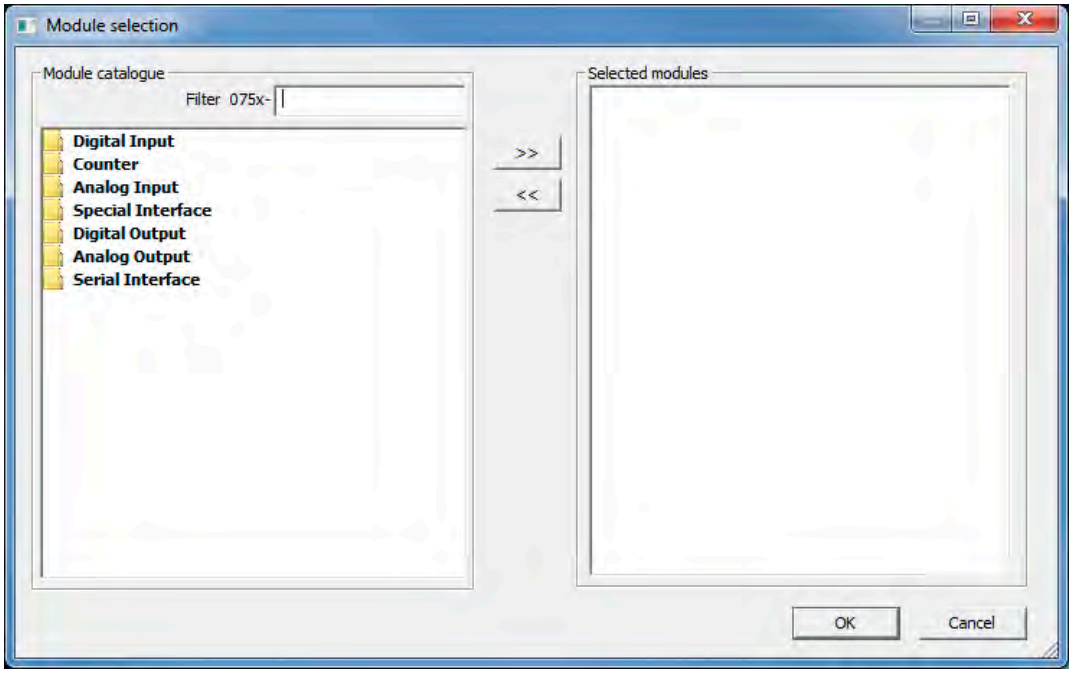


Figure 22: **Module selection** dialog

Table 28: Legend to Figure " **Module selection** dialog"

Module catalogue	
Entry	**Description**
Filter 075x-____	Input field to filter the selection of I/O modules by specific item numbers.
Digital Input **Counter** **Analog Input** **Special Interface** **Digital Outpup** **Analog Output** **Serial Interface**	Folders that contain the item number of the respective I/O modules as subentries for selection.
Selected modules	
Entry	**Description**
____	After applying the selected I/O modules from the "**Module catalogue**" on the left side, the I/O modules are displayed in the "**Selected modules**" on the right side. By clicking **[OK]** to confirm, the I/O modules are then inserted in the tree structure of the network configuration under the "**K-Bus**" entry.
Buttons	
[>>]	Click this button to move the modules selected in the "Module catalogue" area on the left side to the "Selected modules" area on the right side.
[<<]	Click this button to remove all the modules marked in the "Selected modules" on the right side.
[OK]	Click this button to apply all settings and close the dialog.
[Cancel]	Click this button to discard all settings and close the dialog.

6 Operation

1. Make the network presets.
2. Insert the required interfaces.
3. Configure your interfaces.
4. Insert your MODBUS slaves.
5. Configure the communication settings for your MODBUS slaves.
6. Configure the hardware for your MODBUS slaves.
7. Insert the required generic variables and PFC variables.
8. Configure the required generic variables, PFC variables and internal data bus variables.
9. Generate the IEC code from your configuration settings.

