Operating Instruction Manual

SyCon

System Configurator for Ethernet and Serial Protocols

Ethernet (TCP, UDP, ALI, SMTP, FTP), Open Modbus/TCP
ASCII, 3964R, RK512, Modbus RTU, Modbus Plus, Modnet 1/N, Modnet 1/SFB

Edition: 1
Language: English (EN)

Hilscher Gesellschaft für Systemautomation mbH

Web: www.hilscher.com

Contact

Hilscher Europe

Contact	Germany	France
Address	Hilscher Gesellschaft für Systemautomation mbH Rheinstrasse 15 D-65795 Hattersheim	Hilscher France s.a.r.l. 12 rue du 35ème Régiment d'Aviation Miniparc du Chêne F-69500 Bron
Phone	+49 (0) 6190 / 9907-0	+33 (0) 472379840
Phone Sales	+49 (0) 6190 / 9907-90	+33 (0) 472379840
Phone Support	+49 (0) 6190 / 9907-99	+33 (0) 472379840
Fax	+49 (0) 6190 / 9907-50	+33 (0) 478268327
Email	info@hilscher.com	
Email Sales	sales@hilscher.com	
Email Support	hotline@hilscher.com	
Homepage	www.hilscher.com	www.hilscher.com

Contact	Italy	Switzerland
Address	Hilscher Italia s.r.l. Via Grandi, 25 I-20090 Vimodrone (MI)	Hilscher Swiss GmbH Hubelmattstrasse 29 CH-4500 Solothurn
Phone	+39 / 0225007068	+41 (0) 326236633
Phone Sales	+39 / 0225007068	+41 (0) 326236633
Phone Support	+39 / 0225007068	+49 (0) 6190 / 9907-99
Fax	+39 / 0225029973	+41 (0) 326236632
Email	info@hilscher.it	info@hilscher.com
Email Sales	sales@hilscher.it	sales@hilscher.com
Email Support	supporto@hilscher.it	hotline@hilscher.com
Homepage	www.hilscher.com	www.hilscher.com

Hilscher North America

Contact	North America
Address	Hilscher North America, Inc. 4320 Winfield Road, Suite 200 Warrenville, Illinois 60555, USA
Phone	+1 / 630.836.8677
Phone Sales	+1 / 630.836.8677
Phone Support	+1 / 630.836.8677
Fax	+1 / 630.836.8010
Email	info@hilscher.us
Email Sales	info@hilscher.us
Email Support	info@hilscher.us
Homepage	www.hilscher.com

Worldwide: Distributors

Please visit our Homepage at
www.hilscher.com

List of Revisions

Index	Date	Version	Chapter	Revision
1	18.06.04	Protocol.dll 2.704 SyCon.exe 2.653	all	created - Ethernet (TCP, UDP, SMTP, FTP), - Open Modbus/TCP - ASCII, - 3964R, - RK512, - Modbus RTU, - Modbus Plus, - Modnet 1/N, - Modnet 1/SFB

Table of Contents

1 Overview SyCon

1.1 Main Functions

The main functions of the Ethernet/Protocol System Configurator are:

Function	Section	Short Description
Configuration	*Overview Communication Types*	Overview communication types and description of the configuration steps
Diagnostic	*Diagnostic Functions*	Extended Device Diagnostic, Message Monitor,
Documentation	*Project Information*	Set the project information
	Print	Print out the configuration

Table 1: SyCon Main Functions

1.2 Properties

SyCon is a universal Fieldbus Configurator

This means you can configure the most important fieldbus systems like PROFIBUS, InterBus, CANopen, DeviceNet, ControlNet, SDS, AS-Interface etc. with the same tool.

SyCon is a global Fieldbus Configurator

You configure all devices with one tool. SyCon checks the dependencies between the devices. SyCon only allows configurations that make sense. In case of doubt SyCon will give you a warning.

To Hilscher devices you can make downloads of the configuration data. For other devices, export functions or documentation possibilities are available.

SyCon documents your Fieldbus system

After the configuration you can print out a detailed documentation of your fieldbus network. The details can be switched on/off. You can print documentation with details between the bus topology and the detail of one device.

SyCon uses standardized configuration files

Some protocols support standardized files containing information about all features and limitations of the Slave device. SyCon uses these files for the configuration.

SyCon is a diagnostic tool

After the configuration you can switch SyCon into the diagnostic mode. You can watch all status information of Hilscher devices, see protocol dependent diagnostic information. In this case a Slave is not operating correctly will be displayed in a different colour.

SyCon can be extended

SyCon consists of a universal EXE file and several protocol specific DLLs. Most customers demand SyCon only for one bus system.

SyCon can be enlarged later by adding one or more DLLs for any other available protocol. The configuration of the different protocols will be as similar as possible.

2 Installation and Licensing

2.1 System Requirements

- PC with 486-, Pentium processor or higher
- Windows 95/98/ME, Windows NT/2000/XP
- Free disk space: 30 - 80 MByte
- CD ROM drive
- RAM: min. 16 MByte
- Graphic resolution: min. 800 x 600 pixel
- Windows 95: Service Pack 1
- Windows NT: Service Pack 6
- DCOM only for OPC Server
- Keyboard and Mouse

2.2 Software Installation

Close all application programs on the system!

Insert the CD in the local CD ROM drive. The installation program will start by itself (Autostart enabled). Otherwise change into the root directory on the CD and start Autorun.exe (Autostart disabled).

Note: Administrator privileges are required on Windows NT/2000/XP systems for installation!

The installation program asks for the components you want to install. Answer these questions with **Yes** or **No**.

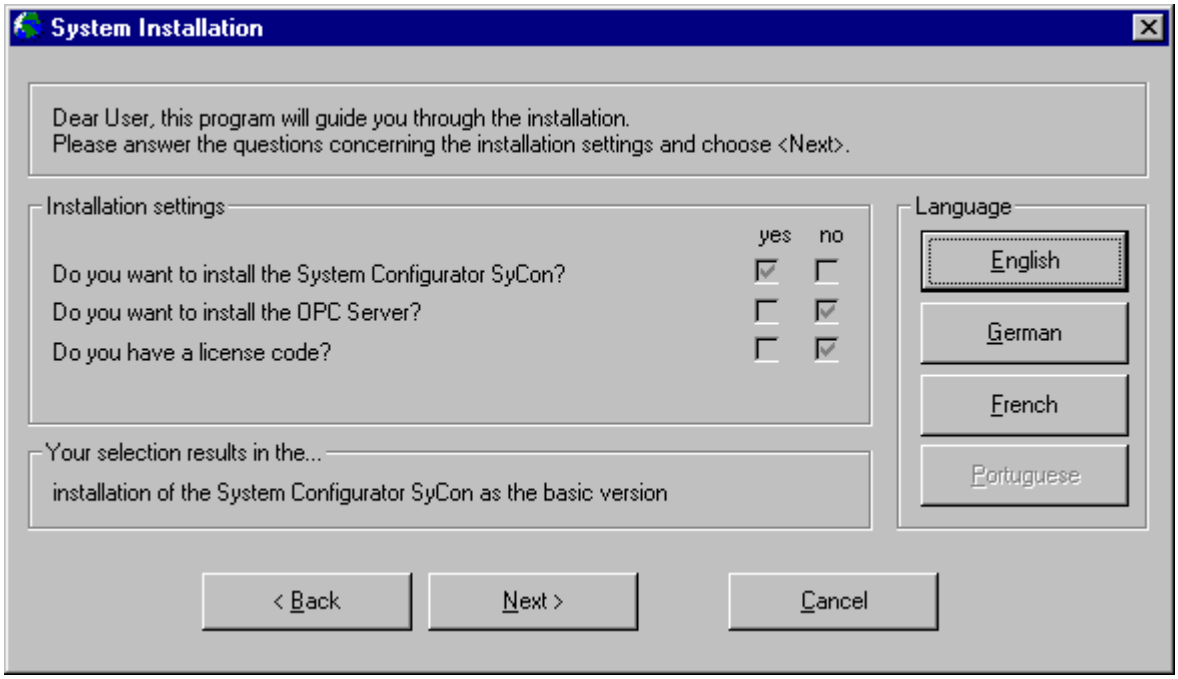

Figure 1: Selection for the Installation of the System Configurator in basic version (sufficient for Ethernet/Protocol)

Note: In the case of the installation of the System Configurator SyCon for Ethernet/Protocol no license code is necessary.

It can be installed:

- System Configurator SyCon (Configuration and diagnostic tool)
- OPC-Server (For OPC communication)
- CIF Device Driver (Device Driver for access to the CIF)

Note: The OPC-Server is designed to be used with PROFIBUS, InterBus, CANopen, DeviceNet, ControlNet and AS-interface. The OPC-Server is not designed to be used for Ethernet and is not designed to be used for serial protocols.

2.3 Scope of functions of the basic version and unlicensed Fieldbus Modules

For the System Configurator SyCon no license is required because the basic version includes all functions for operating the Hilscher devices. This is valid for the protocols

- Ethernet (TCP, UDP, SMTP, FTP),
- Open Modbus/TCP
- ASCII
- 3964R
- RK512
- Modbus RTU
- Modbus Plus
- Modnet 1/N
- Modnet 1/SFB

3 Getting Started - Configuration Steps

3.1 Overview Communication Types

Select the protocol that you want to use from the following table for Ethernet or from the table for serial protocols. The configuration steps are described in the given section.

Note: The booklet with the CD ROM contains information for the hardware installation and information to the cable. At this point it is presupposed that the hardware installation was done.

3.1.1 Ethernet Protocols

Communication	Described in section	Page
Ethernet (TCP, UDP, SMTP, FTP)	*Configuration Ethernet (TCP, UDP, ALI, SMTP, FTP)*	*16*
Open Modbus / TCP	*Configuration Open Modbus / TCP Protocol*	*16*

Table 2: Overview Communication Types Ethernet Protocols

3.1.2 Serial Protocols

Communication	Described in section	Page
ASCII	*Configuration ASCII Protocol*	*17*
3946R	*Configuration 3946R Protocol*	*18*
RK512	*Configuration RK512 Protocol*	*18*
Modbus RTU	*Configuration Modbus RTU Protocol*	*19*
Modbus Plus	*Configuration Modbus Plus Protocol*	*19*
Modnet 1/N	*Configuration Modnet 1/N Protocol*	*20*
Modnet 1/SFB	*Configuration Modnet 1/SFB Protocol*	*20*

Table 3: Overview Communication Types serial Protocols

3.2 Configuration Ethernet

3.2.1 Configuration Ethernet (TCP, UDP, ALI, SMTP, FTP)

The following table describes the steps to configure a Hilscher Ethernet (TCP, UDP, ALI, SMTP, FTP) device as it is typical for many cases.

#	Action	Menu in the System Configurator	Detail information in section	Page
1	Create a new project	**File > New > Ethernet/Protocol**	*Setting up the Ethernet/Serial Protocol Configuration*	*21*
2	Choose Hilscher Ethernet device	**Insert > Device**	*Insert Device*	*22*
3	Set Device Assignment	**Settings > Device Assignment**	*Device Assignment*	*27*
4	Set Parameter	**Settings > Ethernet Parameter**	*Setting of the Ethernet Parameter*	*38*
5	Set Device Settings	**Settings > Device Settings**	*Device Settings*	*65*
6	Save project	**File > Save**	*Save and Save As*	*85*
7	Download	**Online > Download**	*Downloading the Configuration*	*71*
8	Transfer user data	**Online > Message Monitor**	*Message Monitor*	*82*

Table 4: Steps for Configuration of an Ethernet Device

3.2.2 Configuration Open Modbus / TCP Protocol

The following table describes the steps to configure a Hilscher Ethernet device with the Open Modbus / TCP protocol as it is typical for many cases.

#	Action	Menu in the System Configurator	Detail information in section	Page
1	Create a new project	**File > New > Ethernet/Protocol**	*Setting up the Ethernet/Serial Protocol Configuration*	*21*
2	Choose Hilscher Open Modbus/TCP device	**Insert > Device**	*Insert Device*	*22*
3	Set Device Assignment	**Settings > Device Assignment**	*Device Assignment*	*27*
4	Set Parameter	**Settings > Open Modbus / TCP Parameter**	*Setting of the Open Modbus / TCP Parameter*	*44*
5	Set Device Settings	**Settings > Device Settings**	*Device Settings*	*65*
6	Save project	**File > Save**	*Save and Save As*	*85*
7	Download	**Online > Download**	*Downloading the Configuration*	*71*
8	Transfer user data	**Online > Message Monitor**	*Message Monitor*	*82*

Table 5: Steps for Configuration of the Open Modbus / TCP Protocol

3.3 Serial Protocols

3.3.1 Configuration ASCII Protocol

The following table describes the steps to configure a Hilscher device with the ASCII protocol as it is typical for many cases.

#	Action	Menu in the System Configurator	Detail information in section	Page
1	Create a new project	**File > New > Ethernet/Protocol**	*Setting up the Ethernet/Serial Protocol Configuration*	*21*
2	Choose Hilscher ASCII device	**Insert > Device**	*Insert Device*	*22*
3	Set Device Assignment	**Settings > Device Assignment**	*Device Assignment*	*27*
4	Set Parameter	**Settings > ASCII Parameter**	*Setting of the ASCII Parameter*	*46*
5	Set Device Settings	**Settings > Device Settings**	*Device Settings*	*65*
6	Save project	**File > Save**	*Save and Save As*	*85*
7	Download	**Online > Download**	*Downloading the Configuration*	*71*
8	Transfer user data	**Online > Message Monitor**	*Message Monitor*	*82*

Table 6: Steps for Configuration of the ASCII Protocol

3.3.2 Configuration 3946R Protocol

The following table describes the steps to configure a Hilscher device with the 3946R protocol as it is typical for many cases.

#	Action	Menu in the System Configurator	Detail information in section	Page
1	Create a new project	**File > New > Ethernet/Protocol**	*Setting up the Ethernet/Serial Protocol Configuration*	*21*
2	Choose Hilscher 3964R device	**Insert > Device**	*Insert Device*	*22*
3	Set Device Assignment	**Settings > Device Assignment**	*Device Assignment*	*27*
4	Set Parameter	**Settings > 3946R Parameter**	*Setting of the 3964R Parameter*	*51*
5	Set Device Settings	**Settings > Device Settings**	*Device Settings*	*65*
6	Save project	**File > Save**	*Save and Save As*	*85*
7	Download	**Online > Download**	*Downloading the Configuration*	*71*
8	Transfer user data	**Online > Message Monitor**	*Message Monitor*	*82*

Table 7: Steps for Configuration of the 3946R Protocol

3.3.3 Configuration RK512 Protocol

The following table describes the steps to configure a Hilscher device with the RK512 protocol as it is typical for many cases.

#	Action	Menu in the System Configurator	Detail information in section	Page
1	Create a new project	**File > New > Ethernet/Protocol**	*Setting up the Ethernet/Serial Protocol Configuration*	*21*
2	Choose Hilscher RK512 device	**Insert > Device**	*Insert Device*	*22*
3	Set Device Assignment	**Settings > Device Assignment**	*Device Assignment*	*27*
4	Set Parameter	**Settings > RK512 Parameter**	*Setting of the RK512 Parameter*	*53*
5	Set Device Settings	**Settings > Device Settings**	*Device Settings*	*65*
6	Save project	**File > Save**	*Save and Save As*	*85*
7	Download	**Online > Download**	*Downloading the Configuration*	*71*
8	Transfer user data	**Online > Message Monitor**	*Message Monitor*	*82*

Table 8: Steps for Configuration of the RK512 Protocol

3.3.4 Configuration Modbus RTU Protocol

The following table describes the steps to configure a Hilscher device with the Modbus RTU protocol as it is typical for many cases.

#	Action	Menu in the System Configurator	Detail information in section	Page
1	Create a new project	**File > New > Ethernet/Protocol**	*Setting up the Ethernet/Serial Protocol Configuration*	*21*
2	Choose Hilscher Modbus RTU device	**Insert > Device**	*Insert Device*	*22*
3	Set Device Assignment	**Settings > Device Assignment**	*Device Assignment*	*27*
4	Set Parameter	**Settings > Modbus RTU Parameter**	*Setting of the Modbus RTU Parameter*	*55*
5	Set Device Settings	**Settings > Device Settings**	*Device Settings*	*65*
6	Save project	**File > Save**	*Save and Save As*	*85*
7	Download	**Online > Download**	*Downloading the Configuration*	*71*
8	Transfer user data	**Online > Message Monitor**	*Message Monitor*	*82*

Table 9: Steps for Configuration of the Modbus RTU Protocol

3.3.5 Configuration Modbus Plus Protocol

The following table describes the steps to configure a Hilscher device with the Modbus Plus as it is typical for many cases.

#	Action	Menu in the System Configurator	Detail information in section	Page
1	Create a new project	**File > New > Ethernet/Protocol**	*Setting up the Ethernet/Serial Protocol Configuration*	*21*
2	Choose Hilscher Modbus Plus device	**Insert > Device**	*Insert Device*	*22*
3	Set Device Assignment	**Settings > Device Assignment**	*Device Assignment*	*27*
4	Set Parameter	**Settings > Modbus Plus Parameter**	*Setting of the Modbus Plus Parameter*	*57*
5	Set Device Settings	**Settings > Device Settings**	*Device Settings*	*65*
6	Save project	**File > Save**	*Save and Save As*	*85*
7	Download	**Online > Download**	*Downloading the Configuration*	*71*
8	Transfer user data	**Online > Message Monitor**	*Message Monitor*	*82*

Table 10: Steps for Configuration of the Modbus Plus Protocol

3.3.6 Configuration Modnet 1/N Protocol

The following table describes the steps to configure a Hilscher device with the Modnet 1/N protocol as it is typical for many cases.

#	Action	Menu in the System Configurator	Detail information in section	Page
1	Create a new project	**File > New > Ethernet/Protocol**	*Setting up the Ethernet/Serial Protocol Configuration*	*21*
2	Choose Hilscher Modnet 1/N device	**Insert > Device**	*Insert Device*	*22*
3	Set Device Assignment	**Settings > Device Assignment**	*Device Assignment*	*27*
4	Set Parameter	**Settings > Modnet 1/N Parameter**	*Setting of the Modnet 1/N Parameter*	*61*
5	Set Device Settings	**Settings > Device Settings**	*Device Settings*	*65*
6	Save project	**File > Save**	*Save and Save As*	*85*
7	Download	**Online > Download**	*Downloading the Configuration*	*71*
8	Transfer user data	**Online > Message Monitor**	*Message Monitor*	*82*

Table 11: Steps for Configuration of the Modnet 1/N Protocol

3.3.7 Configuration Modnet 1/SFB Protocol

The following table describes the steps to configure a Hilscher device with the Modnet 1/SFB protocol as it is typical for many cases.

#	Action	Menu in the System Configurator	Detail information in section	Page
1	Create a new project	**File > New > Ethernet/Protocol**	*Setting up the Ethernet/Serial Protocol Configuration*	*21*
2	Choose Hilscher Modnet 1/SFB device	**Insert > Device**	*Insert Device*	*22*
3	Set Device Assignment	**Settings > Device Assignment**	*Device Assignment*	*27*
4	Set Parameter	**Settings > Modnet 1/SFB Parameter**	*Setting of the Modnet 1/SFB Parameter*	*63*
5	Set Device Settings	**Settings > Device Settings**	*Device Settings*	*65*
6	Save project	**File > Save**	*Save and Save As*	*85*
7	Download	**Online > Download**	*Downloading the Configuration*	*71*
8	Transfer user data	**Online > Message Monitor**	*Message Monitor*	*82*

Table 12: Steps for Configuration of the Modnet 1/SFB Protocol

4 Configuration of Ethernet/Serial Protocol with SyCon

4.1 Setting up the Ethernet/Serial Protocol Configuration

To create a new configuration, choose the **File > New** menu. This will offer a selection list of fieldbus systems and Ethernet/Protocol. Choose **Ethernet/Protocol**. If only the Ethernet/Protocol software component is installed, the configuration window will open directly.

The name of the configuration file can be assigned when the configuration is finished or with **File > Save As**.

4.2 Devices

4.2.1 Insert Device

In order to insert a (Hilscher) Device into the configuration, choose the **Insert > Device** menu or click on the symbol:

Symbol Insert > Device


Figure 2: Symbol Insert > Device

The following devices can be selected:

4.2.1.1 List of Device Types for Ethernet

Device	Ethernet	Open Modbus / TCP
CIF 50-EN	X	X
CIF 104-EN	X	X
COM-EN	X	X
COM-C-EN	X	-
EC1-DEB-EN	X	-
PKV40-EN/RFC	X	-

Table 13: List of Device Types for Ethernet

Meaning: **X** means available. **-** means not available or not possible.

The dialog box opens, from which exactly one device can be chosen in the list **Available devices**. By clicking at the **Add** the chosen device appears in the right dialog **Selected Devices**.

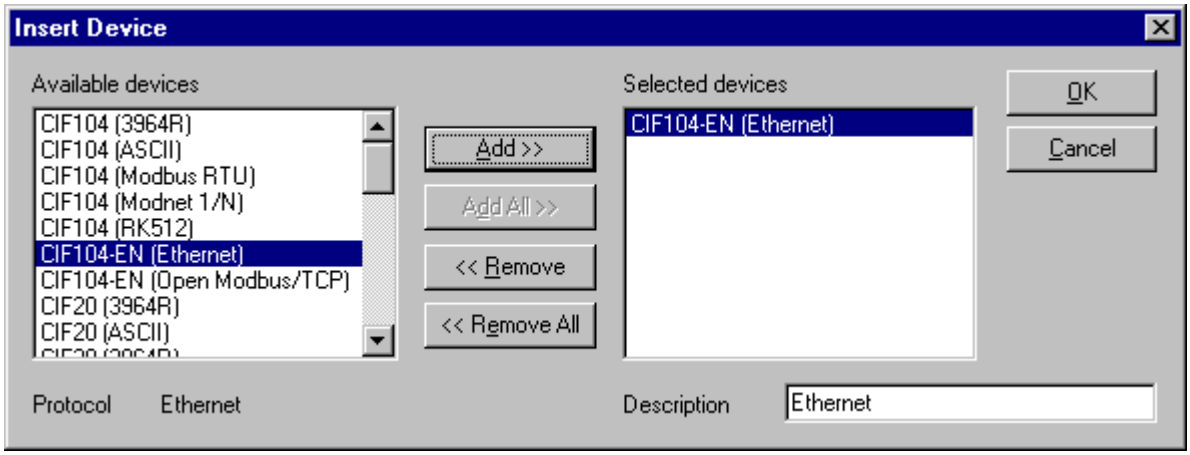

Figure 3: Insert > Device (Ethernet)

This example shows a CIF 104-EN. It gets the standard description **Ethernet**. This description is changeable in the field.