6.1 Make Network Presets

1. In the main view of the MODBUS Master Configurator, click the **[Network View]** button.
   The **MODBUS network** dialog opens.
2. Select the "**MODBUS master**" entry right-click.
   The context menu opens.
3. Click "*Edit*" in the context menu.
   The **Parameter code generator** dialog opens.
4. In the dialog, enable the "Generate CODESYS tasks" function to generate IEC code from the configuration settings.
5. In the dialog, change the minimum task cycle time for the MODBUS RTU task and interface multiplier if desired.
   You can also change the minimum task cycle time for the MODBUS ETHERNET task and the multiplier per configured slave if desired.
6. Click **[OK]** to confirm your settings.
   The **Parameter code generator** dialog closes.

6.2 Add Interface

6.2.1 Add Serial or ETHERNET Interface

1. Select the "**MODBUS master**" entry and right-click.
   The context menu opens.

2. In the context menu, click the "*Add serial interface*" menu item if you want to communicate to your MODBUS master via the serial interface.
   The entry for the serial interface is displayed in the tree structure.

3. In the context menu, click the "Add *ETHERNET interface*" menu item if you want to communicate to your MODBUS master via ETHERNET.
   The entry for the ETHERNET interface is displayed in the tree structure.

6.2.2 Configure Serial Interface

1. Select the "**Interface (COMx)**" entry and right-click.
   The context menu opens.

2. In the context menu, click the "*Edit*" menu item.
   The **Settings: Serial interface** dialog opens.

3. In the dialog for the serial interface, if desired, change the number for the COM port, select the baud rate, number of data bits and stop bits, parity, hardware type and whether flow control and ASCII mode should be enabled.

4. Click **[OK]** to confirm your settings.
   The **Settings: Serial interface** dialog closes.

6.3 Insert and Configure MODBUS Slave

6.3.1 Insert MODBUS Slave Online

1. Select the "**Interface (COMx)**" or "**Interface (ETHERNET)**" entry and right-click.
   The context menu opens..

2. If you can access your MODBUS slaves online, click the "*Find devices*" menu item in the context menu.
   The **Search device** dialog opens..

3. In the dialog, change the IP addresses for the address range according to the IP addresses of the devices that are online.
   If necessary, enable the "Resolve network names" and "Display all devices in the network" functions.

4. Confirm your settings and click **[Browse]** to enable the search.
   The devices found that are accessible online are displayed in the device list.

5. Select the devices found in the device list and click **[Apply]**.
   The **Search device** dialog closes and the MODBUS slaves are applied in the network tree structure as subentries.

6.3.2 Insert MODBUS Slave Offline

1. If configuring a MODBUS slave offline, click the "Add *MODBUS slave*" menu item in the context menu.
   The **MODBUS slave selection** dialog opens.

2. Mark the required devices in the selection list and double-click in the "Choosen devices" list below to apply.

3. Mark the selected devices in the list below and click **[OK]**.
   The **MODBUS slave selection** closes and the MODBUS slaves are applied in the network tree structure as subentries.

6.3.3 Configure MODBUS Slave

1. Select the "**ModbusSlave**" entry and right-click.
   The context menu opens.

2. In the context menu, click the "*Edit*" menu item.
   The **Communication settings** dialog opens.
   The setting options for MODBUS slave communication depend on the type of interface your MODBUS slave is attached to.

6.3.3.1 Configure MODBUS Slave via Serial Interface

1. If your MODBUS slave is connected to a serial interface, select the "MODBUS RTU settings" tab in the dialog.

2. If desired, change the time values for "Time-out", "Delay [ms]" and "MODBUS unit ID".

3. To set direct access to the MODBUS slave from a service PC, select the "Service settings" tab in the dialog.

4. If desired, change the option for access via the ETHERNET address or the required serial interface.

5. Click **[OK]** to confirm your settings.
   The **Communication settings** dialog closes.

6.3.3.2 Configure MODBUS Slave via ETHERNET Interface

1. If your MODBUS slave is connected to an ETHERNET interface, select the "MODBUS ETHERNET settings" tab in the dialog.

2. If desired, change the type for "Transport", "IP address", "Port", time value for "Timeout [ms]" and the "MODBUS unit ID".

3. To set direct access to the MODBUS slave from a service PC, select the "Service settings" tab in the dialog.

4. If desired, change the option for access via the ETHERNET address or the required serial interface.

5. Click **[OK]** to confirm your settings.
   The **Communication settings** dialog closes.

6.3.4 Configure WAGO MODBUS Slave Hardware Online

1. Select the "**ModbusSlave**" entry and right-click.
   The context menu opens.

2. If you can access your MODBUS slave online, click the "*Scan node*" menu item in the context menu.
   All data points for the connected I/O modules in the fieldbus node are displayed in the tree structure under the "**K-Bus** " entry as internal data bus variables.

6.3.5 Configure WAGO MODBUS Slave Hardware Offline

1. For offline configuration, select the "**K-Bus**" subentry under the "**ModbusSlave**" entry and right click.
   The context menu opens.

2. In the context menu, click the "*Add*" menu item.
   The **Module selection** dialog appears.

3. In the "Module catalogue" area on the left side, double-click the modules that are in your fieldbus node as I/O modules.
   After double-clicking, the modules are displayed in the "Selected modules" area on the right.

4. Click **[OK]** to confirm your settings.
   The **Module selection** dialog closes and all data points for the selected I/O modules are displayed in the tree structure under the "**K-Bus**" entry as internal data bus variables.

6.4 Insert and Configure Variables

6.4.1 Add Generic Variables

1. Select the "**Generic variables**" subentry under the "**ModbusSlave**" entry and right-click.
   The context menu opens.

2. In the context menu, click the "*Add*" menu item.
   A variable of the "BOOL" type, without access, is displayed under the "**Generic variables**" entry.

6.4.2 Configure Generic Variables

1. In the tree structure, select the entry for the generic variable that you want to configure. Double-click the entry to edit the name of the variables, if necessary.

2. In the "Type" column for the generic variable, select the data type and double-click.
   A selection list with data types is displayed.

3. Select the required data type for the generic variable.
   The function code for MODBUS access appropriate for the selected data types is automatically displayed in the "IEC Address" or "Communication" column.

4. In the "Access" column for the generic variable, select the default entry "none" and double-click.
   A selection list with the options "read", "write" and "none" is displayed.

5. Select the access required for the generic variable.

6. Select the field in the "Comment" column.
   Double-click to edit the field and to enter a comment, if necessary.

7. In the tree structure, select the subentry of one desired variable and right-click to open the context menu.

8. In the context menu, select the "*Number of items*" menu item.
   The **Enter the number of items** dialog opens.

9. Enter the required element count in the dialog and click **[OK]** to confirm your settings.
   The **Enter the number of items** dialog closes.

10. In the context menu, click the "*Edit address*" menu item.
    The **MODBUS address** dialog opens.

11. In the dialog, select the required function code to read and write variables from the selection list and click **[OK]** to confirm your settings.
    The **MODBUS address** dialog closes.

12. In the context menu, select the "*Communication*" menu item and if required, select the "E*xplicit*" or "O*ptimized*" menu item.
    Optimized communication is set by default.
    Variables with explicit communication are displayed in red font.

13. If required, configure other variables in the same way.

6.4.3 Add PFC Variables

1. Select the "**PFC variables**" subentry under the "**ModbusSlave**" entry and right-click.
   The context menu opens.

2. In the context menu, click the "*Add*" menu item.
   A "BOOL" type variable, without access, is displayed under the "**PFC variables**" entry.

6.4.4 Configure PFC Variables

1. In the tree structure, select the entry for the PFC variable that you want to configure. Double-click the entry to edit the name of the variables, if necessary.

2. In the "Type" column for the PFC variable, select the data type and double-click.
   A selection list with data types is displayed.

3. Select the required data type for the PFC variable.
   The address string for IEC addressing of the selected data type, as well as the appropriate function code for MODBUS access, are automatically displayed in the "IEC address" or "Communication" column.

4. In the "IEC address" column for the PFC variable, select the IEC address and double-click.
   You can set the address as required.

5. In the "Access" column for the PFC variable, select the default entry "none" and double-click.
   A selection list with the options "read", "write" and "none" is displayed.

6. Select the access required for the PFC variable.

7. Select the field in the "Comment" column.
   Double-click to edit the field and enter a comment, if necessary.

8. In the tree structure, select the subentry of one desired variable and right-click to open the context menu.

9. In the context menu, select the "*Communication*" menu item and if required, select the "*Explicit*" or "*Optimized*" menu item.
   Optimized communication is set by default.
   Variables with explicit communication are displayed in red font.