4.2.1.2 List of Device Types for Serial Protocols

Device	ASCII	3946R	RK512	Modbus RTU	Modbus Plus	Modnet 1/N	Modnet 1/SFB
CIF 104	X	X	X	X	-	X	-
CIF 104-MBP	-	-	-	-	X	-	-
CIF 20	X	X	-	-	-	-	-
CIF 30	X	X	X	X	-	X	-
CIF 30-MBP	-	-	-	-	X	-	-
CIF 50	X	X	X	X	-	X	-
CIF 50-MBP	-	-	-	-	X	-	-
CIF 50-SFB	-	-	-	-	-	-	X
COM 10/11	(X)	(X)	(X)	X	-	-	-
PKV 40-MBP	-	-	-	-	X	-	-
PKV 50-MBP	-	-	-	-	X	-	-

Table 14: Possible Device Types for Serial Protocols

Meaning: **X** means available. **(X)** means device available, no firmware available. **-** means not available or not possible.

The dialog box opens, from which exactly one device can be chosen in the list **Available devices**. By clicking at the **Add** the chosen device appears in the right dialog **Selected Devices**.

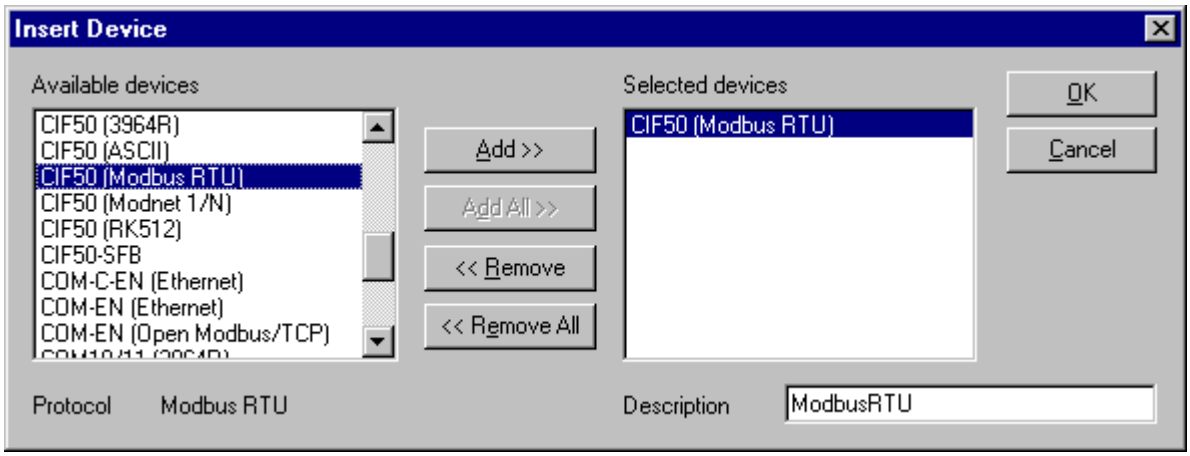

Figure 4: Insert > Device (Serial Protocol)

This example shows a CIF 50 for Modbus RTU. It gets the standard description **ModbusRTU**. This description is changeable in the field.

4.2.1.3 Hardware Assignment

If you have configured the CIF Device Driver Setup for your hardware and you insert the correct device in the SyCon it detects this hardware and displays at which board and which driver was detected and ask if the hardware should be assigned.

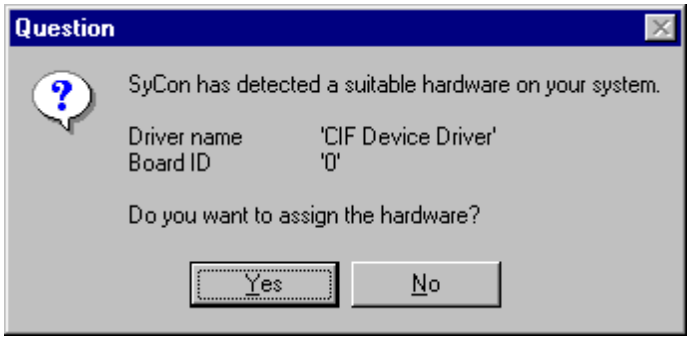

Figure 5: Assign hardware

If you answer with **Yes** the hardware is assigned. If you answer with **No** you have to assign this hardware with **Settings > Device Assignment** (look in section *Device Assignment* at page *27*).

4.2.2 Device Settings

See section *Device Settings* at page *65*.

4.2.3 Replace Device

If a device already exists in the configuration and should be replaced against another device select the menu **Insert > Device** or click on the "Insert device" symbol. A security question appears, if the device should be replaced.

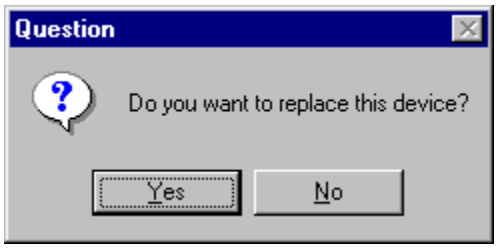

Figure 6: Security Question Replace Device

If you click the **Yes** button a new window opens, where you can replace the existing device against another one.

In this window you select the device from the list **Available devices** by clicking on it and by clicking the **Add** button this device is shown in the first position in the list **Selected devices**. With **OK** you confirm the selection and the device will be replaced.

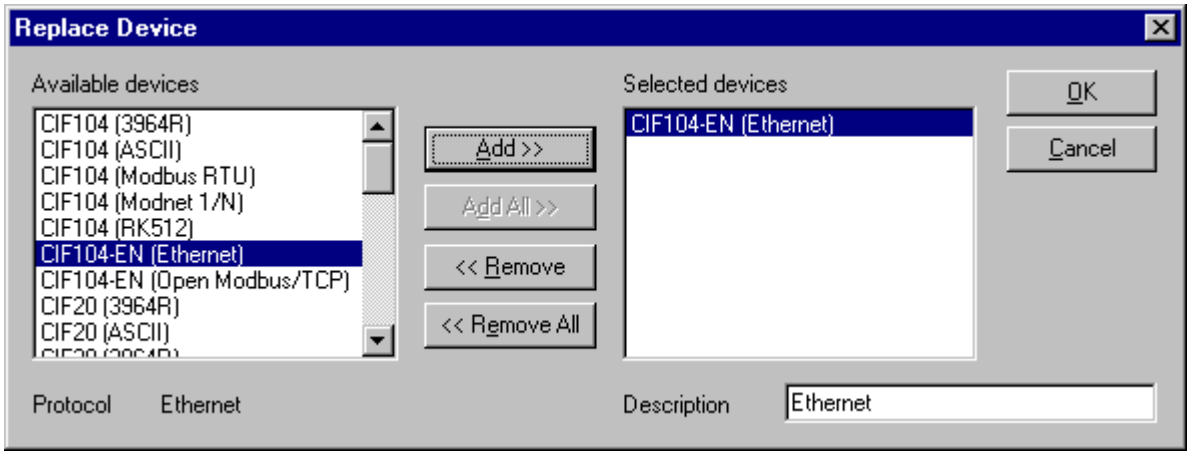

Figure 7: Replace Device

5 Settings

5.1 Device Assignment

The Device Assignment setting determines how the System Configurator communicates with the device. This is selected in the device arrangement via the menu **Settings > Device Assignment**.

5.1.1 Driver Selection

Calling up the **Device Assignment**, firstly a dialog window opens, where a driver has to be selected.

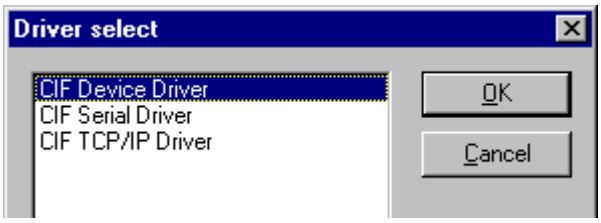

Figure 8: Driver Selection

With the selection of the driver, it is determined, how the System Configurator communicates with the device. The following drivers are available:

- **CIF Device Driver**

  The System Configurator communicates with the Hilscher device via the Dual-port memory of the device.

  This communication is utilized when the System Configurator is used on the same PC on which the Hilscher device is installed.

Note: The CIF Device Driver has to be installed and it must have access to the Hilscher device.

- **CIF Serial Driver**

  The System Configurator communicates with the Hilscher device via a serial connection. In this case a COM port of the PC must be connected with the diagnostic interface of the Hilscher device via a diagnostic cable.

Note: The pin assignment of the diagnostic cable is described in the hardware documentation of the device manufacturer.

This communication is utilized when the System Configurator has to access the device via the diagnostic interface of the Hilscher device. The following two application cases are possible:

Application case 1: The System Configurator is installed on another PC (e.g. a notebook) than the Hilscher device.

Application case 2: The System Configurator is installed on the same PC on which the Hilscher device is situated. Then the application can use the Dual-port memory to access the Hilscher device and the diagnostic interface can be used at the same time to communicate with the device (diagnostic data).

- **CIF TCP/IP Driver**

  The System Configurator communicates with the Hilscher device via an Ethernet TCP/IP connection.

  This communication is utilized when the System Configurator is installed on a PC and the PC and the Hilscher device is connected via Ethernet.

  It has to be distinguished:

  1. The Hilscher device is installed in a PC and the TCP/IP connection is built up to the PC, which means the IP address of the PC is used as IP address.

  2. The Hilscher device has an own Ethernet connection and the TCP/IP connection is built up to the Hilscher device, that means the IP address of the Hilscher device is used as IP address.

Select the favored driver for the communication between the System Configurator and the used device from the lower table.

You find a detailed instruction about the selection of the several drivers in the denoted section:

Driver	Described in section	Page
CIF Device Driver	*CIF Device Driver*	*29*
CIF Serial Driver	*CIF Serial Driver*	*31*
CIF TCP/IP Driver	*CIF TCP/IP Driver*	*33*

Table 15: Driver Selection

To select a driver, mark the favored driver by clicking on it in the dialog window **Driver Select** and confirm your selection with **OK**.

The configuration window of the favored driver opens.

5.1.2 CIF Device Driver

The CIF Device Driver supports up to four Hilscher devices in one PC, and they are accessed via the Dual-port memory.

Driver Description

Figure 9: CIF Device Driver - Driver Description

In the upper part of the **CIF Device Driver** dialog the actual used CIF Device Driver and its version number are displayed.

This display is only for information purposes and is not editable by the user.

Board Selection

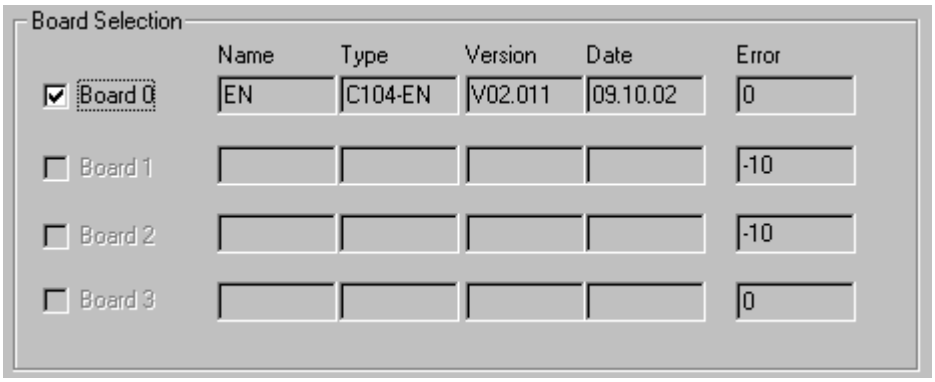

Figure 10: CIF Device Driver - Board Selection

If an assignable device is found by the CIF Device Driver, the checkbox next to the board number is selectable. To select the device, you have to enable it by clicking in the checkbox located left of the desired board and confirm this selection with **OK**.

Checkbox	Description
☐	Device is still not assigned and it can be selected.
☑	Device is assigned. The Assignment can be abrogated by deselecting.
☐	The assignment of the device is not possible.
☑	The device is still assigned in another open configuration and can not be selected here.

Table 16: Device Assignment - Checkboxes of the CIF Device Driver

Now the device is connected with the System Configurator via the CIF Device Driver and the Device Assignment dialog is closed.

By clicking the **Cancel** button, the Device Assignment is closed without an assignment has been completed or respectively changed.

More Details of the CIF Device Driver

Next to the field **Board Selection** there is a button with the name **more>>**. Selecting this button, a dialog opens which displays further information about the CIF Device Driver.

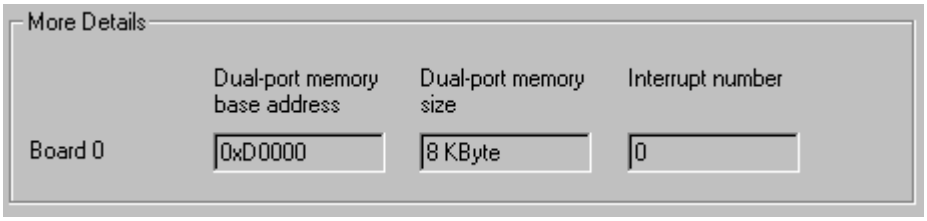

Figure 11: CIF Device Driver - More Details

In this dialog the used **Dual-port memory base address**, the **Dual-port memory size** and the **Interrupt number** of the selected board are displayed. Interrupt number 0 means polling mode.

This display is only for information purposes and is not editable by the user.

5.1.3 CIF Serial Driver

The CIF Serial Driver supports the interfaces COM1 to COM 4 of the PC, in order to get the configuration or to do diagnostic serially via the diagnostic interface of the Hilscher device.

Driver Description

Figure 12: CIF Serial Driver - Driver Description

In the upper part of the **CIF Serial Driver** dialog the actual used driver is displayed.

This display is only for information purposes and is not editable by the user.

Board Selection

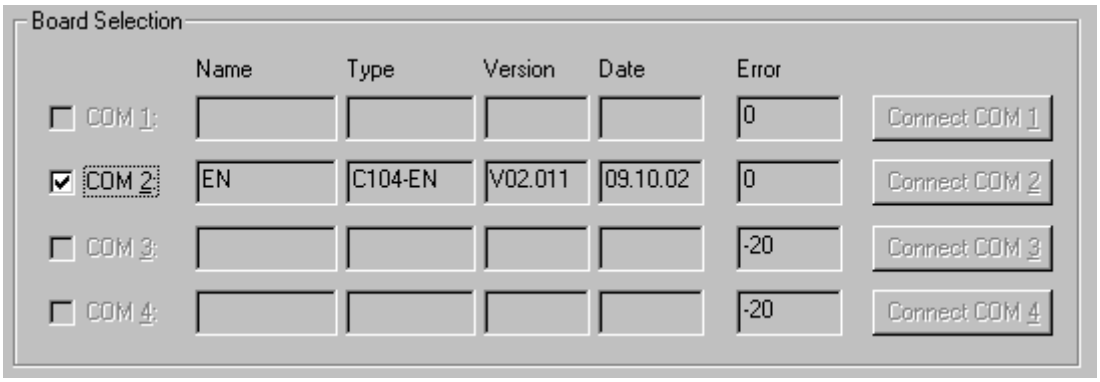

Figure 13: CIF Serial Driver - Board Selection

First the connection must be established by clicking on the button **Connect COM1** or **Connect COM2** or **Connect COM3** or **Connect COM4**.

They can be used depending on which COM interfaces are installed and free on the PC.

The System Configurator sends a request to the corresponding COM Port and polls the Firmware of the device. If the device is connected, the Firmware of the device is displayed and the checkbox of the corresponding COM interface is selectable.

Checkbox	Description
	Device is still not assigned and it can be selected.
	Device is assigned. The Assignment can be abrogated by deselecting.
	The assignment of the device is not possible.
	The device is still assigned in another open configuration and can not be selected here.

Table 17: Device Assignment - Checkboxes of the CIF Serial Driver

This selection has to be confirmed by clicking the **OK** button. Now the device is connected with the System Configurator via the serial driver and the Device Assignment dialog is closed. If the assignment is not possible or if the assignment has failed, this is displayed by an error number in the **Error** column.

If the error number **(-51)** appears after activating one of the buttons, a timeout error has occurred. That means no device is connected to this COM port.

The error number **(-20)** indicates that this COM port is not available or not free (already in use).

By clicking the **Cancel** button, the Device Assignment is closed without an assignment has been completed or respectively changed.

5.1.4 CIF TCP/IP Driver

The CIF TCP/IP Driver builds up a connection to the Hilscher device via Ethernet TCP/IP.

This communication is utilized when the System Configurator is installed on a PC and the PC and the Hilscher device are connected via Ethernet.

It is distinguished between two application possibilities:

1. The Hilscher device is installed in a PC and the TCP/IP connection is built up to the PC, which means the IP address of the PC is used as IP address.

This PC is called Remote PC in the following. The following two requirements have to be accomplished to get access to the Hilscher device via Ethernet TCP/IP:

Note: The CIF Device Driver has to be installed and it must have access to the Hilscher device. Additionally the TCP/IP Server has to be started on the Remote PC.

2. The Hilscher device has an own Ethernet connection and the TCP/IP connection is built up to the Hilscher device, that means the IP address of the Hilscher device is used as IP address.

Driver Description

Figure 14: CIF TCP/IP Driver - Driver Description

In the upper part of the **CIF TCP/IP Driver** dialog the actual used driver and its version number are displayed.

This display is only for information purposes and is not editable by the user.

Build up TCP/IP Connection

There are two possibilities to enter the IP address to build up a TCP/IP connection between the Hilscher device and the PC.

- **Scan network for devices**

  Clicking on the **NetIdent Rescan** button, the local Ethernet network is scanned for Hilscher devices. This devices need to support the identification by the Hilscher NetIdent Protocol.

  Devices found during the network scan and which are connectable to the PC are displayed in the table **Board Selection**.

- **Type in IP Address manually**

  If the device to be connected is not located in the local Ethernet network, it is necessary to type in the IP address of the device manually.

  Also some devices do not support the identification by the Hilscher NetIdent Protocol. In this case the IP address of the device has to be typed in manually, too.

Figure 15: CIF TCP/IP Driver - Type in IP Address manually

The IP address of the device to be connected need to be typed in the field **Add IP Address**. Clicking the **Add** button, it is tried to build up a CIF TCP/IP connection between the PC and the device.

If a device with the typed in IP address was found, it is displayed in the table **Board Selection**.

Board Selection

In the table **Board Selection** the devices are displayed, which were found via inserting the IP address or via the Hilscher NetIdent Protocol and which can be connected to the PC.

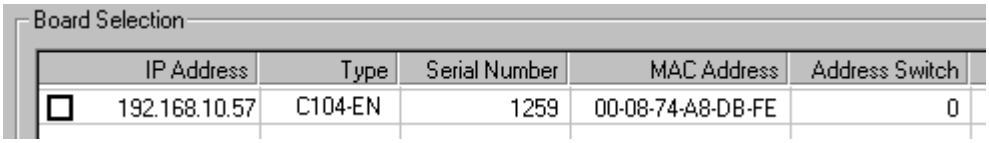

Figure 16: CIF TCP/IP Driver - Board Selection - Found Device

When the device already has an IP address, this is shown in the field **IP Address**.

If the shown IP address is 0.0.0.0, an IP address has to be assigned to the device with the button **Set IP Address**. Further information for changing the IP address you find in section *Change IP Address* on page *36*.

Connect Device

To connect a device to the PC, the checkbox of the favored device has to be selected in front of the **IP Address** field.

Checkbox	Description
☐	Device is still not assigned and it can be selected.
☑	Device is assigned. The Assignment can be abrogated by deselecting.

Table 18: Device Assignment - Checkboxes of the CIF TCP/IP Driver

Note: A connection can be build up to exactly one device.

The following picture shows an assigned device:

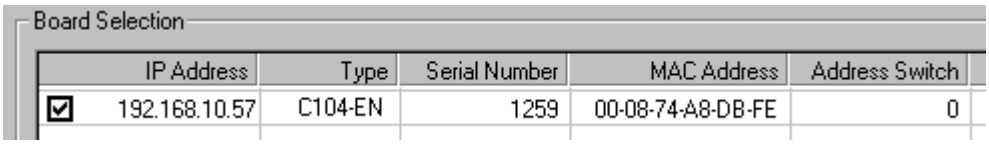

Figure 17: CIF TCP/IP Driver - Board Selection - Assigned Device

This selection has to be confirmed by clicking the **OK** button. Now the device is connected with the System Configurator via the CIF TCP/IP Driver and the Device Assignment dialog is closed.

By clicking the **Cancel** button, the Device Assignment is closed without an assignment has been accomplished or respectively changed.

Filtered Devices

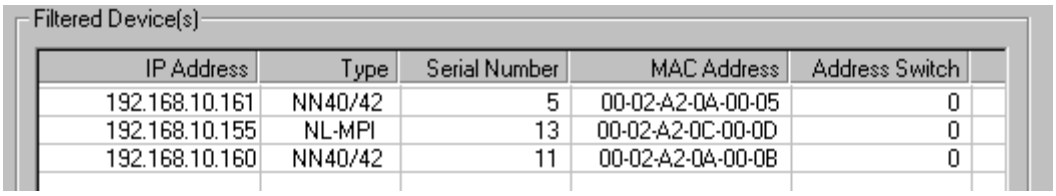

Figure 18: CIF TCP/IP Driver - Filtered Devices

Devices listed in the table **Filtered Device(s)** were found during the network scan in the local Ethernet network, but they can not be assigned, because they belong to another device family.

5.1.4.1 Change IP Address

A new IP address is assigned to a device or respectively an existing IP address of a device is changed via the button **Change IP Address**.

Note: The IP address can only be changed in case of Hilscher devices which are connected directly to the Ethernet and which support the function 'Change IP Address'. These are for example: NL-MPI, NN40, NN42, CIF 104-EN, COM-C-EN, COM-EN.

Therefore the device has to be selected in the table Board Selection by activating the checkbox. Via the **Change IP Address** button the following dialog opens:

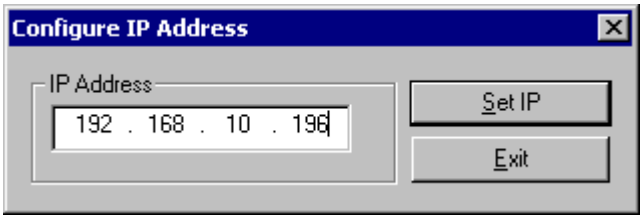

Figure 19: Set IP Address

Type in the IP address for the device and confirm the entry by clicking on the **Set IP** button.

Note: The IP address set by clicking the **Set IP button** is only temporarily adjusted. A permanent storage of the IP address takes place with a download of the configuration from the framework.

5.2 Parameter

The Ethernet/Protocol Parameter are the basis for the working data exchange. This section contains information for setting the Ethernet and Protocol Parameter and a description for the individual parameter.

Select in the following table the protocol, which you inserted in the configuration together with the device. In the given section the settings of the parameters are described.