10. In the context menu, select the "*Copy as IEC 61131 code*" menu item.
    The PFC variable settings are copied to the clipboard as IEC code, so that you can directly paste them into the variable declaration of your IEC project.

11. If required, configure other variables in the same way.

6.4.5 Configure K-Bus Variables

1. In the tree structure, select under the K-Bus entry the desired subentry for the internal data bus variable that you want to configure. Double-click the entry to edit the name of the variables, if necessary.

2. In the "Access" column for the internal data bus variable, select the default entry "none" and double-click.
   A selection list with the options "read", "write" and "none" is displayed.

3. Select the access required for the internal data bus variable.

4. Select the field in the "Comment" column.
   Double-click to edit the field and enter a comment, if necessary.

5. In the tree structure, select the subentry of one desired variable and right-click to open the context menu.

6. In the context menu, select the "*Communication*" menu item and if required, select the "E*xplicit*" or "O*ptimized*" menu item.
   Optimized communication is set by default.
   Variables with explicit communication are displayed in red font.

7. If required, configure other variables in the same way.

8. Click **[OK]** to confirm all settings.
   The **MODBUS network** dialog closes.

6.5 Generate IEC Code

1. In the main view of the MODBUS Master Configurator, click the **[Generate code]** button to automatically generate the IEC code from your configuration settings.

2. You can then use the generated function blocks in your application program in WAGO-I/O-*PRO*.
   Once in WAGO-I/O-*PRO*, go to the "POUs" tab.
   The program modules are created there.

Information

More information about WAGO-I/O-*PRO* function blocks!
Detailed information about libraries and descriptions of the function blocks available in WAGO I/O-*PRO* by code generation are in the appendix.

7 Glossary

Information

More information about MODBUS technical terms!
Detailed information about MODBUS-specific terms and technologies is available on the MODBUS website at:
www.modbus.org

C

CODESYS 2.3
A programming environment based on the IEC 61131-3 standard.

CODESYS Task
Subprocesses that execute specific POUs (Program Organization Units) cyclically and after events.

D

DTU
Date type objects in CODESYS (STRUCT, ENUM).

E

EXP Format
This format uses the CODESYS 2.3 environment to export project elements in a human-readable format.

F

Function
Module that always returns the same results (as a function value). It has no local variables that store values beyond an invoke.

Function block
Module that returns one or more values when executed. It can be saved as a local variable ("memory").

G

Generic Variable
Network variable used to address and employ devices from third-party manufacturers in a network as MODBUS slaves.

I

IEC 61131-3
International standard for modern systems with PLC functionality created in 1993. Based on a structured software model, it defines a series of powerful programming languages to be utilized for different automation tasks.

K

K Bus
Internal data bus in the fieldbus node of the WAGO-I/O-SYSTEM 750.

L

Library
Collection of modules available to the programmer in the WAGO-I/O-*PRO* programming tool (CODESYS) for creating control programs that adhere to IEC 61131-3.

M

MODBUS
Communication protocol standardized in IEC 61158 - CPF15.

MODBUS ASCII
MODBUS protocol that transfers data in ASCII code. This type of communication is not prevalent.

MODBUS Master
Node in the MODBUS network that can make active communication requests to other nodes.

MODBUS Protocol
For communication between MODBUS master and slave, various MODBUS protocols with different data transmission formats can be selected, such as MODBUS ASCII, MODBUS RTU, MODBUS TCP and MODBUS UDP.

MODBUS RTU
MODBUS protocol that transfers data in a binary format. Next to MODBUS TCP, this protocol is used most often.

MODBUS Slave
Node in the MODBUS network that reacts and responds to incoming communication requests.

MODBUS TCP

MODBUS protocol that functions as the RTU implementation. The main difference between MODBUS RTU and TCP is the transmission of data to the TCP / IP stack. This permits communication via the Internet, if necessary, and is not limited by the cable length. Port 502 is reserved for MODBUS. The MODBUS TCP implementation is currently being specified as a standard (IEC PAS 62030 [pre-standard]).

MODBUS UDP

MODBUS protocol that transfers data in a binary format. Next to MODBUS TCP, this protocol is used most often.

Module

Modules consist of functions, function blocks and programs.
Every module is made up of a declaration part and a body. The body is written in one of the IEC programming languages STL (statement list), ST (structured text), AS (process structure), FUP (function plan) or KOP (coupling plan).

N

Network Variable

A network variable is a type-related variable in IEC 61131 programming for implementing logical communication channels and links between the network nodes.
This can be associated with one or more network variables of one or more network nodes.

P

POU

Abbreviation for Program Organization Unit, which are program objects in CODESYS (PRG, FB).

S

Special Feature Register

MODBUS registers from address 0x1000 to 0x2052. With WAGO devices, these registers contain additional information about the present WAGO fieldbus node. In most WAGO applications, this range cannot be considered contiguous flat memory.

8 Appendix

This appendix contains information about the libraries and function blocks available in WAGO-I/O-*PRO* after code generation, as well as descriptions of data access via the function codes.

Information

More information about WAGO libraries!
Please also note the descriptions in the manuals for libraries for WAGO-I/O-*PRO* at:
www.wago.com.

8.1 WAGO-I/O-*PRO*-Libraries

The following Table s provide an overview of the WAGO-I/O-*PRO* libraries (CODESYS 2.x) and the device class of MODBUS master devices used for a specific communication channel.

Table 29: CODESYS 2.x libraries used depending on the device class of the MODBUS master devices.

Device Class	MODBUS RTU	ETHERNET TCP	ETHERNET UDP
16-bit derivatives	Modb_l05.lib (with SerCom.Lib)	MODBUSEthernet_04.lib	MODBUSEthernet_04.lib
32-bit derivatives	Modb_l05.lib Module: MODBUS_EXTENDED_MASTER	WagoLibMODBUS_IP_01.lib Module: ETHERNET_MODBUSMASTER_TCP	WagoLibMODBUS_IP_01.lib Module: ETHERNET_MODBUSMASTER_UDP
PERSPECTO® Control Panel/ Control Panel with Target Visualization	Modb_l05.lib Module: MODBUS_EXTENDED_MASTER (with SerComPerspecto.Lib and CRC_Extern.lib)	WagoLibMODBUS_IP_01.lib Module: ETHERNET_MODBUSMASTER_TCP	WagoLibMODBUS_IP_01.lib Module: ETHERNET_MODBUSMASTER_UDP

The following Table provides an overview of the libraries integrated in your WAGO-I/O-*PRO* project (CODESYS 2.x) during code generation. The libraries are only statically linked when required.

Table 30: CODESYS 2.x used depending on the code generation

Library	Description
Modb_l05.lib	Makes the MODBUS RTU function available, and is available for each device class.
WagoLibMODBUS_IP_01.lib	Makes the MODBUS function available for TCP/IP, used for 32-bit and *PERSPECTO*® device classes.
MODBUSEthernet_04.lib	Makes the MODBUS function available for TCP/IP, used for 16-bit device class.
Ethernet.lib	TCP/IP and UDP/IP functionality for 16-bit device class.
Serial_Interface_01.lib	Encapsulates the functions of the SerComm.Lib for easier access.
SerComm.lib	Makes basic functionality for serial communication transparent via internal data bus or device connection.
SercomPerspecto.lib	Adaptation of SerComm.Lib for *PERSPECTO*® device class
mod_com.lib	Makes help functions available, required for MODBUS communication.
CRC_Extern.lib	
MbMasterCommon.lib	Contains specific function blocks that define MODBUS communication to one specific slave or via one specific serial interface. Also included are user-defined data types generated in response to user input.

8.1.1 WAGO-I/O-*PRO* Function Blocks

After making the communication settings and generating function blocks, the generated function blocks are available with MODBUS communication and variable definitions in your WAGO-I/O-*PRO* project for a PLC application.

8.1.2 Directory Structure

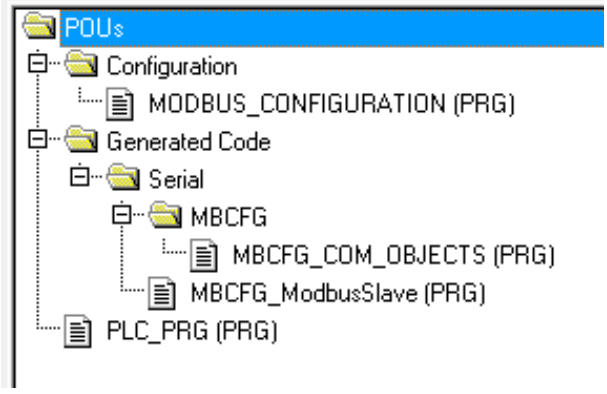


Figure 23: Directory structure of the function blocks (overview)