5.2.1 Ethernet Parameter

Protocol	Described in section	Page
Ethernet	*Setting of the Ethernet Parameter*	*38*
Open Modbus / TCP	*Setting of the Open Modbus / TCP Parameter*	*44*

Table 19: Setting of the Ethernet Parameter

The range of parameter depends on the loaded firmware and its scope of functions. The following table shows the parameters in dependence on the used firmware.

Parameter	Ethernet Device	Open Modbus TCP Device	Described in section	Page
IP Address	Yes	Yes	*IP Address*	*38*
Ethernet	Yes	Yes	*Ethernet*	*39*
SMTP Setup	Yes	No	*SMTP Setup*	*40*
FTP Server Setup	Yes	No	*FTP Server Setup*	*41*
TCP	Yes	No	*TCP*	*43*
Open Modbus/TCP	No	Yes	*Open Modbus / TCP Setup*	*44*

Table 20: Setting of the Ethernet / Open Modbus/TCP Parameter

5.2.2 Protocol Parameter

Protocol	Described in section	Page
ASCII	*Setting of the ASCII Parameter*	*46*
3946R	*Setting of the 3964R Parameter*	*51*
RK512	*Setting of the RK512 Parameter*	*53*
Modbus RTU	*Setting of the Modbus RTU Parameter*	*55*
Modbus Plus	*Setting of the Modbus Plus Parameter*	*57*
Modnet 1/N	*Setting of the Modnet 1/N Parameter*	*61*
Modnet 1/SFB	*Setting of the Modnet 1/SFB Parameter*	*63*

Table 21: Setting of the Protocol Parameter

5.2.3 Setting of the Ethernet Parameter

Via the menu **Settings > Ethernet Parameter** the Ethernet Parameter can be set, if an Ethernet configuration is loaded.

5.2.3.1 IP Address

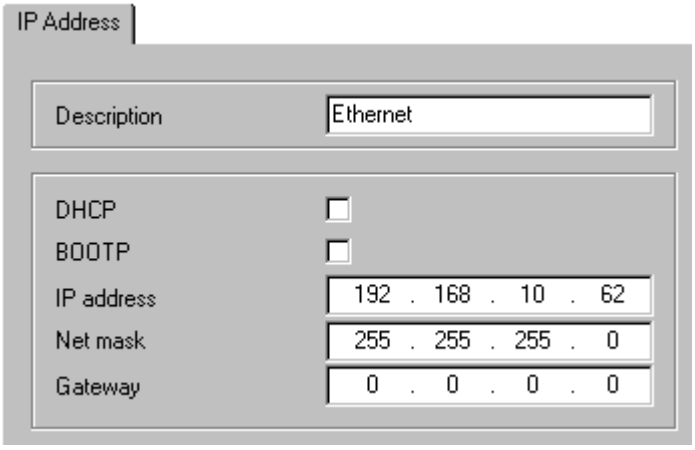

Figure 20: Settings > Ethernet Parameter > IP Address

Description:

The description of the device is shown in SyCon as the name of the device. The description is changeable in this field.

The handing over of the IP parameters (IP address, Net mask, Gateway) can result in three ways.

DHCP:

The device gets the IP parameters from a DHCP server.

BOOTP:

The device gets the IP parameters from a BOOTP server.

IP address, Net mask and Gateway:

The IP parameters can be entered in these fields.

If more than one configuration way is activated (for example DHCP and manually entered IP parameters), the device tries to process the different configuration ways one after the other. As soon as it got an IP configuration in one of this ways, the device starts with these parameters.

5.2.3.2 Ethernet

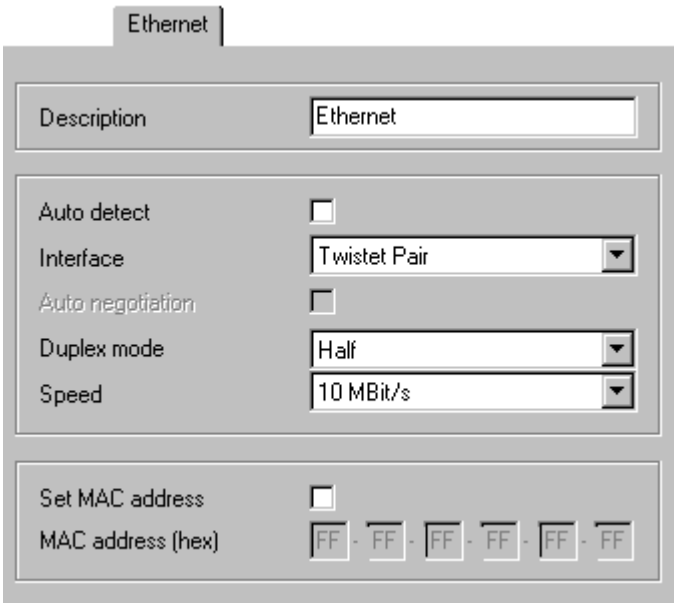

Figure 21: Settings > Ethernet Parameter > Ethernet

Description:

The description of the device is shown in SyCon as the name of the device. The description is changeable in this field.

Auto detect:

An automatic detection of the Ethernet interface results if this option is selected.

Interface:

Here the manually setting of the Ethernet interface can be done. The user has the possibility to select Twisted Pair or AUI but at the moment just Twisted Pair is supported. The description of this you find in section *Twisted Pair and AUI* at page *131*.

Duplex mode:

Duplex mode of the Ethernet interface. You can select between Full Duplex and Half Duplex but only Half Duplex is supported at the moment. The description of this you find in section *Full Duplex and Half Duplex* at page *131*.

Speed:

Transmission speed of the data.

Set MAC address:

If this option is selected you activate the manual MAC address configuration. If this option is not selected the card has the default address which was set by the manufacturer.

MAC address (hex):

Here you can enter the manual settings of the MAC address of the device. The address format is 6 byte in the Hex Code.

5.2.3.3 SMTP Setup

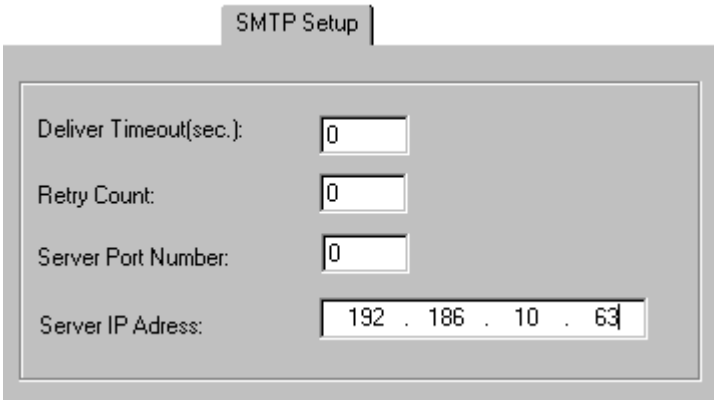

Figure 22: Settings > Ethernet Parameter > SMTP Setup

Deliver Timeout (sec.):

The setting field states the time in seconds, after the command message at the latest will be answered to the user. The message will be answered earlier, if either the message was successfully delivered or the SMTP server rejected the mail, or the TCP connection was closed, for any reason.

Retry Count:

The setting states how often the device retries to deliver the mail. This parameter comes only to the validity, if the SMTP server temporally refused the mail. If there is a try left and the remaining deliver timeout is greater than 30 minutes, the device waits 30 minutes and tries again to deliver the mail.

Server Port Number:

The port number on which the device tries to connect to the SMTP server is set. Although the specification reserves port number 25 for SMTP services, the device allows using another port. If this parameter is 0, a default value of 25 will be used.

Server IP Address:

In this field the IP Address of the Server has to be entered.

5.2.3.4 FTP Server Setup

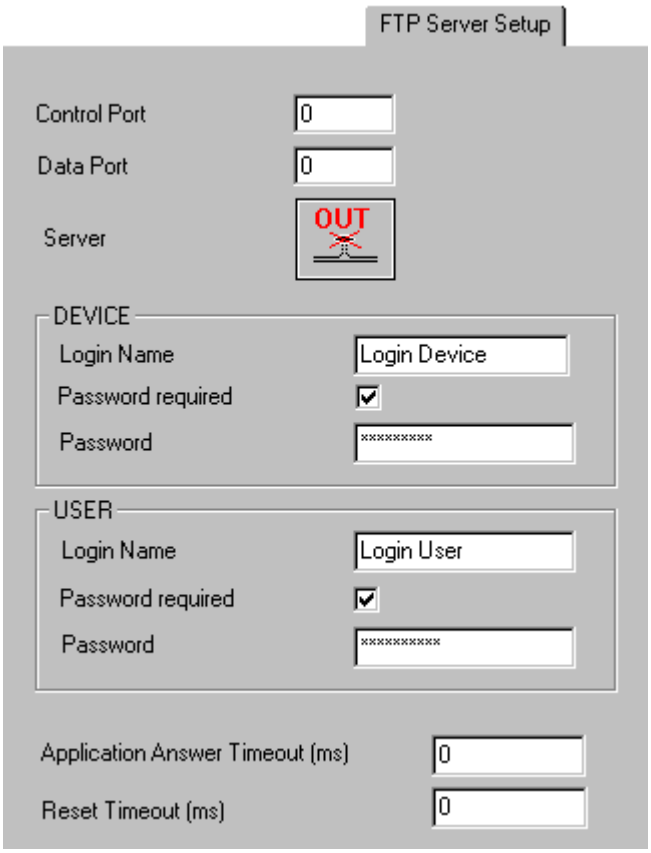

Figure 23: Settings > Ethernet Parameter > FTP Server Setup

Control Port:

The Control Port defines the port number for FTP control connections. Although the specification reserves port 21 for FTP control connections, the value is adjustable. If this parameter is 0, a default value of 21 will be used.

Data Port:

The Data Port defines the port number for FTP data connections. Although the specification reserves port 20 for FTP data connections, the value is adjustable. FTP clients may also change the data port number during a session. If this parameter is 0, a default value of 20 will be used.

Server:

The Server field can be used to turn the FTP server on or off. If the server is turned off, no connections will be opened.

DEVICE:

If the field **Password required** is activated, the login for the device has to be typed in the field **Login** and a password has to be entered in the field **Password**.

USER:

If the field **Password required** is activated, the login for the user has to be typed in the field **Login** and a password has to be entered in the field **Password**.

Application Answer Timeout (ms):

An application must answer a command from the FTP task within a preset time. The Answer Timeout is a timeout for the application in milliseconds. If the application fails to answer an FTP command within time, the device will answer the FTP client with an error. If this value is 0, a default value of 10 seconds will be used.

Reset Timeout (ms):

The Reset Timeout will be used if the device has to perform a reset. The device has to make a COLDSTART after a database is deleted or a database is stored in the device. The device will wait the preset time (in milliseconds), close all TCP sockets and will make a COLDSTART. If Reset Timeout is 0, the device will wait until all FTP clients are disconnected, before closing the remaining TCP connections and reset.

5.2.3.5 TCP

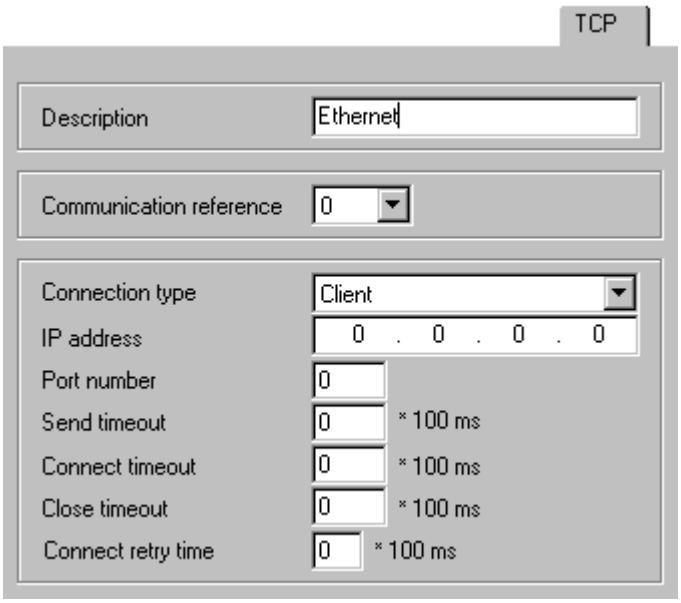

Figure 24: Settings > Ethernet Parameter > TCP

Description:

The description of the device is shown in SyCon as the name of the device. The description is changeable in this field.

With the different **Communication references 0**, **1**, **2** and **3** you are able to set different communication kinds which are independent form each other. By selecting a communication kind the settings for this communication are loaded and the communication is done this way.

In the field **Connection type** you can select between **unused**, **Server** and **Client**. Depending on the selected Connection type the under settings (IP address, Connect timeout and so on) are grey.

If the **Connection type** unused is selected all other field are grey because no connection is selected.

If Server is selected as **Connection type** the field **Port number**, **Send timeout** and **Close timeout** are adjustable.

If Client is selected as **Connection type** all field are adjustable. In the field **IP Address** you have to type in the IP of the wanted Client. The field **Send timeout**, **Connect timeout**, **Close timeout** and **Connect retry time** are adjustable in multiple of 100 milliseconds.

5.2.4 Setting of the Open Modbus / TCP Parameter

Via the menu **Settings > Open Modbus / TCP Parameter** the Open Modbus / TCP Parameter can be set, if an Open Modbus / TCP configuration is loaded.

5.2.4.1 IP Address

See section *IP Address* at page *38* in the chapter about Ethernet parameters.

5.2.4.2 Ethernet

See section *Ethernet* at page *39* in the chapter about Ethernet parameters.

5.2.4.3 Open Modbus / TCP Setup

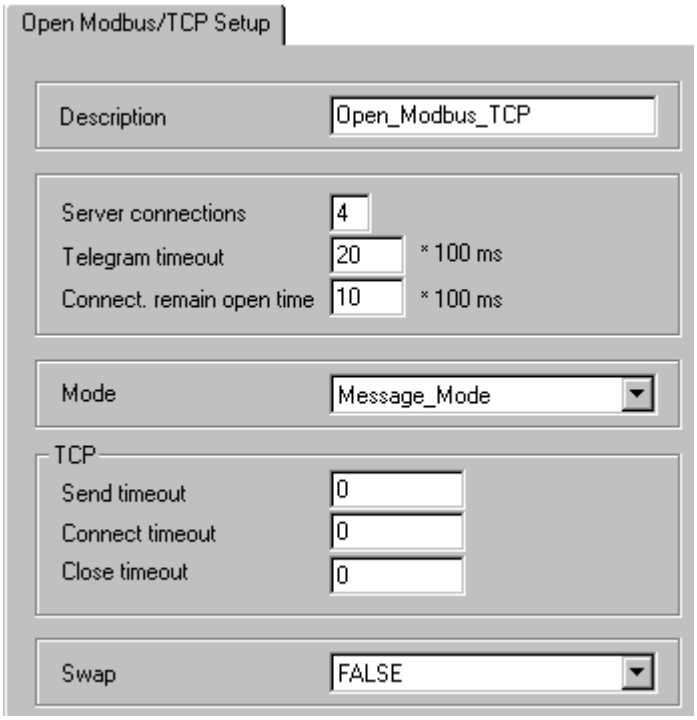

Figure 25: Settings > Open Modbus / TCP Parameter > Open Modbus / TCP Setup

Description:

The description of the device is shown in SyCon as the name of the device. The description is changeable in this field.

Server connections:

You can set up max. 4 Server connections simultaneously.

Telegram timeout:

Only valid for Client orders in the message mode. The order is stopped by the application when no answer results from the couple partner depending on the set time in this field.

Connect. remain open time:

Only valid for Client orders in the message mode. The connection to the Server is maintained depending on the set time and after expiration of this time it is stopped, if no new order was sent by the application.

Mode:

If the card works in the **Message Mode**, it means the it works in Client and Server mode. The application sends messages to the CIF to transmit data.

If the card works in the I/O Mode, it means that it works only in the Server mode. The application exchanges data with the CIF via the data image in the DPM.

Send timeout:

This is the Send timeout for the TCP connection. It is only used internally. In this field you type in the timeout, how long it is tried to take off the orders via TCP/IP.

Connect timeout:

This is the connect timeout for the TCP connection. It is only used internally. Here you give the timeout, how long the TCP Task tries to build up a connection to the Server.

Close timeout:

This is the close timeout for the TCP connection built up. This is only used internally. In this field you type in the timeout, how long the TCP Task tries to build up a connection.

Swap:

Data format, filling of the I/O data in the DPM. Motorola / Intel Format.

5.2.5 Setting of the ASCII Parameter

Via the menu **Settings > ASCII Parameter** the ASCII Parameter can be set, if an ASCII configuration is loaded.

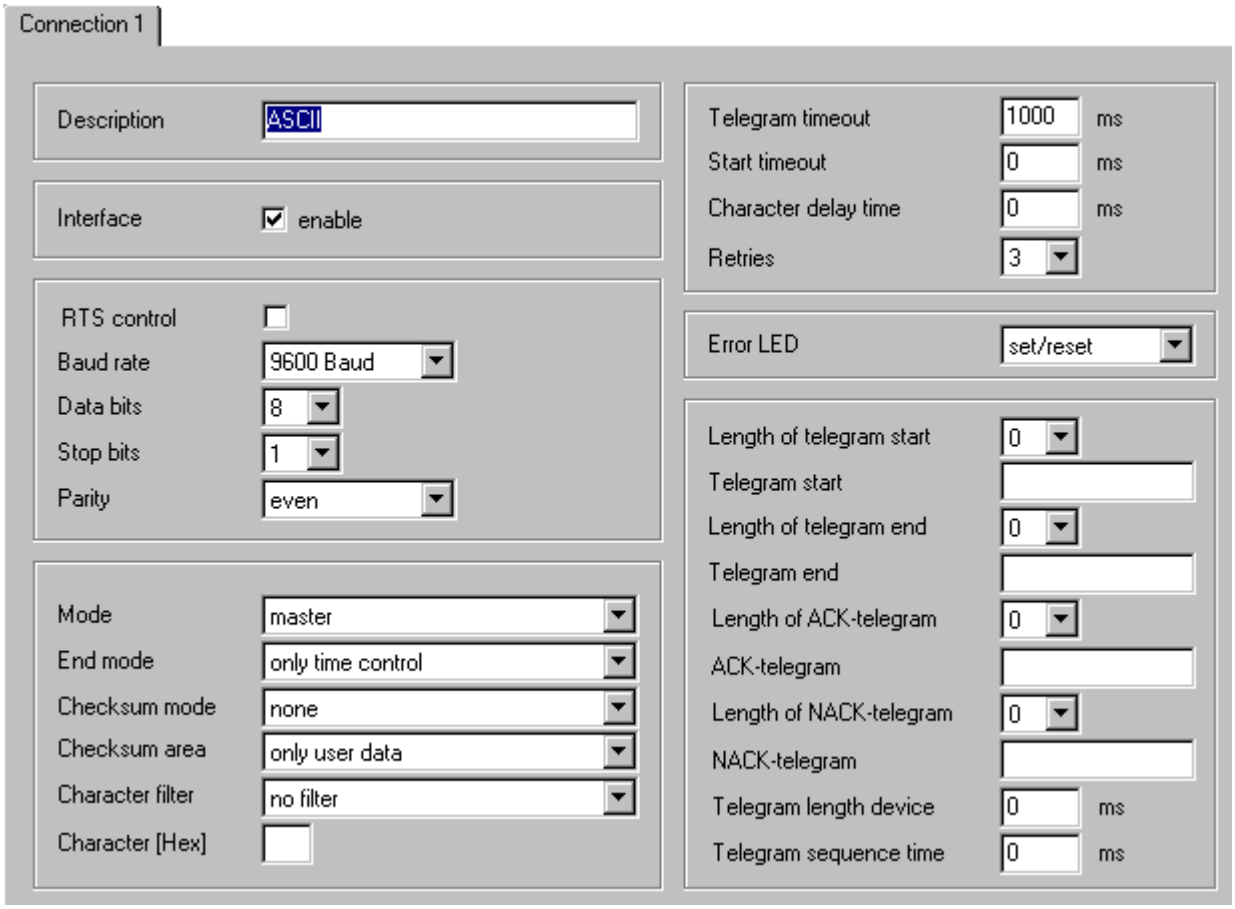

Figure 26: Settings > ASCII Parameter > Connection 1

Description:

The description of the device is shown in SyCon as the name of the device. The description is changeable in this field.

Interface:

Interface of the device which is served by the protocol. This option can be activated or deactivated by the user.

RTS control:

The RTS control must be activated (selected) if you use a RS485 interface. If you use a RS422 or RS232 interface you need not to select this option.

Baud rate:

Determination of the transmission rate. You have the possibility the select a Baud rate between 50 Baud and 38400 Baud. As default setting 9600 Baud is selected this is used mostly.

Data bits:

Number of data bits. You can select **7** data bits or **8** data bits.

Stop bits:

Number of stop bits. You can select **1** stop bit or 2 stop bits.

Parity:

Determination of the Parity bit. You can select **no**, **even** and **odd**.

Mode:

The protocol works in two different modes: Master and Slave mode.

Master mode: The data transmission is introduced by the ASCII protocol. Data is sent to the remote partner and the answers are read in and transmit to the device internal processing.

Slave mode: The remote partner sends a telegram which is received and passed on by the ASCII protocol. Depending on the application related design it is already answered at the protocol layer with a positive acknowledgement and in case of a transmission error with a negative acknowledgement. After the processing of the telegram an answer telegram can be hand out to the remote partner.

These modes are selected to avoid initialization conflicts by simultaneous sending. This has to be controlled at the protocol layer by the user.

In the Master mode the remote partner is only allowed to answer, if it has received a telegram before. Just then the ASCII protocol is ready to receive.

In the Slave mode the ASCII protocol is always ready for send and receives. As a rule the telegram transfer is started by the Master, which waits for an answer afterwards before it sends the next telegram. In case of simultaneous receiving of the receive telegram and the transmit job the receive telegram has priority. The data transmission in this mode can be realized only from one transmission direction.

During the ASCII protocol sends data it is not ready to receive. Continuous receiving of data suppresses the send mode. The Master device and the user respectively are responsible for the synchronization.

End mode:

The receive direction of the telegram is fixed over the end mode. At the moment the following end criterions are configurable:

only time control: As long as the set time control allows this additional characters can be received. The ending of the time control is not viewed as an error but as the end of the telegram.

end identifier: The end of a telegram is recognized at the character 'telegram end'. The time control is active corresponding to the set time.

acknowledge telegram: For the sent telegram only the configured acknowledge telegrams 'ACK telegram' and 'NACK telegram' are expected. The time control is active corresponding to the set time. Because in the Slave mode no acknowledge telegrams are defined in the receive direction, this mode is not allowed here.

end identifier / acknowledge telegram: This setting is the combination of the both last modes. That means the received data is controlled for a valid acknowledge telegram and for the telegram end. The control time is active according to the set time. Because in the Slave mode no acknowledge is defined, this mode is not allowed.

fixed data count: The end of the receive telegram is fixed by the 'telegram length device'. The time control is active according to the set time.

pass data count forward: The end of the receive telegram is defined by a passed forward data count in the send task. This is only possible in the Master mode.

Checksum mode:

The ASCII protocol is able to calculate the checksum for the telegrams which have to be send and it can insert the user data by itself. It also can check the checksum of the received telegrams. For these different processes of checksum mode and telegram possibilities for the calculation are configurable:

none: No checksum is determined and the send- and receive telegram get no checksum.

binary 7 bit: The checksum is formed by a byte fashioned addition of all data without overflow. The upper bit (D7) is firmed at 0.

binary 8 bit: The checksum is formed by a byte fashioned addition of all data without overflow.

CRC: The checksum is formed by 'exclusive or' of all data.

CRC in ASCII: The checksum is formed by 'exclusive or' of all data and afterwards converted into ASCII.

Checksum area:

Defines the telegram part by which the checksum is formed. You can select the following settings: **only user data**, **with start identifier, with end identifier** or **complete telegram**.

Character filter:

Some protocol use defined characters to mark the telegram end. If these characters are allowed inside the user data a distinction to the end character can be made by doubling the character. If the character appears inside the user data it is sent two times.

By receiving the character two times directly one after the other it is reject one time. If it was received just one time it is the telegram end.

This function can be activated in the parameter 'character filter'. Additional the character which has to be filtered must be set.

You have to take care that the ASCII protocol leads the filter function only inside the user data. The telegram start or the telegram end and the acknowledge telegram have to be provided with a doubled character and respectively as end mode with simple characters. The doubled characters are included for the ascertaining of the checksum. The checksum is unlessed from the filter function.

Character (Hex):

Gives the character for the active filter. The input is always hexadecimal.

Telegram timeout:

The telegram timeout is activated by switching on the receive mode and controls the receive of the hole telegram. It has priority over the 'Start timeout' and the 'character delay time'.

Start timeout:

The start timeout is activated with the switching on of the receive mode and controls the time until the receiving of the first character. Requirement for this is that the 'telegram control time' is configured with 0.

Character delay time:

Controls the time between the receiving of the single characters. Requirement is that the 'telegram control time' is configured with 0.

Retries:

Number of telegram retries in the case of errors.

Error LED:

Operating mode of the error LED. It can be set: **set/delete** or only **set**.

Length of telegram start:

Length of the telegram start. In case of length = 0 no start designation is sent. In case of negative length the belonging text is interpreted as text as hex characters of 0...F. Otherwise it is used as ASCII text.

Telegram start and Telegram end:

Some transmission protocols expect defined characters at the beginning and at the end of the telegram. These characters are no user data but serve for the identification of the telegram start respectively the telegram end. This data can be independently add or delete by the ASCII protocol. For this a telegram start and a telegram end can be configured. These can be max. 8 characters long.

Length of telegram end:

Length of the telegram end identification. In case of length = 0 no end identification is sent. In case of negative length the belonging text is interpreted as hex character of 0...F. Otherwise it is used as ASCII text.

Length of ACK-telegram:

Length of the acknowledge telegram. In case of length = 0 no acknowledge telegram is sent. In case of negative length the belonging text is interpreted as hex character of 0...F. Otherwise it is used as ASCII text.

ACK-telegram:

Acknowledge telegram exists of 0-8 characters.

Length of NACK-telegram:

Length of the not acknowledge telegram. In case of length = 0 no not acknowledge telegram is sent. In case of negative length the belonging text is interpreted as hex character of 0...F. Otherwise it is used as ASCII text.

NACK-telegram:

Not acknowledge telegram exists of 0-8 characters.

Telegram length device:

Defines the telegram length of the remote partner when it is valid as ended. Just in the end mode the 'fixed data number' of importance.

Telegram sequence time:

With the parameter 'telegram sequence time' it is possible to definite a minimum time between the sending. In case the parameter is higher than 0 the ASCII telegram waits the defined time until it sends the next telegram. It counts the time from telegram start to telegram start. If more telegrams are sent one after the other these are hand out buffered and in the time of the 'telegram sequence time'.

5.2.6 Setting of the 3964R Parameter

Via the menu **Settings > 3964R Parameter** the 3964R Parameter can be set, if a 3964R configuration is loaded.

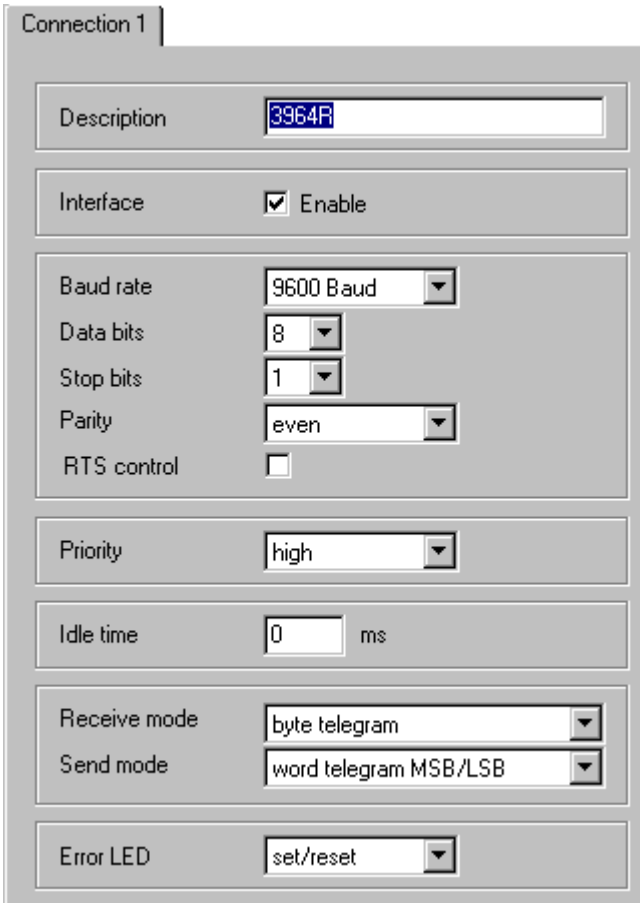

Figure 27: Settings > 3964R Parameter > Connection 1

Description:

The description of the device is shown in SyCon as the name of the device. The description is changeable in this field.

Interface:

Interface of the device which is served by the protocol. This option can be activated or deactivated by the user.

Baud rate:

Determination of the transmission rate. You have the possibility the select a Baud rate between 50 Baud and 38400 Baud. As default setting 9600 Baud is selected this is used mostly.

Data bits:

Number of data bits. You can select **7** data bits or **8** data bits.

Stop bits:

Number of stop bits. You can select **1** stop bit or 2 stop bits.

Parity:

Determination of the Parity bit. You can select **no**, **even** and **odd**.

RTS control:

The RTS control must be activated (selected) if you use a RS485 interface. If you use a RS422 or RS232 interface you need not to select this option.

Priority:

Fixes which device is put back the send telegram in case of an initialization conflict. The priority can be set to **low** or **high**.

Idle time:

Idle time in milliseconds before the start of a telegram repeat. The range of value is between 0 and 10000.

Receive mode:

Defines the type of the receive data. The following settings can be selected: **word telegram MSB/LSB**, **word telegram LSB/MSB**, **byte telegram** or **transparent**.

Send mode:

Defines how the word telegrams are sent of the line. **Either word telegram MSB/LSB** or **word telegram LSB/MSB**.

Error LED:

Operating mode of the error LED. It can be set: **set/delete** or only **set**.

5.2.7 Setting of the RK512 Parameter

Via the menu **Settings > RK512 Parameter** the 3964R Parameter can be set, if an RK512 configuration is loaded.

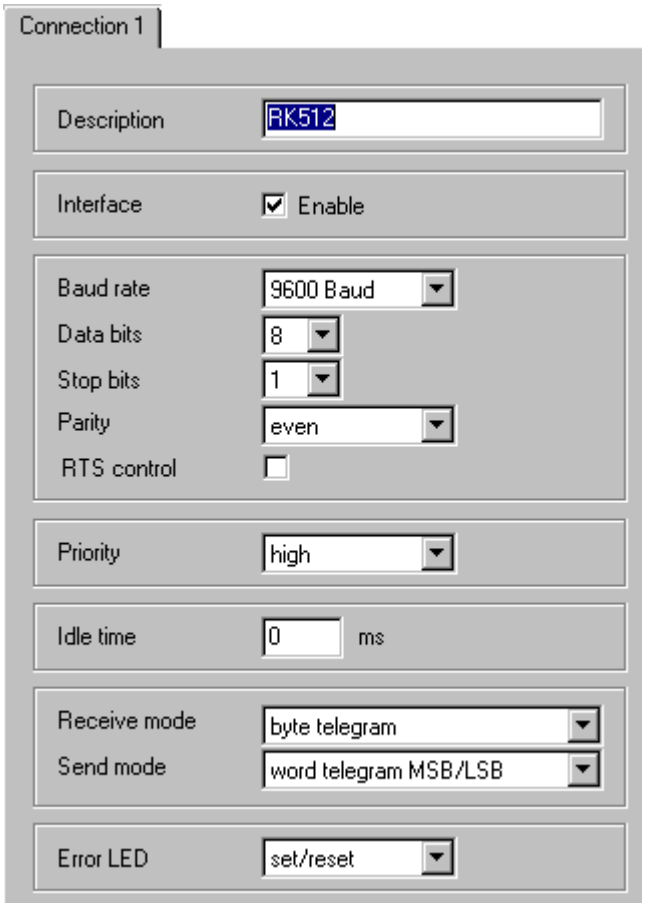

Figure 28: Settings > RK512 Parameter > Connection 1

Description:

The description of the device is shown in SyCon as the name of the device. The description is changeable in this field.

Interface:

Interface of the device which is served by the protocol. This option can be activated or deactivated by the user.

Baud rate:

Determination of the transmission rate. You have the possibility the select a Baud rate between 50 Baud and 38400 Baud. As default setting 9600 Baud is selected this is used mostly.

Data bits:

Number of data bits. You can select **7** data bits or **8** data bits.

Stop bits:

Number of stop bits. You can select **1** stop bit or 2 stop bits.

Parity:

Determination of the Parity bit. You can select **no**, **even** and **odd**.

RTS control:

The RTS control must be activated (selected) if you use a RS485 interface. If you use a RS422 or RS232 interface you need not to select this option.

Priority:

Fixes which device is put back the send telegram in case of an initialization conflict. The priority can be set to **low** or **high**.

Idle time:

Idle time in milliseconds before the start of a telegram repeat. The range of value is between 0 and 10000.

Receive mode:

Defines the type of the receive data. The following settings can be selected: **word telegram MSB/LSB**, **word telegram LSB/MSB**, **byte telegram** or **transparent**.

Send mode:

Defines how the word telegrams are sent of the line. **Either word telegram MSB/LSB** or **word telegram LSB/MSB**.

Error LED:

Operating mode of the error LED. It can be set: **set/delete** or only **set**.

5.2.8 Setting of the Modbus RTU Parameter

Via the menu **Settings > Modbus RTU Parameter** the Modbus RTU Parameter can be set, if a Modbus RTU configuration is loaded.

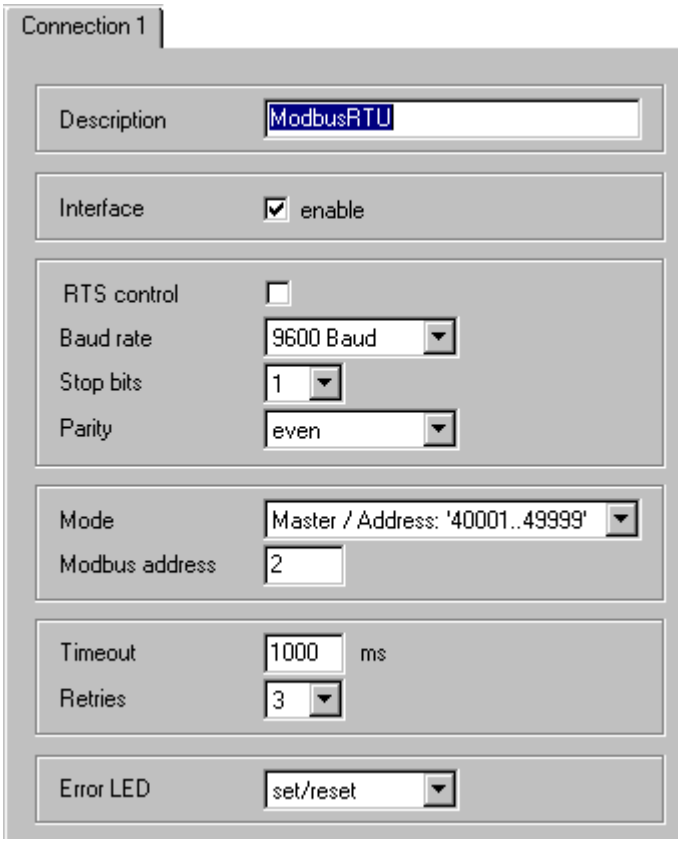

Figure 29: Settings > Modbus RTU Parameter > Connection 1

Description:

The description of the device is shown in SyCon as the name of the device. The description is changeable in this field.

Interface:

Interface of the device which is served by the protocol. This option can be activated or deactivated by the user.

RTS control:

The RTS control must be activated (selected) if you use a RS485 interface. If you use a RS422 or RS232 interface you need not to select this option.

Baud rate:

Determination of the transmission rate. You have the possibility the select a Baud rate between 50 Baud and 38400 Baud. As default setting 9600 Baud is selected this is used mostly.

Stop bits:

Number of stop bits. You can select **1** stop bit or **2** stop bits.

Parity:

Determination of the Parity bit. You can select **no**, **even** and **odd**.

Mode:

Defines the operation mode. It makes a difference between Slave- and Master mode. Defines additionally the valid address sector of the data.

Slave / Address: 40001-49999

Master / Address: 40001-49999 (Default setting)

Slave / Address: 1-65535

Master / Address: 1-65535

Slave / Address: 0-65535

Master / Address: 0-65535

Modbus address:

Gives the state of the address at the Modbus. The range of value is 1..2..247. 2 is the default setting.

Timeout:

Master mode: Gives the maximum time in milliseconds how long the Master waits for an answer telegram of the Slave.

Slave mode: Gives the maximum time in milliseconds how long the Master waits for an answer telegram of the remote partner.

The range of value is 1 to 10000 and 1000 is the default setting.

Retries:

Defines the number of telegram retries in the case of errors. Only of importance for the Master mode. The range of value is 1 to 10 and 3 is the default setting.

Error LED:

Operating mode of the Error LED. You can set **set/delete** or only **set**.

5.2.9 Setting of the Modbus Plus Parameter

Via the menu **Settings > Modbus Plus Parameter** the Modbus Plus Parameter can be set, if a Modbus Plus Configuration is loaded.

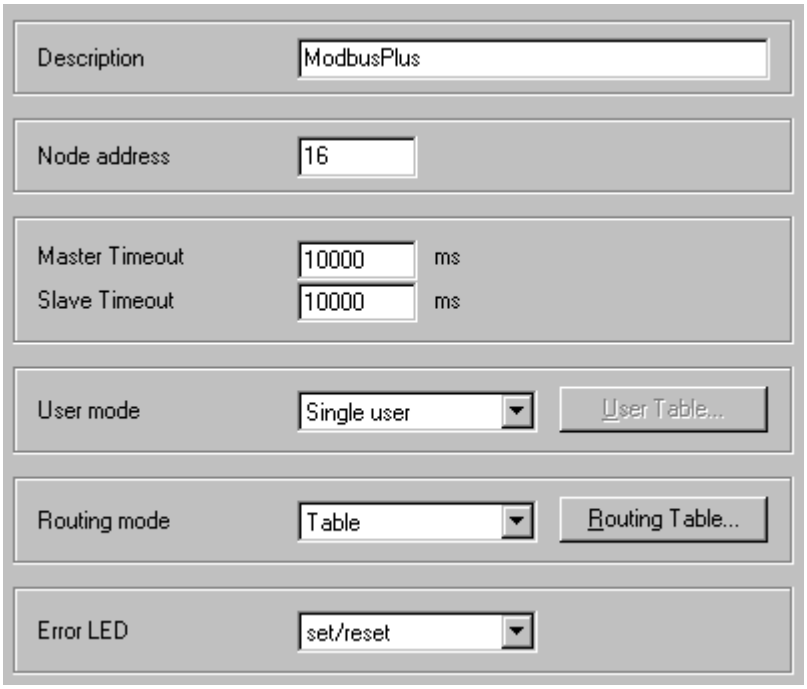

Figure 30: Settings > Modbus Plus Parameter

Description:

The description of the device is shown in SyCon as the name of the device. The description is changeable in this field.

Node address:

Stipulates the Node address in the Modbus Plus network.

Master Timeout:

Fixes the Timeout for Master orders (Client orders are from the 'user'). In this time the Modbus Plus couple partner must have answered. Otherwise the Client order is stopped and the user gets a Timeout error.

Slave Timeout:

Gives the Timeout for Slave orders ('user' is Server). In this time the 'user' must have answered a Server order. Otherwise the Server order is stopped and the Modbus Plus couple partner (Client) gets an Exception.

User mode:

Gives the User mode.

If the mode **Single user** is selected, exactly one 'user' exists, who communicates with the Modbus Plus Task via the user interface. There eight Modbus Plus paths are assigned to this 'user'.

If the mode **Multi user** is selected, more 'users' exist, who communicate with the Modbus Plus Task via the user interface. There each 'user' gets exactly one Modbus Plus path. See also section *User Table* on page *59* for more information about this.

Further information you find in the Protocol Manual *Modbus Plus Coupling*, chapter *Parameter 'User-Mode'*.

Routing mode:

Gives the routing mode.

If you select the routing **Table** an index is given to the Modbus Plus Task. For this index (logical address) 5 routing bytes are put in the routing table. In section *Routing Table* on page *59* you find more information about this.

If you select the routing **Implicit** only the first or the first and second routing byte are given to the Modbus Plus Task. The other routing bytes are zero.

Further information you find in the Protocol Manual *Modbus Plus Coupling*, chapter *Parameter Routing Mode'*.

Error LED:

Operating mode of the Error LED. You can set **set/delete** or only **set**.

5.2.9.1 User Table

The **User Table** determines in the **User mode 'Multi user'** how the paths are used. Per path can be set Client, Server or unused. The combination Client and Server is not possible.

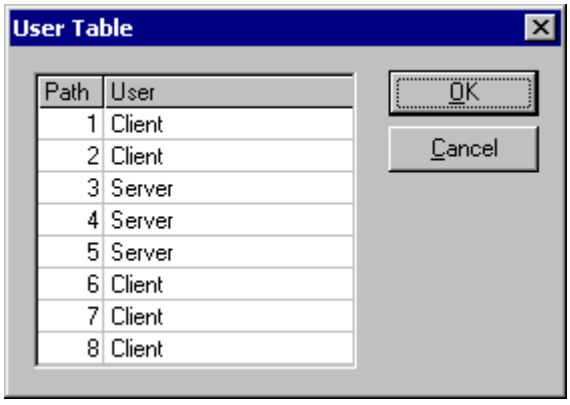

Figure 31: Modbus Plus Parameter - User Table

5.2.9.2 Routing Table

The five routing bytes for the **Routing mode 'Table'** can be determinded to a logical address (**Log Adr.**) in the **Routing Table**.

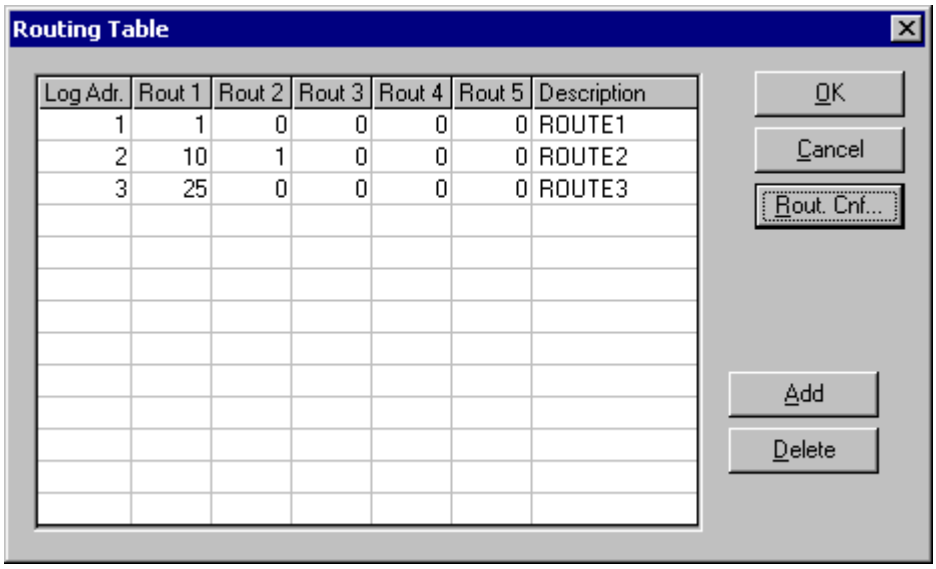

Figure 32: Modbus Plus Parameter - Routing Table

Via the button **Rout. Cnf...** (Routing Configuration) a routing entry can be edited. The entry which should be edit need to be selected. See also section *Routing Configuration* on page *60* for this.

Via the **Add** button a new entry can be created in the routing table.

With the button **Delete** a entry which was selected before can be removed from the routing table.

5.2.9.3 Routing Configuration

In this dialog it is possible to edit a routing entry. In addition to the logical address the five routing bytes and a description can be entered.

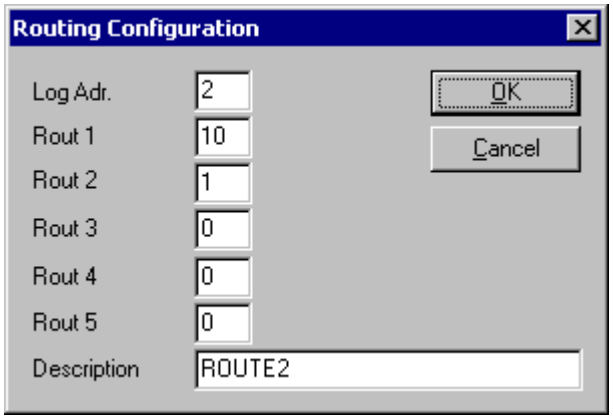

Figure 33: Modbus Plus Parameter - Routing Table - Routing Configuration

5.2.10 Setting of the Modnet 1/N Parameter

Via the menu **Settings > Modnet 1/N Parameter** the Modnet 1/N Parameter can be set, if a Modnet 1/N Configuration is loaded.