In the tree structure view, the main program entry "**PLC_PRG [PRG]**" always appears automatically when creating a project in WAGO-IO-*PRO*.
The PLC_PRG [PRG] is a module of the "Program" type and is automatically generated by WAGO-I/O-*PRO* when creating a new project.
This main program module must be contained in each PLC program for project execution if there is no task configuration for the execution sequence. This module may not be deleted or renamed.
PLC_PRG [PRG] is called up once per control cycle.

The settings and function blocks generated in the MODBUS Master Configurator are in the "**Configuration**" and "**Generated Code**" directories.

In the "**Generated Code**" directory, a program module ([PRG] module) is generated for each configured MODBUS slave.
The generated "PRG" modules are stand-alone programs that allow data exchange between the PLC main program and the corresponding MODBUS slave. These are cyclically executed by a task.

Within the task "MBCFG_MASTER_TASK", the entire list of MODBUS slave PRG modules is processed sequentially.
The "MBCFG_MASTER_TASK" is used for all MODBUS communication. Managed by the MODBUS Master Configurator, this task is automatically included as needed.

Note

Manually remove the "MBCFG_MASTER_TASK"!
Please note that automatic removal of the MODBUS Master "MBCFG_MASTER_TASK" is not possible for MODBUS communication. If you no longer want MODBUS Master functionality in your WAGO-I/O-*PRO* project, the task must be manually removed!

There is a list of communication jobs and corresponding variable addresses in the MODBUS process image of the slave within the MODBUS slave program modules.

The variables of the MODBUS slaves defined by you are executed as INPUT or OUTPUT objects of the modules.

8.1.3 Program Execution of the Generated Code

When the generated code is executed, each slave program generates a list of variables to be communicated, corresponding communication jobs and one communication object (TCP/UDP/RTU) at startup.

There can be fewer communication jobs than variables and several variables can be processed with one communication job.
Each variable contains a reference to a communication job.
The two lists are passed to the communication object.
During the task run-time, the communication object runs through the job list, performs the tasks specified (MODBUS read, MODBUS write) and then updates the values of the variables in the list passed.

Before a "Write" job is executed, the variables assigned to the job are first processed and compiled from the data collected in the data range.

For a "Read" job, the process is similar, i.e., the values are first read from the MODBUS slave and then distributed to the assigned variables.

8.1.4 Cycle Time of the "MBCFG_MASTER_TASK"

The task cycle time does not depend on the number of communication jobs, but on the number of variables across all slave PRG objects.
During each cycle, one new job per slave is started or stopped — most of the time is needed to process the variable list (distribute data read, collect data to be written).
For this reason, the cycle time of the task per 100 variables is increased by 1 ms.
Initially, the cycle is set to 10 ms.

8.1.5 Data Storage, Persistent

The communication settings and data points you defined must be cached to be available the next time the configuration dialog is called up.

To save these settings along with the WAGO-I/O-*PRO* project, a "**MODBUS_CONFIGURATION [PRG]**" module is generated that is not assigned to any task.
It is located in the "**Configuration**" directory.

In a comment field for this module, the entire configuration is stored as text with information about the node structure on the internal data bus of the WAGO slaves, the communication settings to the slaves, as well as the designations and data point types.

To load it, WAGO-I/O-*PRO* is first instructed to export the module to a temporary file. The file generated is compiled by the MODBUS Master Configurator (source code to target code) and then displayed as XML in WAGO-I/O-*PRO*.

Once defined, this makes it possible to apply the structure in other WAGO-I/O-*PRO* projects by EXP export of the PRG module.

If there are no configuration settings, an empty configuration is created.

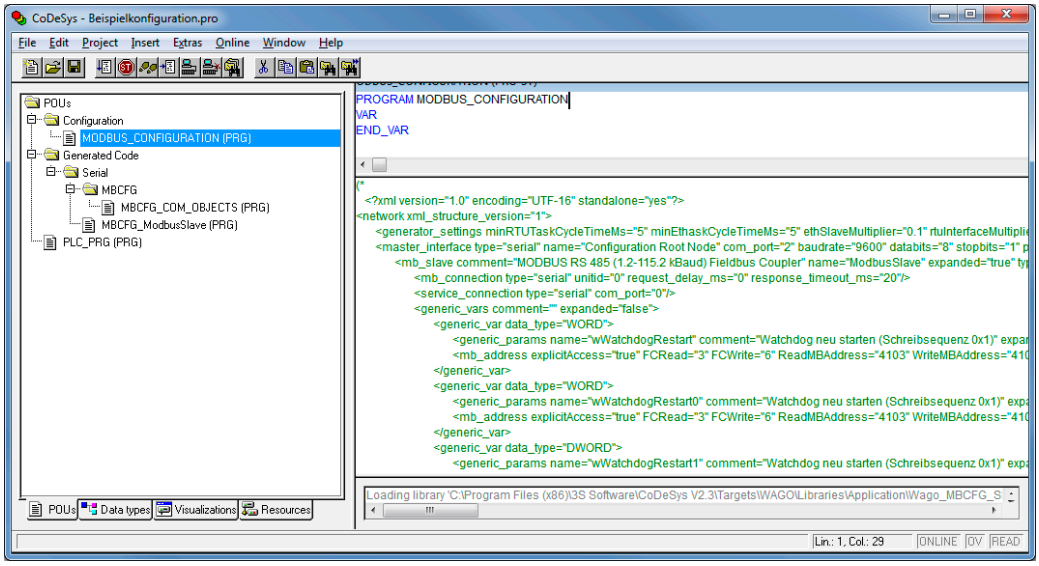


Figure 24: Configuration settings, persistent

Note

Do not manually change the saved format!
Please note that you should not manually change the saved format; the structure is already saved, and any manual change will be detected.

8.1.6 MODBUS Function Codes

For access to WAGO MODBUS slaves, no manual input of the MODBUS communication parameters is required. Which MODBUS services are used can be specified via the MODBUS Master Configurator depending on what data ranges (bit/register area) should be accessed. The definitions are included when generating the code.

Table 31: MODBUS function codes used

Data Range	MODBUS function code (FC)
Bit range	FC01 (Read Coils) FC02 (Read Discrete Inputs) FC05 (Write Single Coil) FC15 (Write Multiple Coils)
Register range	FC03 (Read Holding Registers) FC04 (Read Input Registers) FC06 (Write Single Registers) FC16 (Write Multiple Registers) FC22 (Mask Write Registers) FC23 (Read/Write Multiple Registers)

8.1.7 Access to WAGO Slaves

When accessing MODBUS slaves from WAGO, the following rules apply:

- All register- and bit-based services access the same logical range.
- The data points can be read and written under the same MODBUS address/bit offset.
- Bit-based services do not have access to complex or analog I/O modules.
- The Special Feature Register is not implemented as flat-addressable memory, i.e., neighboring addresses must have no logical connection.

8.1.8 Access to the Hardware Area

For access to the input and output data of the WAGO MODBUS slaves (I/O modules), use either the register services or the bit services for digital data.

8.1.9 Access to the PFC Variable Area and Flags

The register services access the PFC area of the WAGO MODBUS slaves with CODESYS 2.x runtime system.

8.1.10 Access to the Special Feature Registers

Access to the Special Feature Registers is realized by means of the register services.

8.1.11 Optimizing Communication with WAGO MODBUS Slaves

The number of communication processes when reading and writing can be reduced when several data points can be combined into one read/write operation. This optimization cannot be applied for Special Feature Registers as there is no flat address on the MODBUS level, i.e., neighboring addresses must have no logical connection.

8.1.12 Access to Generic MODBUS Slaves, Manufacturer-Independent

If MODBUS slaves from other manufacturers are used, then the data points must be manually defined with corresponding read/write commands and MODBUS addresses/bit offsets.

When accessing generic MODBUS slaves from other manufacturers, the following rules apply:

- Data points cannot be read and written under the same MODBUS address/bit offset.
- The length of the data points when reading and writing may not differ.
- When accessing data points with lengths not from $M=\{x*16 \text{ bit} \mid x \in N\}$, it is accepted that remaining areas of the accessed address are overwritten with 0.
- Optimizing communication is optional since it cannot be assumed that corresponding FC codes (FC16, FC15) are implemented in the target system, and that there is a flat-addressable area. As many MODBUS slave types should be supported as possible.