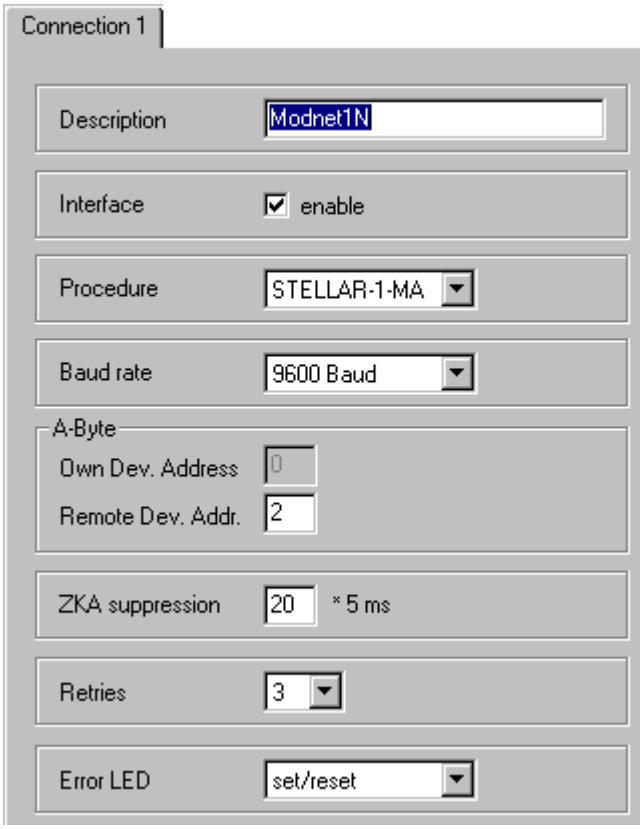

Figure 34: Settings > Modnet 1/N Parameter > Connection 1

Description:

The description of the device is shown in SyCon as the name of the device. The description is changeable in this field.

Interface:

Interface of the device which is served by the protocol. This option can be activated or deactivated by the user.

Procedure:

The procedure information defines how the protocol behaves. Doing this you have to make a distinction between Master and Slave and star- and bus mode.

Baud rate:

Determination of the transmission rate. You have the possibility the select a Baud rate between 50 Baud and 38400 Baud. As default setting 9600 Baud is selected this is used mostly.

Own Dev. Address:

In this parameter the own device address (A-Byte) is taken. This is used in the case of a Slave device to identify a call and to enter it in the telegram header. The value 0 is only permissible in case of a Master device. For Slaves the range of value is 1 to 126.

Remote Dev. Addr.:

In this parameter the device address of the remote device (A-Byte) is entered. For a bus Master without any meaning: look parameter 'active Slaves'. The range of value is 0..2..126.

ZKA suppression:

This parameter fixes the suppression of the cyclic short scan (ZKA) for the Slave. With this you give the Slave time to lead its internal processing's. If the ZKA suppression is set to 0, after every answer of the Slave a question telegram is sent. The value is given in 5 milliseconds units. This corresponds to the AEG-configuration. The ZKA time has to be set same for both devices at the data transmission section! It is not allowed to set the value to 0 for the Master. The range of value is 0..2..250.

Retries:

With this specification the telegram retries in the case of errors can be activated. For the bus Master the poll telegrams are retried to the same Slave with this number and than to the next Slave.

Error LED:

Operating mode of the error LED. You can select **set/delete** or only **set**.

5.2.11 Setting of the Modnet 1/SFB Parameter

Via the menu **Settings > Modnet 1/SFB Parameter** the Modnet 1/SFB Parameter can be set, if a Modnet 1/SFB Configuration is loaded.

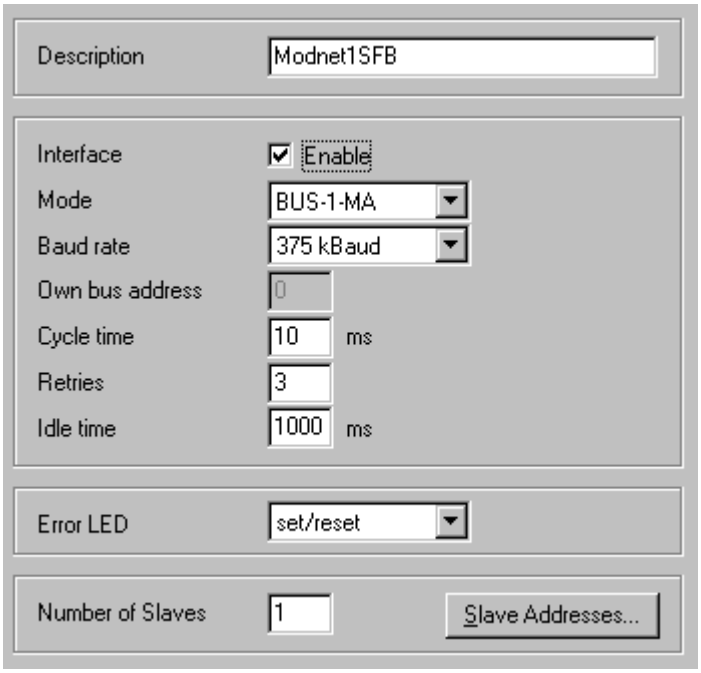

Figure 35: Settings > Modnet 1/SFB Parameter

Description:

The description of the device is shown in SyCon as the name of the device. The description is changeable in this field.

Interface:

Interface of the device which is served by the protocol. This option can be activated or deactivated by the user.

Mode:

Defines the operation mode. It makes a difference between Slave- and Master mode.

Baud rate:

Determination of the transmission rate. You have the possibility the select a Baud rate between 62,5 kBaud and 375 kBaud.

Own bus address:

In this field the own device address is laid down. The range of value is between 0 and 126.

Cycle time:

With this time the protocol asks itself for the projected Slave stations in the Master mode. The range of value is between 1 and 1000 milliseconds, whereby 10 milliseconds is the standard value.

Retries:

Defines the number of telegram retries in case of errors. This is only of importance for the Master mode. The range of value is 1 to 10 and 3 is the default setting.

Idle time:

Idle time in milliseconds before the start of a telegram repeat. The range of value is between 0 and 10000.

Error LED:

Operating mode of the error LED. You can select **set/delete** or only **set**.

Number of Slaves:

Here is set how much of the following Slaves should be scanned cyclically in the Master mode. 1 to 32 Slaves can be scanned, whereby 1 is the default setting.

5.2.11.1 Slave Addresses

The addresses of the Slave devices in the **'Master' Mode** are set in this dialog.

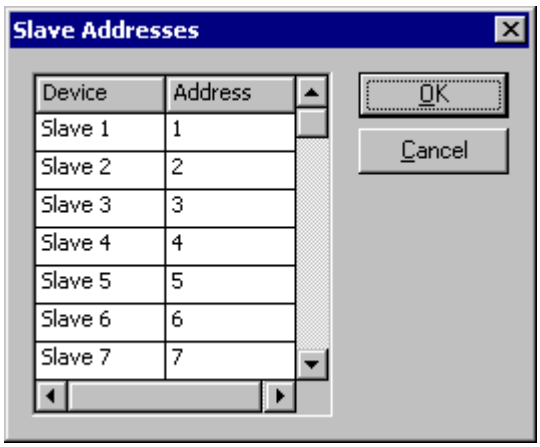

Figure 36: Modnet 1/SFB Parameter - Slave Addresses

5.3 Device Settings

To enter the Device Settings, choose the **Settings > Device Settings** or click with the right mouse button on the corresponding device symbol and select from the list that opens up.

The Device Settings contain parameters that determine the behaviour of the device as well as the user interface. These settings are only valid for Hilscher devices and are included in the download of the configuration.

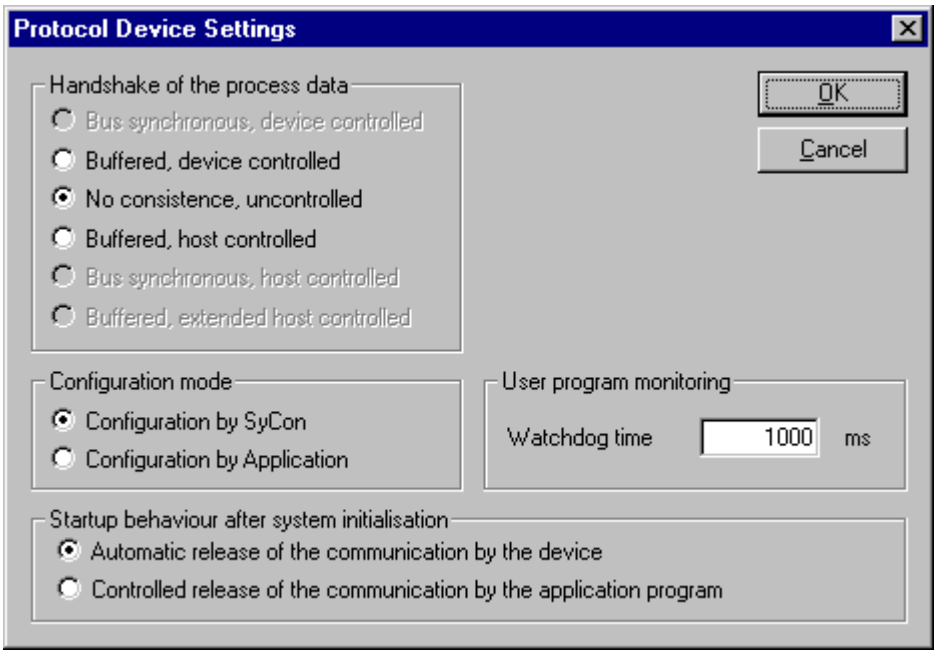

Figure 37: Settings > Device Settings

- **Handshake of the process data**

  These various types are used for setting the handshake of the process data for the device. The choice of used type is important for the correct data exchange between the application program and the device.

  The used handshake of the process data needs to be supported by the application program. Mostly the buffered, host controlled handshake is supported. The setting no consistence, uncontrolled works without handshake and the processes run free.

- **Configuration mode**

  If the device is to use the parameters of the configuration that is downloaded from SyCon then the **Configuration by SyCon** mode must be selected for the configuration mode. If the configuration is written online from an application into the Dual-port memory, then the **Configuration by Application** mode must be selected.

- **User program monitoring**

  The **Watchdog time** determines how long the device waits for a triggering of the software watchdog by the application program until it sets the outputs of the Slave devices to zero. This behaviour must be activated by the user program and does not start automatically.

- **Startup behaviour after system initialization**

  When **Automatic release of the communication by the device** has been set, the Master device starts with the data exchange at the bus after the initializing has been ended. When **Controlled release of communication by the application program** has been set, the application program must activate the data exchange at the bus.

Note: This is not a special Ethernet/Protocol function. But the possibility of choosing in this window depends on the used protocol.

5.4 Project Information

If the user creates a project, the project information can be typed in into the **Settings > Project Information** menu. Anybody can then read this entry when this menu is called up.

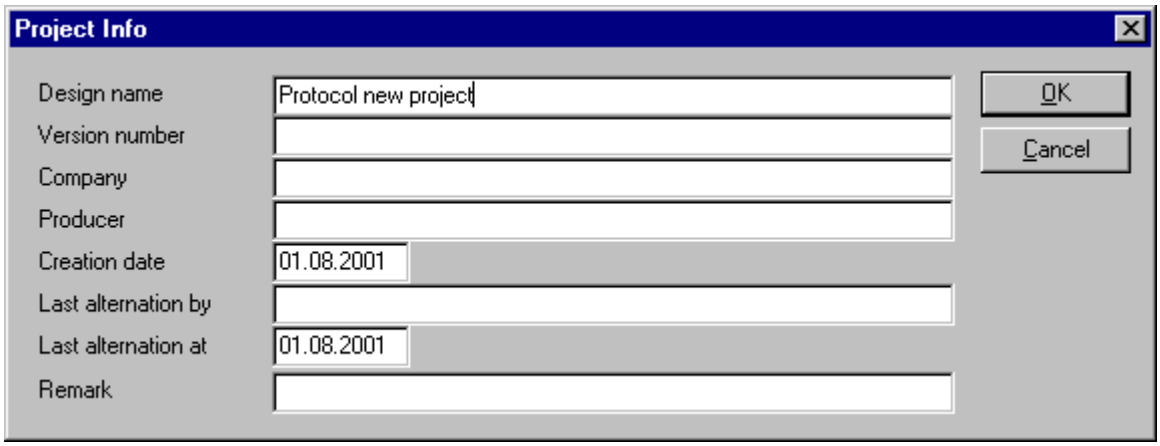

Figure 38: Settings > Project Information

By clicking the **OK** button the Project Information is saved.

5.5 Language

Choose the **Settings > Language** menu and the following window opens:

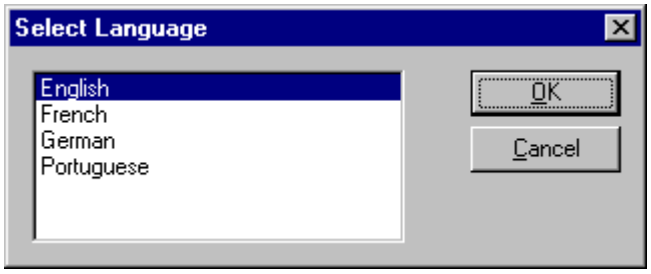

Figure 39: Settings > Language

Here you have the possibility of setting the language of the System Configurator. Select the desired language and confirm the entry with the **OK** button.

A message appears that the System Configurator must be started again in order to activate the selected language. Please carry this out.

After restarting the System Configurator, the language will have changed to the one selected.

Note: Up to now not all languages are available for all SyCon modules!

5.6 Start Options

Starting from the window Network View (menu **Window > Network View**) the menu **Settings > Start...** opens the window **Start Options**. The different start options or modes can be set. Some of these settings are only for the OPC server.

Note: This menu option Start Options is only displayed in the selection Settings, if a project is loaded.

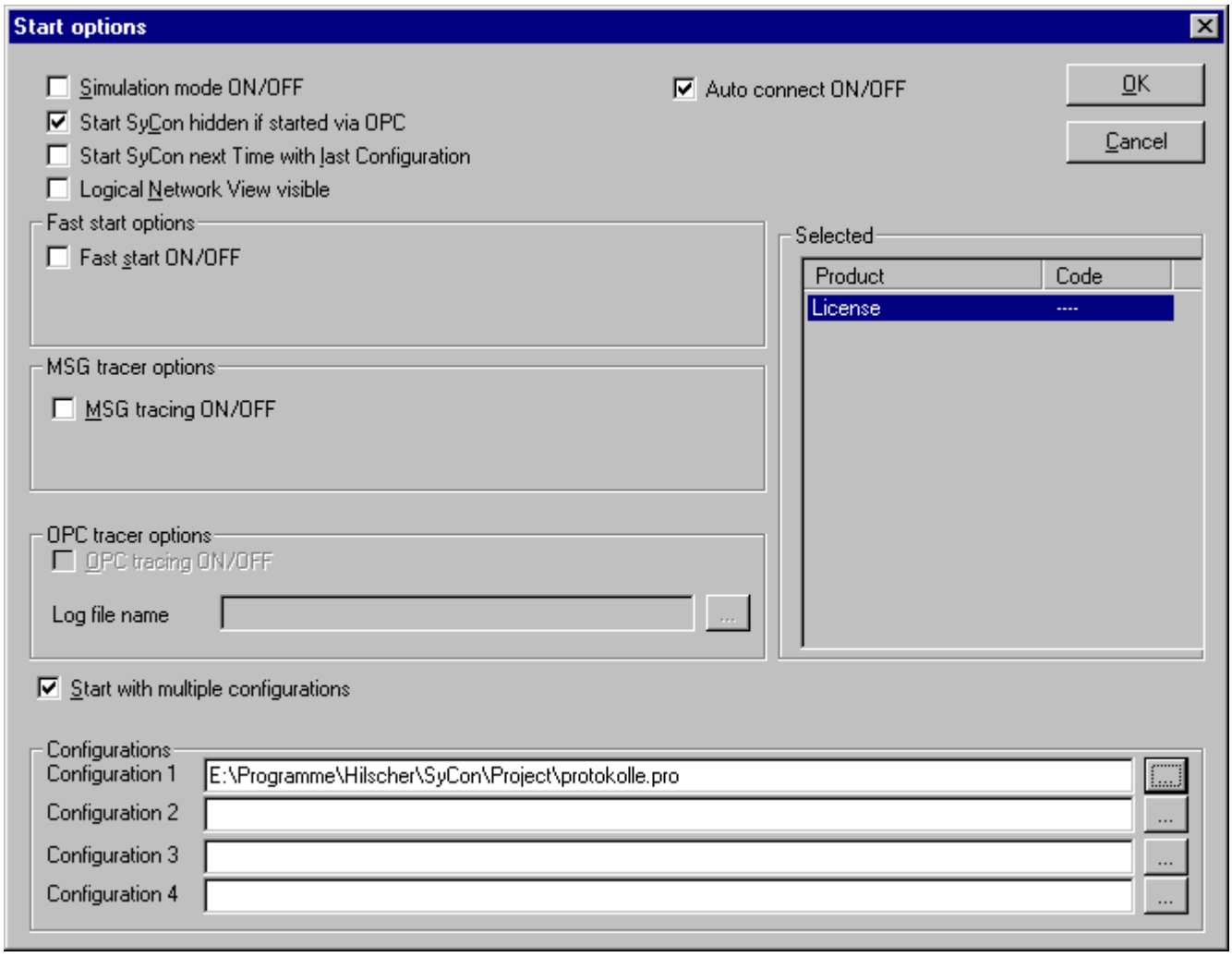

Figure 40: Settings > Start Options

- **Simulation mode ON/OFF**

  Only valid for the OPC Server.

- **Start SyCon hidden if started via OPC**

  Only valid for the OPC Server.

- **Start SyCon next time with last Configuration**

  When this is marked the last saved configuration in the SyCon is automatically loaded when the SyCon is started again.

- **Logic Network View visible**

  When this is marked, there is the possibility of diverting to the network mode without having to install the SyCon with OPC. It is also possible to use the Watch List from the network mode.

- **Fast start ON/OFF**

  Only valid for the OPC Server.

- **TAG tracing ON/OFF**

  Only valid for the OPC Server.

- **OPC tracing ON/OFF**

  Only valid for the OPC Server.

- **Auto connect ON/OFF**

  If this is marked, when opening a configuration automatically a connection to that Hilscher devices is manufactured without the device assignment additionally have to be executed.

- **Start with multiple configurations**

  If this option is selected you have the possibility to start SyCon with up to four configurations simultaneously. The paths are shown in the window and they are changeable there.

6 Online Functions

6.1 Introduction

In this section all the functions that directly influence Hilscher Ethernet/Protocol devices, e.g. CIF 104-EN, CIF 30-EN are presented.

Note: Please note that this also permits an interruption of the running communication or that input and output can be switched On or Off.

6.2 Online to the CIF

6.2.1 Downloading the Configuration

First, the requested device must be chosen for downloading by a left mouse click on the symbol of the device.

In order to release the configuration and network access, a transfer (Download) to the CIF and COM devices must be carried out on the **Online > Download** menu. A warning will appear that the device will interrupt the communication. This warning must be confirmed with **Yes**.

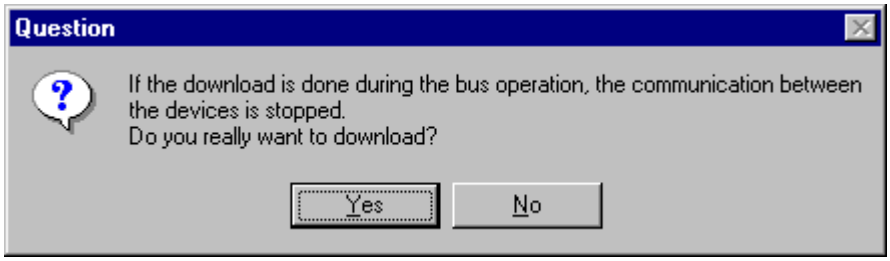

Figure 41: Security question before download

Attention: The download overwrites the configuration in the device and the communication with the connected devices is interrupted.

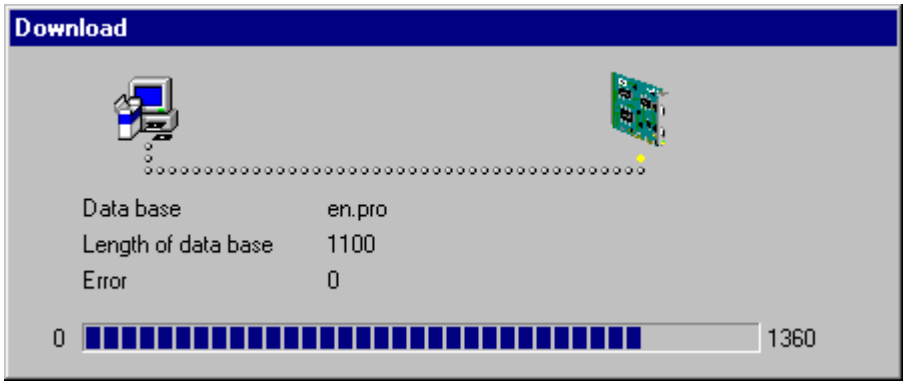

Figure 42: Online > Download

Before the Download is executed, the configuration is checked by the Configurator.

The configuration is transferred into the selected device and is stored there in FLASH memory in a zero voltage manner so that the configuration is available when the power supply is switched off and on again.

After the download, the device carries out an internal restart.

6.2.2 Firmware Download

If you want to carry out a Firmware download, act as follow: Call up the **Online > Firmware Download** menu.

A warning appears that the communication will be interrupted. This warning must be confirmed by clicking at the **Yes** button.

Figure 43: Security question before Firmware Download

Select the new Firmware and retrieve it with **Download** into the device. The Firmware is now retrieved.

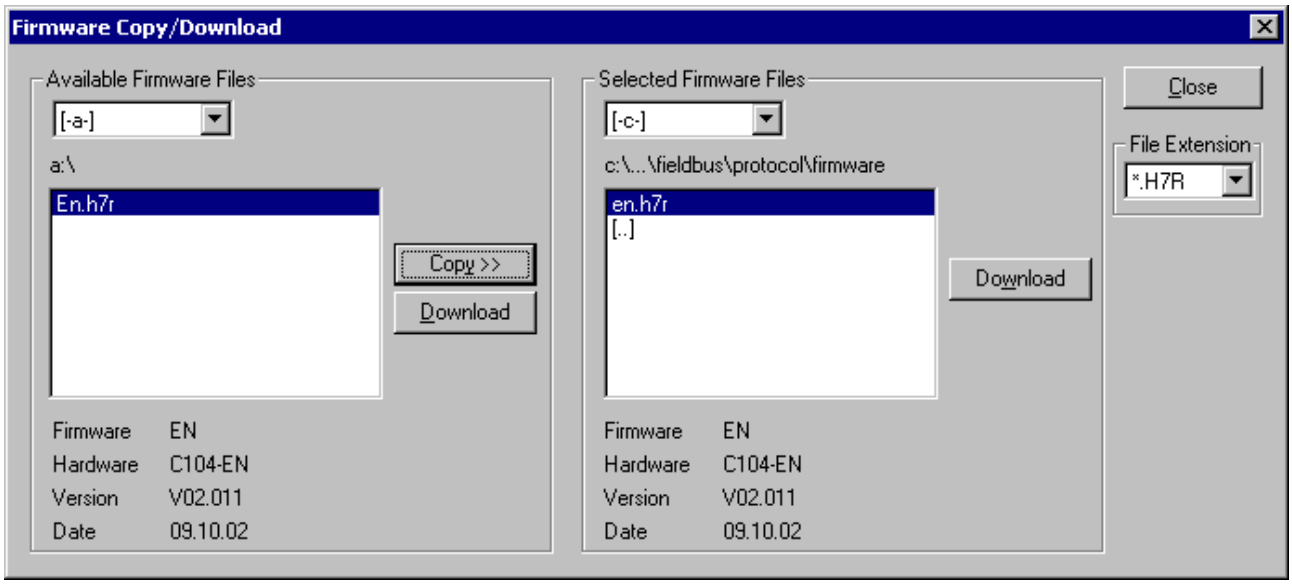

Figure 44: Online > Firmware Download

6.2.3 Firmware / Reset

First the requested device must be chosen with a left mouse click on the symbol of the device. Then the **Online > Firmware / Reset** menu must be called up and the name and version of the Firmware are displayed.

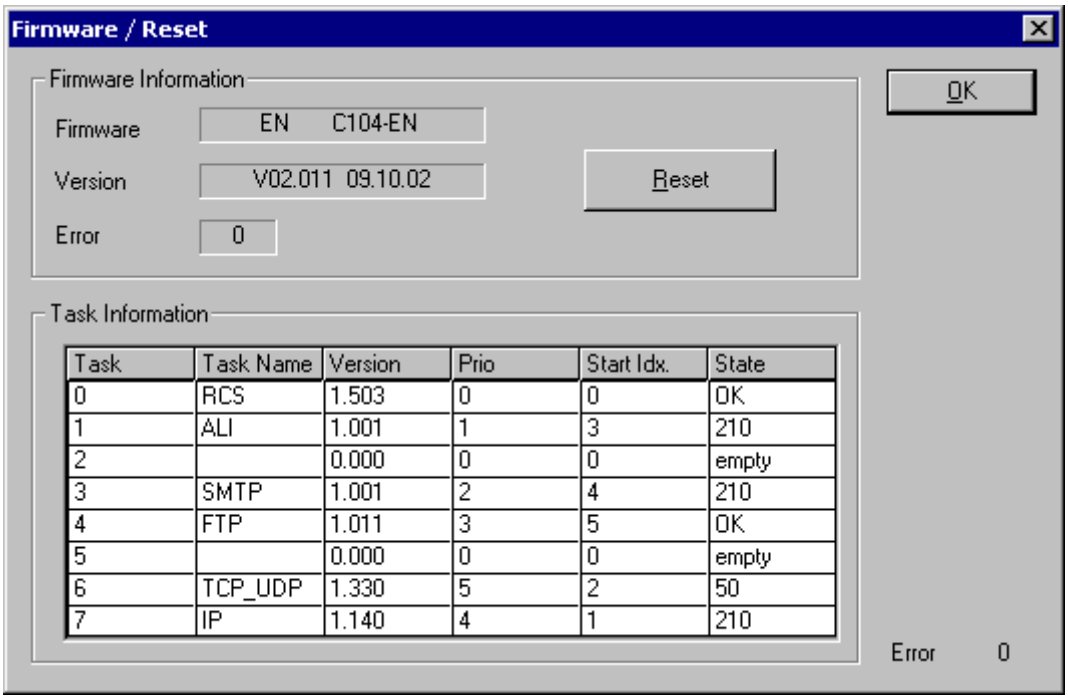

Figure 45: Online > Firmware / Reset

The device can be resetted with the **Reset** button.

6.2.4 Device Info

First the requested device must be chosen with a left mouse click on the symbol of the device. Then select the **Online > Device Info** menu in order to obtain further information on the selected device.

The Manufacturer date, device number, available driver activations and serial number of the device is retrieved and shown.

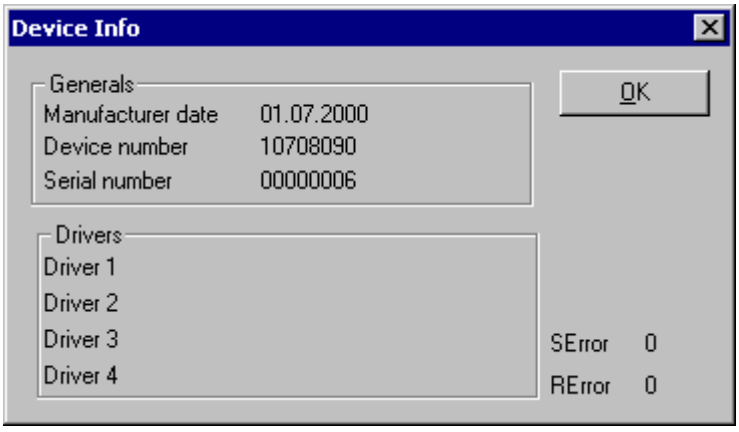

Figure 46: Online > Device Info

6.3 Start/Stop Communication

First the requested device must be chosen with a left mouse click on the symbol of the device. The communication can be manually started or stopped. In order to do this select the **Online > Communication start** or **Online > Communication stop** menu.

Note: The function Start/Stop Communication is firmware depending and is not supported by all firmwares.

6.4 Diagnostic Functions

The following table shows Diagnostic Functions and their usability for Ethernet/Protocol devices.

Diagnostic Function	Using
Extended Device Diagnostic	Statistic information and State information of the Hilscher Ethernet/Protocol device
Message Monitor	Access to mailbox of the device

Table 22: Overview Diagnostic Functions

6.4.1 Extended Device Diagnostic

The extended device diagnostic assists in finding communication and configuration errors. Select the **Online > Extended Device Diagnostic** menu. This menu opens a list of diagnostic structures. These contain online counters, states and parameters.

The counter is resetted after a cold start.

Select the protocol which is used in your configuration in the following table. In the given section you find the Extended Device Diagnostic for the corresponding protocol.

Protocol	Page
Extended Device Diagnostic Ethernet	*76*
Extended Device Diagnostic Open Modbus / TCP	*77*
Extended Device Diagnostic ASCII	*78*
Extended Device Diagnostic 3964R	*78*
Extended Device Diagnostic RK512	*79*
Extended Device Diagnostic Modbus RTU	*79*
Extended Device Diagnostic Modbus Plus	*80*
Extended Device Diagnostic Modnet 1/N	*80*
Extended Device Diagnostic Modnet 1/SFB	*81*

Table 23: Selection of the Protocol for the Extended Device Diagnostic

6.4.1.1 Extended Device Diagnostic Ethernet

If an Ethernet configuration is loaded and the menu **Online > Extended Device Diagnostic** is selected the following window with the diagnostic structures of the Ethernet configuration appears.

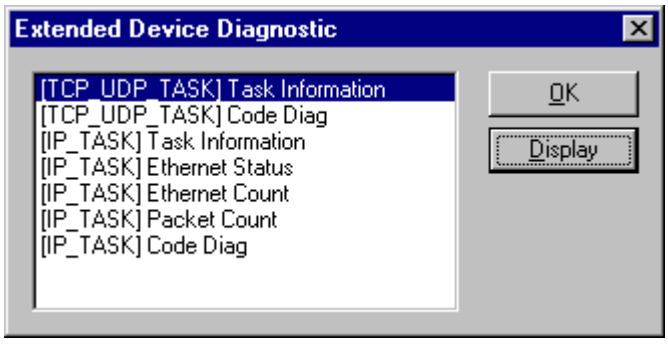

Figure 47: Online > Extended Device Diagnostic Ethernet

Task / Task State	Page
TCP_UDP_TASK Task Information	*108*
TCP_UDP_TASK Code Diag	*109*
IP_TASK Task Information	*110*
IP_TASK Ethernet Status	*120*
IP_TASK Ethernet Count	*112*
IP_TASK Packet Count	*113*
IP_TASK Code Count	*114*

Table 24: Ethernet Task States

6.4.1.2 Extended Device Diagnostic Open Modbus / TCP

If an Open Modbus / TCP configuration is loaded and the menu **Online > Extended Device Diagnostic** is selected the following window with the diagnostic structures of the Open Modbus / TCP configuration appears.

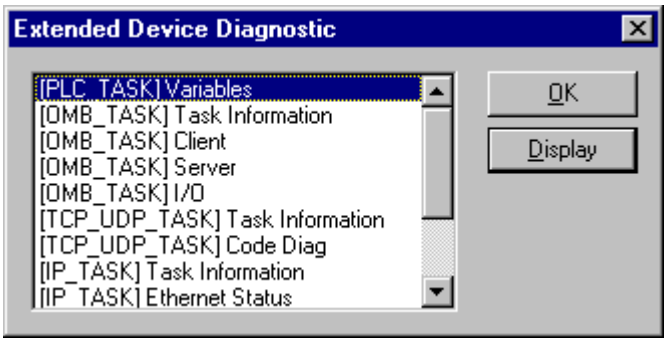

Figure 48: Online > Extended Device Diagnostic Open Modbus / TCP

Task / Task State	Page
Open Modbus / TCP PLC_TASK Variables	*115*
OMB_TASK Task Information	*116*
OMB_TASK Client	*117*
OMB_TASK Server	*118*
OMB_TASK I/O	*119*
TCP_UDP_TASK Task Information	*120*
TCP_UDP_TASK Code Diag	*120*
IP_TASK Task-Information	*120*
IP_TASK Ethernet Status	*120*
IP_TASK Ethernet Count	*120*
IP_TASK Packet Count	*120*
IP_TASK Code Diag	*120*

Table 25: Open Modbus / TCP Task States

6.4.1.3 Extended Device Diagnostic ASCII

If an ASCII configuration is loaded and the menu **Online > Extended Device Diagnostic** is selected the following window with the diagnostic structures of the ASCII configuration appears.

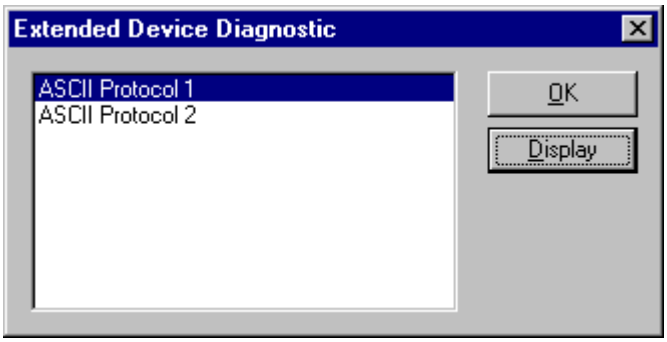

Figure 49: Online > Extended Device Diagnostic ASCII

Task / Task State	Page
ASCII Protocol	*121*

Table 26: ASCII Task states

6.4.1.4 Extended Device Diagnostic 3964R

If a 3964R configuration is loaded and the menu **Online > Extended Device Diagnostic** is selected the following window with the diagnostic structures of the 3964R configuration appears.

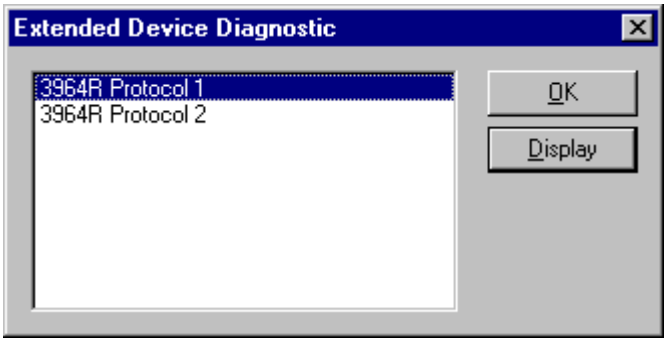

Figure 50: Online > Extended Device Diagnostic 3964R

Task / Task State	Page
3964R Protocol	*122*

Table 27: 3964R Task states

6.4.1.5 Extended Device Diagnostic RK512

If a RK512 configuration is loaded and the menu **Online > Extended Device Diagnostic** is selected the following window with the diagnostic structures of the RK512 configuration appears.

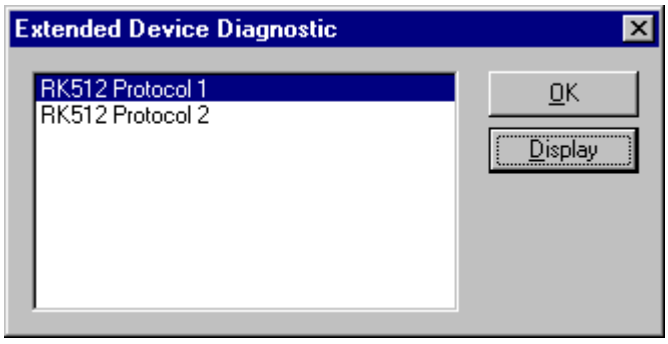

Figure 51: Online > Extended Device Diagnostic RK512

Task / Task State	Page
RK512 Protocol	*123*

Table 28: RK512 Task states

6.4.1.6 Extended Device Diagnostic Modbus RTU

If a Modbus RTU configuration is loaded and the menu **Online > Extended Device Diagnostic** is selected the following window with the diagnostic structures of the Modbus RTU configuration appears.

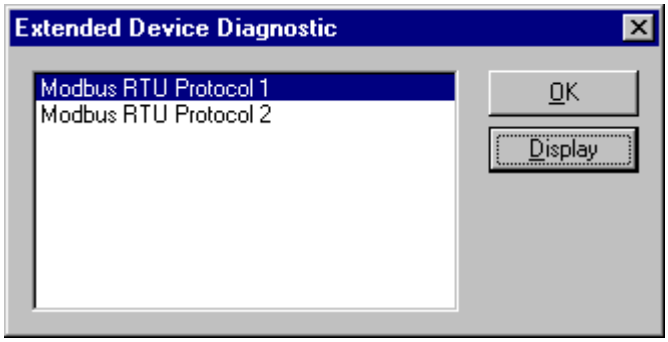

Figure 52: Online > Extended Device Diagnostic Modbus RTU

Task / Task State	Page
Modbus RTU Protocol	*124*

Table 29: Modbus RTU Task states

6.4.1.7 Extended Device Diagnostic Modbus Plus

If a Modbus Plus configuration is loaded and the menu **Online > Extended Device Diagnostic** is selected the following window with the diagnostic structures of the Modbus Plus configuration appears.

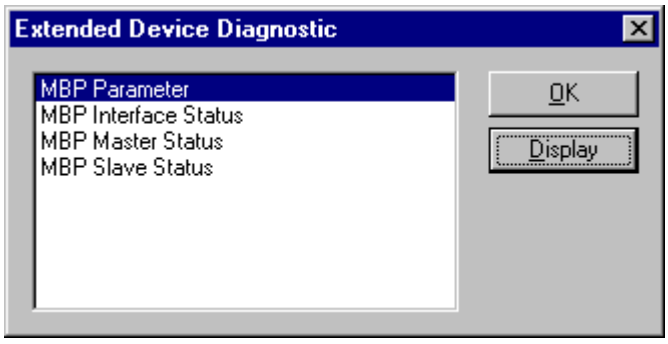

Figure 53: Online > Extended Device Diagnostic Modbus Plus

Task / Task State	Page
Modbus Plus Parameter	*125*
Modbus Plus Interface Status	*126*
Modbus Plus Master Status	*127*
Modbus Plus Slave Status	*128*

Table 30: Modbus Plus Task states

6.4.1.8 Extended Device Diagnostic Modnet 1/N

If a Modnet 1/N configuration is loaded and the menu **Online > Extended Device Diagnostic** is selected the following window with the diagnostic structures of the Modnet 1/N configuration appears.

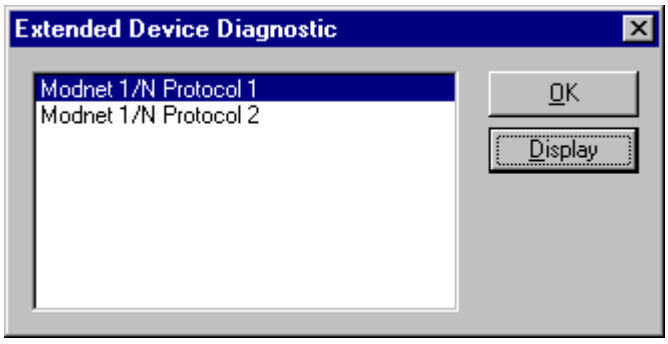

Figure 54: Online > Extended Device Diagnostic Modnet 1/N

Task / Task State	Page
Modnet 1/N Protocol	*129*

Table 31: Modnet 1/N Task states

6.4.1.9 Extended Device Diagnostic Modnet 1/SFB

If a Modnet 1/SFB configuration is loaded and the menu **Online > Extended Device Diagnostic** is selected the following window with the diagnostic structures of the Modnet 1/SFB configuration appears.

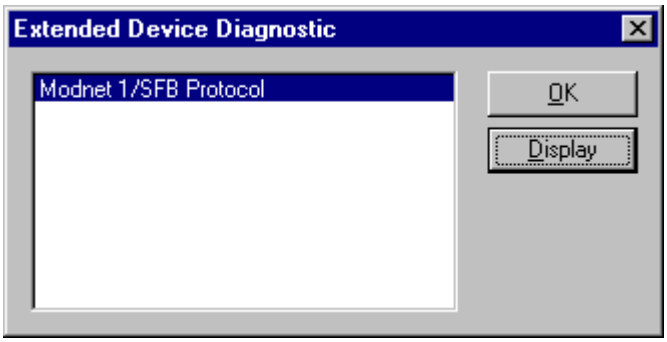

Figure 55: Online > Extended Device Diagnostic Modnet 1/SFB

Task / Task State	Page
Modnet 1/SFB	*130*

Table 32: Modnet 1/SFB Task states

6.5 Message Monitor

The Message Monitor permits access to the Mailbox of the device. The usage of the Message Monitor assumes advanced knowledge on the part of the user.

Select the **Online > Message Monitor** menu.

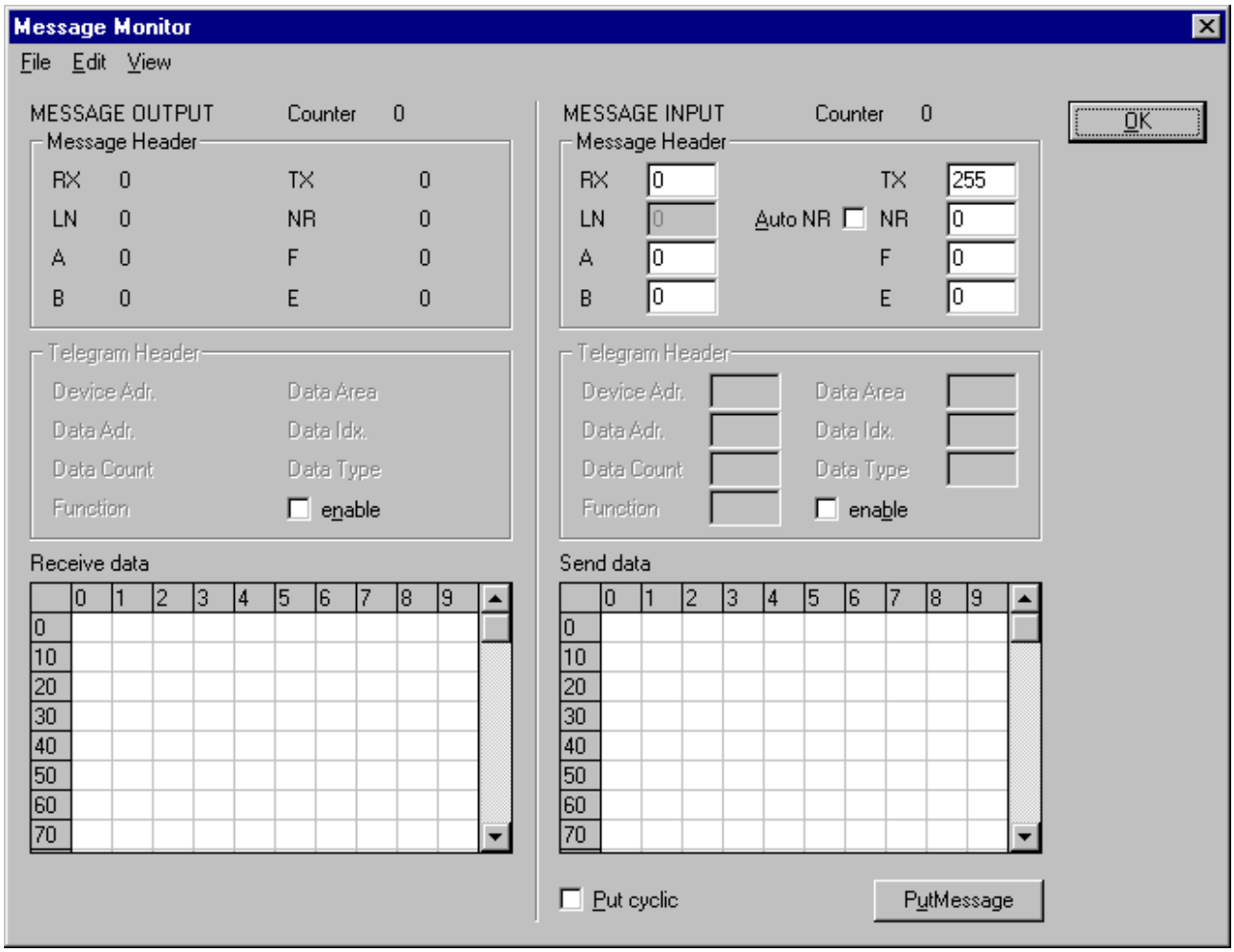

Figure 56: Online > Message Monitor

A Message can be saved and retrieved and has the file suffix *.MSG.

File > New: closes the window

File > Open: opens a Message (Message can be retrieved)

File > Save or **File > Save As**: saves a Message

File > Exit: ends the Message Monitor and returns to the SyCon

Edit > Create answer: creates an answer Message

Edit > Reset counter: resets the Message counter

View > Receive Data Overview: all received data are shown

View > Send Data Overview: all the send data are shown

View > Receive Error Count: the numbers of the received errors are shown

View > Decimal/Hexadecimal: switch the display format

It is recommend to create a sub-directory Msg and to store the messages in it.

Figure 57: Save a Message

7 File, Print, Export, Edit and View

7.1 File

7.1.1 Open

An existing project can be opened with **File > open**.