8.1.13 Access to MODBUS Data Points from the IEC 61131 Program

Because a separate program module is created for each MODBUS slave when generated, access to the individual data points of the MODBUS slaves is provided by the input or output variables of the respective slave program.

List of Figures

Figure 1: View of the controller configuration with MODBUS Master Configurator 14
Figure 2: Example configuration for using the MODBUS Master Configurator.. 15
Figure 3: Example of "optimized" and "explicit" data access 16
Figure 4: User interface of the MODBUS Master Configurator 20
Figure 5: **MODBUS netzwork** dialog (without configuration) 21
Figure 6: **MODBUS netzwork** dialog (with example configuration) 21
Figure 7: Context menu for the""**MODBUS master"** entry 22
Figure 8: **Parameter code generator** dialog 23
Figure 9: Context menu for the "**Interface (COMx)"** entry 24
Figure 10: **Settings: Serial interface** dialog 25
Figure 11: **Search device** dialog 26
Figure 12: Context menu for the "**Interface (ETHERNET)"** entry 27
Figure 13: **Search device** dialog 29
Figure 14: Context menu for the "**ModbusSlave**" entry 30
Figure 15: **Communication settings** dialog [for Interface (COMx)] 31
Figure 16: **Communication settings** dialog [for Interface (ETHERNET)] 33
Figure 17: **MODBUS slave selection** Dialog 35
Figure 18: Context menu for the "**Generic variables**" entry 35
Figure 19: **WAGO MODBUS register** dialog 36
Figure 20: Context menu for the "**PFC variables**" entry 37
Figure 21: Context menu for the "**K-Bus**" entry 38
Figure 22: **Module selection** dialog 39
Figure 23: Directory structure of the function blocks (overview) 52
Figure 24: Configuration settings, persistent 54

List of Tables

Table 1: Number notation 8
Table 2: Font conventions 8
Table 3: Required PC hardware 9
Table 4: Required software 10
Table 5: Optional software 10
Table 6: Supported MODBUS master devices 11
Table 7: Hardware interfaces 11
Table 8: Possible WAGO MODBUS slaves 12
Table 9: Configuration procedure overview 17
Table 10: General settings from the context menus of tree structure entries 18
Table 11: Legend to figure "User interface of the MODBUS Master Configurator" 20
Table 12: Legend to figure "**MODBUS netzwork** dialog" 22
Table 13: Legende to Figure "Context menu for the "**MODBUS master** entry" . 22
Table 14: Legende to Figure "**Parameter code generator** dialog" 24
Table 15: Legend to Figure "Context menu for the **Interface (COMx)** entry 24
Table 16: Legend to Figure "**Settings: Serial interface** dialog" 25
Table 17: Legend to Figure " **Search device** dialog" 27
Table 18: Legend to Figure "Context menu for the **Interface (ETHERNET)** entry" 28
Table 19: Legend to Figure "**Search device** dialog" 30
Table 20: Legend to Figure "Context menu for the **ModbusSlave** entry" 31
Table 21: Legende to Figure "**Communication settings** dialog [for Interface (COMx)]**)**" 32
Table 22: Legend to Figure " **Communication settings** dialog [for Interface (ETHERNET)]" 33
Table 23: Legend to Figure " **MODBUS slave selection** Dialog" 35
Table 24: Legend to Figure " Context menu for the **Generic variables** entry " .. 35
Table 25: Legend to Figure "**WAGO MODBUS register**" dialog 36
Table 26: Legend to Figure " Context menu for the "**PFC variables**" entry" 37
Table 27: Legend to Figure "Context menu for the **K-Bus** entry" 38
Table 28: Legend to Figure " **Module selection** dialog**"** 39
Table 29: CODESYS 2.x libraries used depending on the device class of the MODBUS master devices. 51
Table 30: CODESYS 2.x used depending on the code generation 52
Table 31: MODBUS function codes used 55

WAGO Kontakttechnik GmbH & Co. KG
Postfach 2880 • D-32385 Minden
Hansastraße 27 • D-32423 Minden
Phone: +49/5 71/8 87 – 0
Fax: +49/5 71/8 87 – 1 69
E-Mail: info@wago.com

Internet: http://www.wago.com