7.1.2 Save and Save As

When the file name is known, the configuration can be saved under the **File > Save** menu, otherwise the **File > Save As** menu must be selected.

7.1.3 Close

The current project can be closed with **File > Close**.

7.2 Print

After the current printer has been selected in the **File > Printer Setup** menu, the configuration can be printed out under the **File > Print** menu. For a page view, select the **File > Page View** menu.

7.3 Export Functions

7.3.1 DBM Export

Select the **File > Export > DBM** menu in order to save the previously saved project file (*.IB Microsoft Access Format) in a DBM file (Hilscher binary format). This DBM file can be retrieved in the DOS Compro program. The configuration is stored in the Project directory in the path of the SyCon Installation with the extension *.DBM.

Attention: The file name may be long max. 8 characters.

7.4 Edit

7.4.1 Delete

To delete a device you have to have to select the menu **Edit > Delete**. Before SyCon deletes the device a security question appears. If you really want to delete this device you have to confirm this question with **Yes**, and the device will be deleted.

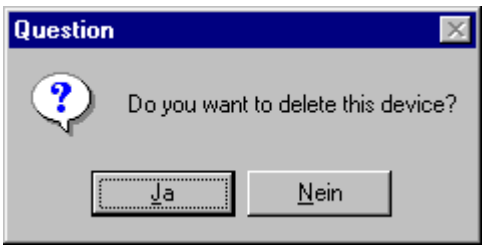

Figure 58: Security question delete device

Note: When you delete a device the settings and the configuration of this device get lost.

7.5 View

7.5.1 Logical Network View

In the menu **View > Logical Network View** the user can activate or deactivate the network view by selecting its (with hook) or by not selecting it (without hook).

The network view is used for example for the Start Options.

7.5.2 Toolbars

In the menu **View > Toolbars** the user has the possibility to activate or deactivate the Toolbars **Standard** and **Fieldbus**. If this function is deactivated the toolbars are not shown.

7.5.3 Status Bar

In the menu **View > Status Bar** this bar can be activated (with hook) or deactivated (without hook).

8 Error Numbers

8.1 CIF Device Driver (Dual-port memory) Error Numbers (-1 .. -49)

This is the list of error numbers of dual-port memory access using the CIF Device Driver.

Error Number	Description
-1	Driver: Board not initialized The communication board is not initialized by the driver. No or wrong configuration found for the given board, check the driver configuration. Driver function used without calling DevOpenDriver() first.
-2	Driver: Error in internal ‘Init state‘
-3	Driver: Error in internal ‘Read state‘
-4	Driver: Command on this channel is active
-5	Driver: Unknown parameter in function occurred
-6	Driver: Version is incompatible The device driver version does not correspond to the driver DLL version. From version V1.200 the internal command structure between DLL and driver has changed. Make sure to use the same version of the device driver and the driver DLL.
-10	Device: Dual port memory RAM not accessible (board not found) Dual-ported RAM (DPM) not accessible / no hardware found. This error occurs, when the driver is not able to read or write to the Dual-port memory. Check the BIOS setting of the PC Memory address conflict with other PC components. Try another memory address, check the driver configuration for this board, and check the jumper setting of the board.
-11	Device: Not ready (RDY flag=Ready flag failed) Board is not ready. This could be a hardware malfunction or another program writes inadmissible to the dual-port memory.
-12	Device: Not running (RUN flag=Running flag failed) The board is ready but not all tasks are running, because of an initialization error. No data base is loaded into the device or a wrong parameter can causes that a task can't initialize.
-13	Device: Watch dog test failed
-14	Device: Signals wrong Operating System version No license code found on the communication board. Device has no license for the used operating system or customer software. No firmware or no data base on the device is loaded.

Table 33: CIF Device Driver Error Numbers (-1..-14)

Error Number	Description
-15	Device: Error in dual port memory flags
-16	Device: Send mailbox is full
-17	Device: Function PutMessage timeout No message could be send during the timeout period given in the DevPutMessage() function. If you use an interrupt, then check the interrupt on the device and in driver setup. These settings have to be the same! Is an interrupt on the board set? Is the right interrupt set? The interrupt could already be used by another PC component, also if the operating system reports it as unused. If you use polling mode, then make sure that no interrupt is set on the board and that polling is set in the driver setup. These settings have to be the same! Device internal segment buffer full and therefore PutMessage() function is not possible, because all segments on the device are in use. This error occurs, when only PutMessage() is used but not GetMessage(). HOST flag is not set for the device. No messages are taken by the device. Use DevSetHostState() to signal a board an application is available.
-18	Device: Function GetMessage timeout No message received during the timeout period given in the DevGetMessage() function. If you use an interrupt, then check the interrupt on the device and in driver setup. These settings have to be the same! Is an interrupt on the board set? Is the right interrupt set? The interrupt could already be used by another PC component, also if the operating system reports it as unused. If you use polling mode, then make sure that no interrupt is set on the board and that polling is set in the driver setup. These settings have to be the same! The used protocol on the device needs longer than the timeout period given in the DevGetMessage() function.
-19	Device: No message available

Table 34: CIF Device Driver Error Numbers (-15..-19)

Error Number	Description
-20	Device: Reset command timeout The board is ready but not all tasks are running, because of an initialization error. No data base is loaded into the device or a wrong parameter can causes that a task can't initialize. The device needs longer than the timeout period given in the DevReset() function. Using device interrupts. The timeout period can differ between fieldbus protocols. If you use an interrupt, then check the interrupt on the device and in driver setup. These settings have to be the same! Is an interrupt on the board set? Is the right interrupt set? The interrupt could already be used by another PC component, also if the operating system reports it as unused. If you use polling mode, then make sure that no interrupt is set on the board and that polling is set in the driver setup. These settings have to be the same!
-21	Device: COM flag not set The device can not reach communication state. Device not connected to the fieldbus. No station found on the fieldbus. Wrong configuration on the device.
-22	Device: IO data exchange failed
-23	Device: IO data exchange timeout The device needs longer than the timeout period given in the DevExchangeIO() function. If you use an interrupt, then check the interrupt on the device and in driver setup. These settings have to be the same! Is an interrupt on the board set? Is the right interrupt set? The interrupt could already be used by another PC component, also if the operating system reports it as unused. If you use polling mode, then make sure that no interrupt is set on the board and that polling is set in the driver setup. These settings have to be the same!
-24	Device: IO data mode unknown
-25	Device: Function call failed
-26	Device: Dual-port memory size differs from configuration
-27	Device: State mode unknown

Table 35: CIF Device Driver Error Numbers (-20..-27)

Error Number	Description
-30	User: Driver not opened (device driver not loaded) The device driver could not be opened. Device driver not installed. Wrong parameters in the driver configuration. If the driver finds invalid parameters for a communication board and no other boards with valid parameters are available, the driver will not be loaded.
-31	User: Can't connect with device board
-32	User: Board not initialized (DevInitBoard not called)
-33	User: IOCTRL function failed A driver function could not be called. This is an internal error between the device driver and the DLL. Make sure to use a device driver and a DLL with the same version. An incompatible old driver DLL is used.
-34	User: Parameter DeviceNumber invalid
-35	User: Parameter InfoArea unknown
-36	User: Parameter Number invalid
-37	User: Parameter Mode invalid
-38	User: NULL pointer assignment
-39	User: Messagebuffer too short
-40	User: Size parameter invalid
-42	User: Size parameter with zero length
-43	User: Size parameter too long
-44	User: Device address null pointer
-45	User: Pointer to buffer is a null pointer
-46	User: SendSize parameter too long
-47	User: ReceiveSize parameter too long
-48	User: Pointer to send buffer is a null pointer
-49	User: Pointer to receive buffer is a null pointer

Table 36:CIF Device Driver Error Numbers (-30..-49)

Error Number	Description
1000	If the operating system of the device reports an initialization error, then a value of 1000 will be add to the error number and shown to the user

Table 37: CIF Device Driver Error Numbers (1000)

8.2 CIF Serial Driver Error Numbers (-20 .. -71)

This is the list of error numbers using the serial driver.

Error Number	Description
-20	Driver: No COM port found or COM port already in use.
-21	Driver: COM port already opened
-22	Driver: Function call into driver has failed
-23	Driver: Internal driver error
-24	Driver: Could not create read thread
-25	Driver: Could not create read event
-26	Driver: Could not create write event
-27	Driver: Could not create timer event
-28	Driver: Error by writing data
-29	Driver: Wrong COM state
-30	Driver: COM state error is set
-31	Driver: COM buffer setup failed
-32	Driver: COM set timeout failed
-33	Driver: Receive buffer overrun
-34	Driver: Receive buffer full
-35	Driver: Send busy
-36	Driver: Error during close driver
-40	User: COM port not opened
-41	User: Invalid handle value
-42	User: Invalid COM number
-43	User: Size parameter invalid
-44	User: Size parameter zero
-45	User: Buffer pointer is NULL
-46	User: Buffer too short
-47	User: Setup error

Table 38: CIF Serial Driver Error Numbers (-20..-47)

Error Number	Description
-50	User: Send message, timeout error
-51	User: Could not send a message Cable not connected. Wrong cable. Device does not respond.
-52	User: Send message, no device connected
-53	User: Error by send message, message receiving
-54	User: Telegram collision
-55	User: Telegram, no acknowledgement received
-56	User: Telegram, noise
-57	User: Telegram, data overrun
-58	User: Telegram, parity error
-59	User: Telegram, framing error
-60	User: Telegram, unknown error
-70	User: Timeout by receive a message
-71	User: No message received

Table 39: CIF Serial Driver Error Numbers (-20..-47)

8.3 CIF TCP/IP Driver Error Numbers

This is the list of error numbers using the CIF TCP/IP Driver.

8.3.1 Standard Win32 Socket API Errors

Error Number	Description
10013	Permission denied
10024	Too many open sockets.
10048	Address already in use
10049	Cannot assign requested address.
10050	Network is down
10051	Network is unreachable
10052	Network dropped connection on reset
10053	Software caused connection abort. An established connection was aborted by the software in your host machine, possibly due to a data transmission time-out or protocol error.
10054	Connection reset by peer
10055	No buffer space available
10056	Socket is already connected
10057	Socket is not connected.
10058	Cannot send after socket shutdown
10060	Connection timed out
10061	Connection refused
10065	No route to host
10092	Winsock.dll version out of range

Table 40: CIF TCP/IP Driver Error Numbers - Standard Win32 Socket API errors

8.3.2 Specific NetIdent Errors

Error Number	Description
0x8004c701	Unknown Device Error
0x8004c702	Request Pending
0x8004c703	Set IP time exceeded
0x8004c704	IP address invalid
0x8004c705	Returned IP address invalid
0x8004c706	Answer from wrong device
0x8004c707	Wrong OP code received
0x8004c708	NetIdent Timeout

Table 41: CIF TCP/IP Driver Error Numbers - Specific NetIdent Errors

8.4 RCS Error Numbers (4 .. 93)

This is the list of error numbers returned by the RCS (Realtime Communication System), that is the operating system of Hilscher devices. The error number is returned in an answer message. Command messages and answer messages are used to communicate between the application (e.g. the System Configurator) and the Hilscher device. An example of this communication is the download of a configuration.

Error Number	Description
4	Task does not exist
5	Task is not initialized
6	The MCL is locked
7	The MCL rejects a send command because of an error
20	The user will download a database into the device that is not valid for this device type.
21	Data base segment not configured or not existent
22	Number for message wrong during download
23	Received number of data during download does not match to that in the command message
24	Sequence identifier wrong during download
25	Checksum after download and checksum in command message do not match
26	Write/Read access of data base segment
27	Download/Upload or erase of configured data base type is not allowed
28	The state of the data base segment indicated an error. Upload not possible
29	The access to the data base segment needs the bootstraploader. The bootstraploader is not present
30	Trace buffer overflow
31	Entry into trace buffer too long
37	No or wrong license. The OEM license of the System Configurator allows only communication to devices that have the same license inside
38	The data base created by the System Configurator and the data base expected by the firmware is not compatible
39	DBM module missing

Table 42: RCS error numbers (answer message) (4..39)

Error Number	Description
40	No command free
41	Command unknown
42	Command mode unknown
43	Wrong parameter in the command
44	Message length does not match to the parameters of the command
45	Only a MCL does use this command to the RCS
50	FLASH occupied at the moment
51	Error deleting the FLASH
52	Error writing the FLASH
53	FLASH not configured
54	FLASH timeout error
55	Access protection error while deleting the FLASH
56	FLASH size does not match or not enough FLASH memory
60	Wrong structure type
61	Wrong length of structure
62	Structure does not exist
70	No clock on the device
80	Wrong handle for the table (table does not exist)
81	Data length does not match the structure of this table
82	The data set of this number does not exist
83	This table name does not exist
84	Table full. No more entries allowed
85	Other error from DBM
90	The device info (serial number, device number and date) does already exist
91	License code invalid
92	License code does already exist
93	All memory locations for license codes already in use

Table 43: RCS error numbers (answer message) (40..93)

8.5 Database Access Error Numbers (100 .. 130)

The following table lists the error numbers of the database access errors

Error Number	Description
100	Database already opened
101	Dataset could not be opened
103	Error while opening database occurred
104	No valid path name
105	No connection to data base. Call function DbOpen().
106	Error in parameter
107	Error during opening a table
108	Null pointer occurred
109	Table not opened. Call function OpenTable() first.
110	The first record is reached
111	The last record is reached
112	Unknown type in the record found
113	Data has to be truncated
114	No access driver installed on the system
115	Exception received
116	This table is set to read only
117	There is no data set in the table
118	The requested table could not be edit
119	An operation could not be completed
120	User gives an unexpected length in WritsDs().
121	An assertion failed
122	DLL not found
123	DLL couldn't be freed
124	Specified function not found in the DLL
125	ODBC Function returns an error
126	Count of data bytes in the record exceeds 1938
127	DBM32 DLL is not loaded
128	Field with the given index was not found
129	This table contains no records
130	Invalid character (' ') found in a Table or Column

Table 44: Database Access Error Numbers (100..130)

8.6 Online Data Manager Error Numbers

8.6.1 Online Data Manager Error Numbers (1000 .. 1018)

The following table lists the error numbers of the Online Data Manager.

Error Number	Description
1000	Driver OnlineDataManager not opened
1001	Initialization of the OnlineDataManager has failed
1002	No DriverObject found. OnlineDataManager Sub DLL not found.
1003	No DeviceObject found. Device not found.
1004	Application not found
1010	Application has requested an unknown event
1011	Application has requested an unknown function mode, operating mode. Known function modes, operating modes are Reset, Download, Register Server, Unregister Server.
1012	Application has requested an unknown command
1013	Message Server already exists
1014	Message Server not registered
1015	Device already in use
1016	Device not assigned
1017	Device has changed
1018	Command active

Table 45: Online Data Manager Error numbers (1000..1018)

8.6.2 Message Handler Error Numbers (2010 .. 2027)

The following table lists the error numbers of the Message handler of the Online Data Manager.

Error Number	Description
2010	Message handler: Messagebuffer empty
2011	Message handler: Messagebuffer full
2021	Message handler: Invalid Message ID (msg.nr)
2022	Message handler: No entry
2023	Message handler: Message already active
2024	Message handler: Wrong Application
2025	Message handler: Message Timeout
2026	Message handler: Wait for Delete
2027	Message handler: No cyclic Message

Table 46: Error Numbers of the Message Handler of the Online Data Manager (2010..2027)

8.6.3 Driver Functions Error Numbers (2501 .. 2512)

The following table lists the error numbers of the Driver Functions of the Online Data Manager.

Error Number	Description
2501	OnlineDataManager Sub DLL not found
2502	Function missing
2503	'Read Thread' not created
2504	'Write Thread' not created
2505	'IO Thread' not created
2510	Function failed
2512	Assign reports error. Return neither OK or cancel

Table 47: Error Numbers of the Driver Functions of the Online Data Manager (2501..2512)

8.6.4 Online Data Manager Subfunctions Error Numbers (8001 .. 8035)

The following table lists the error numbers of the Subfunctions of the Online Data Manager.

Error Number	Description
8001	Driver not opened. E.g. CIF Device Driver
8002	Application has requested an unknown event
8003	Application has requested an unknown command
8004	Command has failed
8005	Command active
8006	Device invalid
8010	No device was assigned
8011	Device was already assigned
8020	Driver not connected
8021	Driver already connected
8030	Faulty 'GetState'
8031	Send error (PutMessage returns error)
8032	Send active (PutMessage active)
8033	Receive error (GetMessage returns error)
8034	Receive active (GetMessage active)
8035	IO Error (ExchangeIO returns error)

Table 48: Sub function Error Numbers of the Driver Functions of the Online Data Manager (8001..8035)

8.7 Data Base Functions Error Numbers (4000 .. 4199)

The following table lists the error numbers of data base errors.

Error Number	Description
4000	File does not exist
4001	Success in compromising
4002	Dataset does not exist
4003	Last respectively first entry reached
4004	Not enough memory
4005	File directory full
4006	Max number of entries reached
4007	No writing to this table possible, because the table is located in the FLASH
4008	Table name does already exist
4009	File name does not exist
4010	Free RAM length from RCS_CNF.P86 is smaller than E_F_INDEX * 2
4011	Parameter ‘next’ wrong
4012	Not enough free space to copy data set
4013	Set is deleted
4014	Value for Index is wrong
4015	Access not allowed
4016	open_file used before init_file
4017	Drive is not ready
4018	Not enough drive memory
4019	File name or path does not exist
4020	Cannot create path
4021	Wrong path
4022	Wrong flag
4023	The delete path is the root path
4024	Path file exists
4025	Write error during write a file
4026	Error during create a file
4027	Error during close a file
4028	No DBM file
4029	Length of the read data is unequal of the file length

Table 49: Error numbers of converting functions (4000..4029)

Error Number	Description
4030	Path too long
4031	Directory changed
4032	Directory created
4034	Length of converting stream is 0
4035	Non equal data set found
4036	Non equal data set found
4037	Non equal data set found
4038	Data set has length 0
4039	The function DbmInit has assigned a Zero pointer during RCS initialization
4040	Printer not ready
4041	The data base is used from an other function
4042	New length of data base is smaller than used
4043	Unknown access mode
4044	Old data base has to be converted
4045	Error while converting. Function not known
4046	Unknown type in set 0 found
4047	No float function available
4048	Function not in RCS module
4049	Check failed
4050	Checksum check failed
4051	More segments are existing in file, than in the structure FILE_INFO_T in wMaxEintraege
4052	SegLen in structure FILE_INFO_T is smaller then the length in the file. Return of function dbm_restore_data
4053	The header file holds an other information for a length than in the segment itself
4054	Not enough memory for allocation on the PC
4055	No index for file handle in structure FLASH_DIR of RCS found
4056	
4057	File type 2 can not be printed because of too many definitions
4058	The definitions need too many lines to display them, than in the program available
4059	An unknown format for the parameter. Valid is U, H, or S
4060	Unknown parameter type

Table 50: Error numbers of converting functions (4030..4060)

Error Number	Description
4061	The data base was transmitted into the FLASH
4062	Set 0 contains no structure definition
4063	Set 0 can not be deleted
4064	Error during execution of a ODBC data base access
4065	Initializing of DBM through RCS had no success
4066	Passed data length incorrect
4067	Sorting function not linked
4068	Error in function parameter
4069	Error from ODBC table
4070	No free handle available. Too many data base links are already opened
4071	Unknown data type found in the table
4072	Structure of table GLOBAL not correct or no such table existing
4073	No name of an ACCESS data base
4074	Download window can’t be created
4075	Download not fully performable

Table 51: Error numbers of converting functions (4061..4075)

Error Number	Description
4082	More than 32 tables should be created
4083	No entry in element szSourceFile
4084	ODBC connection initialization not possible. This could happen when in file ODBCINST.INI in section [Microsoft Access Driver (*.mdb)] is no valid path to ODBCJT16/32.DLL.
4085	Error in structure in the ACCESS data base that is in DBM format
4086	Error in structure in the ACCESS data base that is in DBM format
4087	No data in a ODBC table
4088	No entry
4089	ODBC set length not valid
4090	Not enough data sets in ODBC table
4091	Table CreateTab not found
4092	Error in structure of table CreateTab
4093	No entry in element szSourceTable
4094	No entry in element szDestTable
4095	Entry in iSourceType of table CreateTab is wrong
4096	Entry in iTranslate of table CreateTab is wrong
4097	Function SQLAllocStmt reports an error
4098	ODBC source table not found
4099	ODBC data truncated
4100	Download timeout
4101	Library load error
4102	Library function error
4103	Error in description 'toggle'
4104	Error in description 'KB'
4105	Column does not exists
4106	ODBC structure different
4107	ODBC address error
4108	No CRC sum exists (table GLOBAL exists or old)
4109	Table GLOBAL is old
4110	Calculated CRC different to CRC in table GLOBAL
4199	Programming error

Table 52: Error numbers of converting functions (4082..4199)

8.8 Converting Functions Error Numbers (5001 .. 5008)

The following table lists the error numbers of converting functions.

Error Number	Description
5000	Function PackLongToByteShort: Not enough space in pvD (Number of elements greater than reserved memory)
5001	Function PackLongToByteShort: Not enough space in pvD. Detected during converting of pvS
5002	Function PackLongToByteShort: Not enough space in pvD
5003	Function StringToByte: Not enough space in pvD
5004	Function IntToByte: Not enough space in pvD
5005	Function LongToShort: Not enough space in pvD
5006	Function PackStringDumpToByteArray: Not enough space in pvD
5007	Function PackStringBumpToByteArray: A character was found, which is not convertible into a HEX value
5008	Function PackStringDumpToByteArray: Number of character odd
5009	Function PackStringDumpToByteArray: Not enough space in pvD
5010	Function PackStringDumpToByteArray: The current data set needs to be appended the previous one
5011	Function PackStringDumpToByteArray: No corresponding function to the given number exist
5012	Converting error

Table 53: Error Numbers of data base functions (5000 .. 5012)

9 Appendix

9.1 Extended Device Diagnostic

The extended device diagnostic assists in finding communication and configuration errors. Select the **Online > Extended Device Diagnostic** menu. This menu opens a list of diagnostic structures. These contain online counters, states and parameters.

Select the protocol which is used in your configuration in the following table. In the given section you find the Extended Device Diagnostic for the corresponding protocol.

Protocol	Page
Extended Device Diagnostic Ethernet	*108*
Extended Device Diagnostic Open Modbus / TCP	*115*
Extended Device Diagnostic ASCII	*121*
Extended Device Diagnostic 3964R	*122*
Extended Device Diagnostic RK512	*123*
Extended Device Diagnostic Modbus RTU	*124*
Extended Device Diagnostic Modbus Plus	*125*
Extended Device Diagnostic Modnet 1/N	*129*
Extended Device Diagnostic Modnet 1/SFB	*130*

Table 54: Selection of the protocol for the Extended Device Diagnostic

9.1.1 Extended Device Diagnostic Ethernet

In this section the task states for Ethernet devices are described.

9.1.1.1 TCP_UDP_TASK Task Information

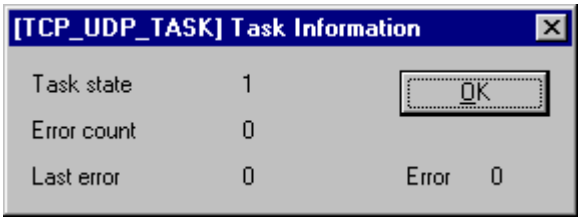

Figure 59: TCP_UDP_TASK Task Information

Variable	Meaning
Task state	State of the Task: 0 = Task not initialized 1 = Task is running 2 = Task initialized 3 = reports an error by initialization
Error count	Number of appeared errors
Last error	Last appeared error (Description see belonging manual)

Table 55: TCP_UDP_TASK Task Information

9.1.1.2 TCP_UDP_TASK Code Diag

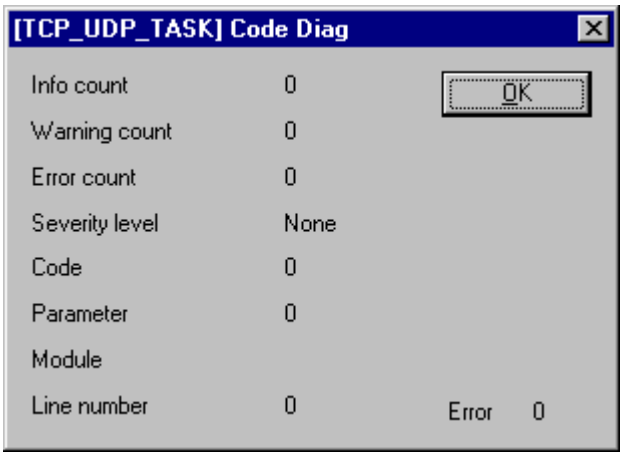

Figure 60: TCP_UDP_TASK Code Diag

Variable	Meaning
Info count	Counter for information reports
Warning count	Counter for warning reports
Error count	Counter for errors
Severity level	Level of the last appeared error
Code	Code of the last appeared error
Parameter	Additional information to the error
Module	Software-module
Line number	Line number inside the software-module

Table 56: TCP_UDP_TASK Code Diag

9.1.1.3 IP_TASK Task Information

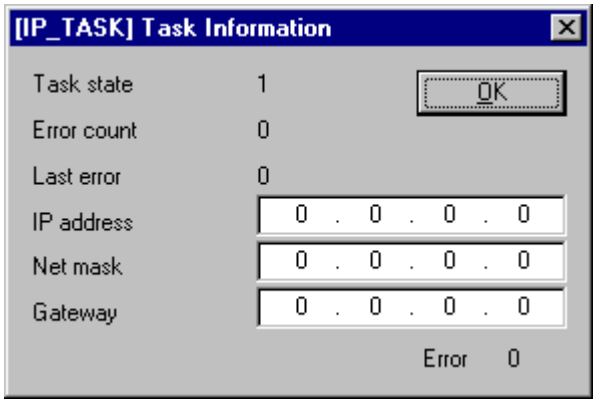

Figure 61: IP_TASK Task Information

Variable	Meaning
Task state	State of the Task: 1 = Task is running 2 = initialization is running 3 = initialization has failed
Error count	Counter for appeared errors
Last error	Last appeared error
IP address	IP-Address of the device
Net mask	Net mask of the device
Gateway	Gateway of the device

Table 57: IP_TASK Task Information

9.1.1.4 IP_TASK Ethernet Status

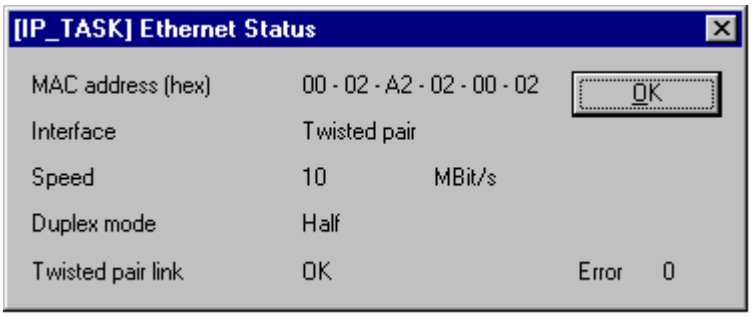

Figure 62: IP_TASK Ethernet Status

Variable	Meaning
MAC address (hex)	MAC address of the device
Interface	Actual known Ethernet interface
Speed	Transmission rate
Duplex mode	Shows the actual Duplex mode: Half-/Full duplex
Twisted pair link	State of the Twisted Pair connection

Table 58: IP_TASK Ethernet Status

9.1.1.5 IP_TASK Ethernet Count

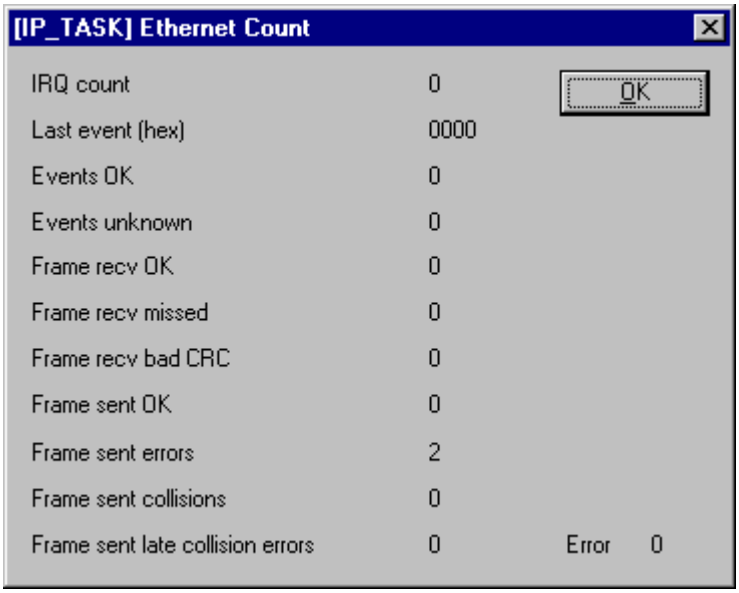

Figure 63: IP_TASK Ethernet Count

Variable	Meaning
IRQ count	Counter for interrupts of the Ethernet controller
Last event (hex)	Last appeared interrupt type
Events OK	Counter for known interrupt types
Events unknown	Counter for unknown interrupt types
Frame recv OK	Counter for received Ethernet frames
Frame recv missed	Counter for missed Ethernet frames
Frame recv bad CRC	Counter for Ethernet frames with CRC errors
Frame sent OK	Counter for sent Ethernet frames
Frame sent errors	Counter for send errors
Frame sent collisions	Counter for sending collisions
Frame sent late collision errors	Counter for late sending collisions

Table 59: IP_TASK Ethernet Count

9.1.1.6 IP_TASK Packet Count

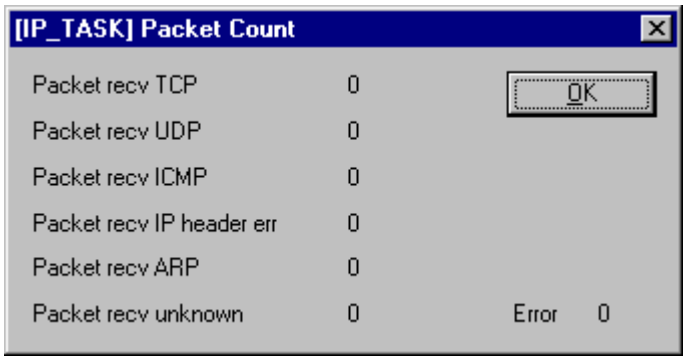

Figure 64: IP_TASK Packet Count

Variable	Meaning
Packet recv TCP	Counter for received TCP packets
Packet recv UDP	Counter for received UDP packets
Packet recv ICMP	Counter for received ICMP packets
Packet recv IP header err	Counter for received IP packets with errors
Packet recv ARP	Counter for received ARP packets
Packet recv unknown	Counter for received packets of an unknown type

Table 60: IP_TASK Packet Count

9.1.1.7 IP_TASK Code Count

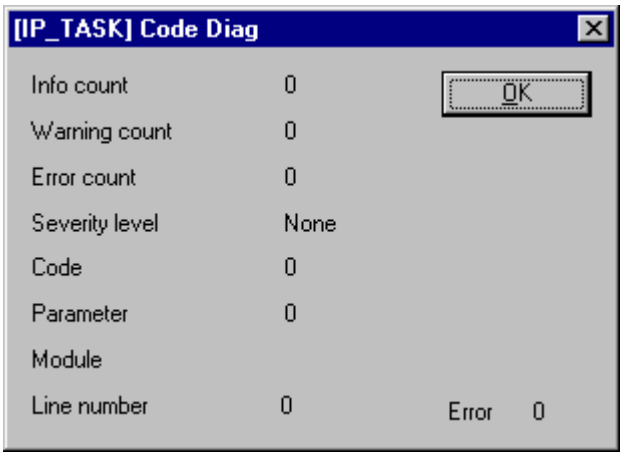

Figure 65: IP_TASK Code Count

Variable	Meaning
Info count	Counter for information reports
Warning count	Counter for warning reports
Error count	Counter for errors
Severity level	Level of the last appeared error
Code	Code of the last appeared error
Parameter	Additional information to the error
Module	Software-module
Line number	Line number inside the software-module

Table 61: IP_TASK Code Count

9.1.2 Extended Device Diagnostic Open Modbus / TCP

In this section the task states for Open Modbus / TCP devices are described.

9.1.2.1 Open Modbus / TCP PLC_TASK Variables

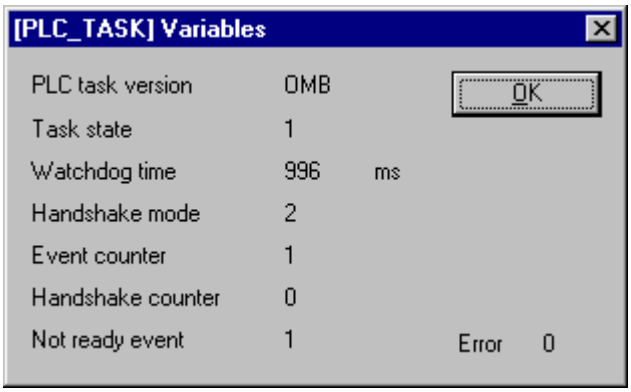

Figure 66: Open Modbus / TCP PLC_TASK Variables

Variable	Meaning
PLC task version	Version of the PLC task
Task state	State of the Task: 0 = The task is not initialized 1 = The task is initialized and running 2 = The task is actually initializing 3 = Initialization error 4 = Task is initialized and waits for TCP Task
Watchdog time	Watchdog time in milliseconds
Handshake mode	The following PCL modes are supported by the device: 1 = buffered / device controlled 2 = inconsistent / uncontrolled 3 = buffered / host controlled
Event counter	Counter for received events
Handshake counter	Counter for Handshakes
Not ready event	Number of Not ready events

Table 62: Open Modbus / TCP PLC_TASK Variables

9.1.2.2 OMB_TASK Task Information

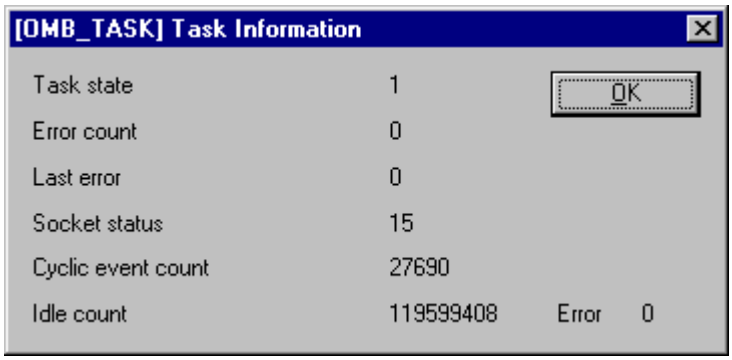

Figure 67: OMB_TASK Task Information

Variable	Meaning
Task state	State of the Task: 0 = The task is not initialized 1 = The task is initialized and running 2 = The task is actually initializing 3 = Initialization error 4 = Task is initialized and waits for TCP Task
Error count	Number of occurred errors
Last error	Last occurred error
Socket status	Information about the actual used TCP socket: 1 = Socket 0 (Connection 0 Close/Open, Bit 0) 2 = Socket 1 (Connection 1 Close/Open, Bit 1) 4 = Socket 2 (Connection 2 Close/Open, Bit 4) 8 = Socket 3 (Connection 3 Close/Open, Bit 8) 16 = Socket 4 (Connection 4 Close/Open, Bit 16)
Cyclic event count	Counter for cyclic events
Idle count	Idle count

Table 63: OMB_TASK Task Information

9.1.2.3 OMB_TASK Client

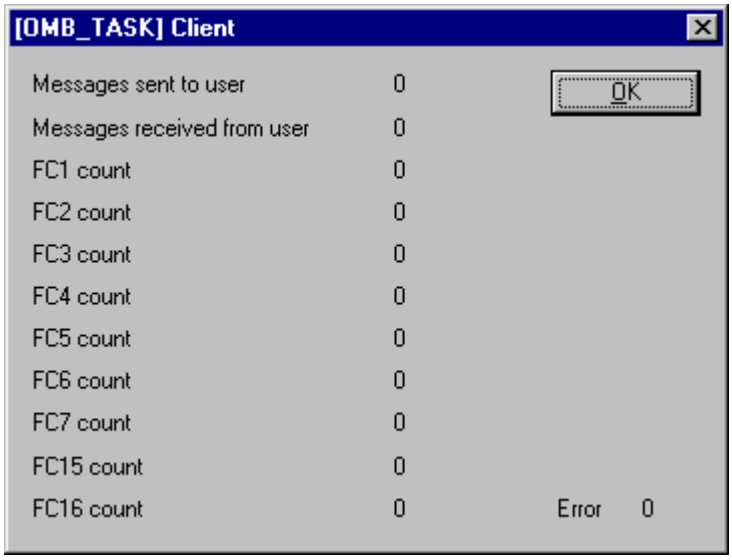

Figure 68: OMB_TASK Client

Variable	Meaning
Messages sent to user	Number of messages sent to the user
Messages received from user	Number of messages received from the user
FC1 count	Number of Function Calls for function code 1
FC2 count	Number of Function Calls for function code 2
...	...
FC16 count	Number of Function Calls for function code 16

Table 64: OMB_TASK Client

9.1.2.4 OMB_TASK Server

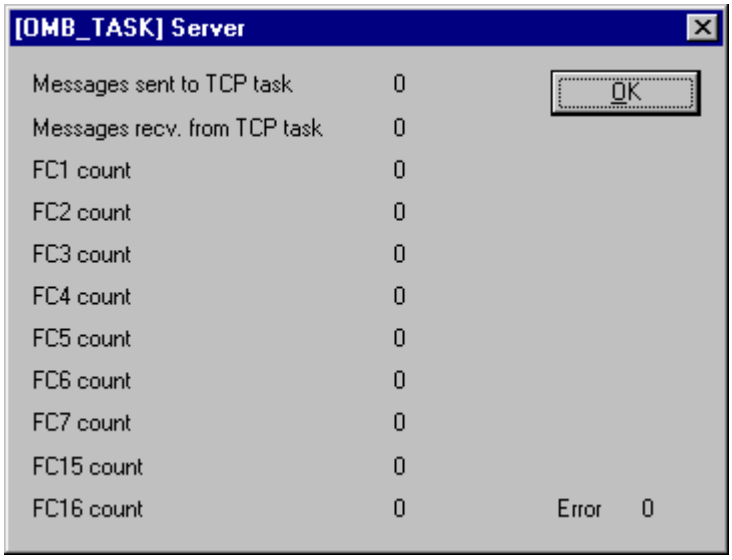

Figure 69: OMB_TASK Server

Variable	Meaning
Messages sent to TCP task	Number of messages which were sent to the TCP task
Messages recv. from TCP task	Number of messages which were received from the TCP task
FC1 count	Number of Function Calls for function code 1
FC2 count	Number of Function Calls for function code 2
...	...
FC16 count	Number of Function Calls for function code 16

Table 65: OMB_TASK Server

9.1.2.5 OMB_TASK I/O

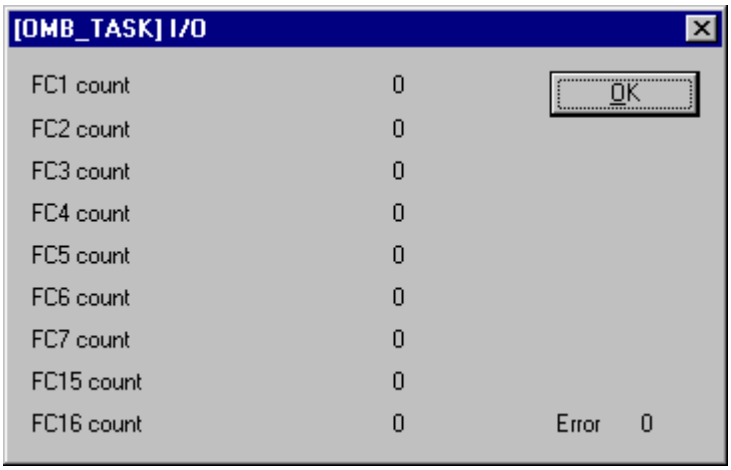

Figure 70: OMB_TASK I/O

Variable	Meaning
FC1 count	Number of Function Calls for function code 1
FC2 count	Number of Function Calls for function code 2
...	...
FC16 count	Number of Function Calls for function code 16

Table 66: OMB_TASK I/O

9.1.2.6 TCP_UDP_TASK Task Information

See section *TCP_UDP_TASK Task Information* at page *108* in the chapter about the extended device diagnostic Ethernet.

9.1.2.7 TCP_UDP_TASK Code Diag

See section *TCP_UDP_TASK Code Diag* at page *109* in the chapter about the extended device diagnostic Ethernet.

9.1.2.8 IP_TASK Task-Information

See section *IP_TASK Task Information* at page *110* in the chapter about the extended device diagnostic Ethernet.

9.1.2.9 IP_TASK Ethernet Status

See section *IP_TASK Ethernet Status* at page *111* in the chapter about the extended device diagnostic Ethernet.

9.1.2.10 IP_TASK Ethernet Count

See section *IP_TASK Ethernet Count* at page *112* in the chapter about the extended device diagnostic Ethernet.

9.1.2.11 IP_TASK Packet Count

See section *IP_TASK Packet Count* at page *113* in the chapter about the extended device diagnostic Ethernet.

9.1.2.12 IP_TASK Code Diag

See section *IP_TASK Code Count* at page *114* in the chapter about the extended device diagnostic Ethernet.

9.1.3 Extended Device Diagnostic ASCII

In this section the task states for ASCII devices are described.

9.1.3.1 ASCII Protocol

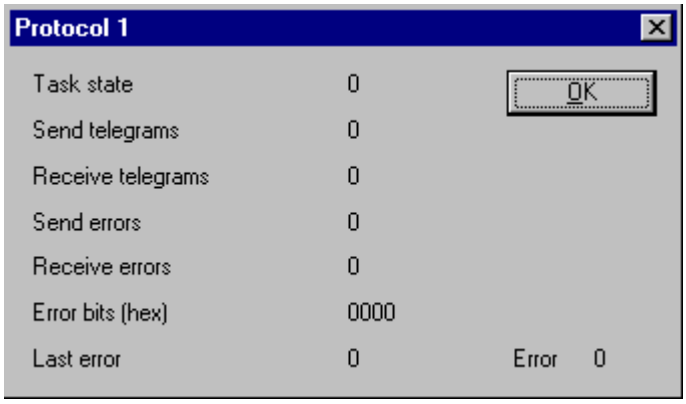

Figure 71: ASCII Protocol

Variable	Meaning
Task state	Actual state of the protocol process: 0 = not initialized 1 = basic state 2 = telegram sequence time expires 3 = sending mode 4 = receiving mode
Send telegrams	Number of faultless transmitted telegrams
Receive telegrams	Number of faultless received telegrams
Send errors	Number of send tasks which got lost because of Syntax or transmission errors
Receive errors	Number of errors which have occurred by receiving data
Error bits (hex)	Assigns every reported error to an error class and shows this by setting a bit. It is only displayed, if the error has executed in spite of repeat a telegram loss
Last error	Number of the last reported error. It is only displayed, if the error has occurred in spite of repeat a telegram loss The description of the error numbers see on the CD in the manual asc_pre.pdf

Table 67: ASCII Protocol

9.1.4 Extended Device Diagnostic 3964R

In this section the task states for 3964R devices are described.

9.1.4.1 3964R Protocol

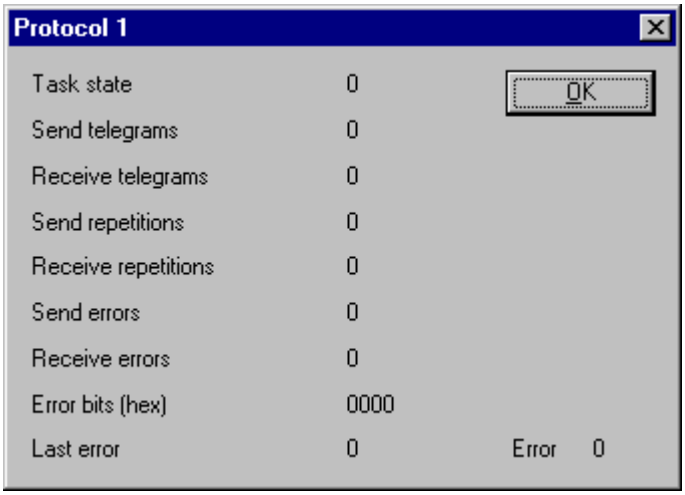

Figure 72: 3964R Protocol

Variable	Meaning
Task state	Actual state of the protocol process: 0 = not initialized 1 = basic state 2 = telegram sequence time expires 3 = sending mode 4 = receiving mode
Send telegrams	Number of faultless transmitted telegrams
Receive telegrams	Number of faultless received telegrams
Send repetitions	Number of send repetitions
Receive repetitions	Number of receive repetitions
Send errors	Number of send tasks which got lost because of Syntax or transmission errors
Receive errors	Number of errors which have occurred by receiving data
Error bits (hex)	Assigns every reported error to an error class and shows this by setting a bit. It is only displayed, if the error has executed in spite of repeat a telegram loss
Last error	Number of the last reported error. It is only displayed, if the error has occurred in spite of repeat a telegram loss The description of the error numbers see on the CD in the manual Nvr_pre.pdf

Table 68: 3964R Protocol

9.1.5 Extended Device Diagnostic RK512

In this section the task states for RK512 devices are described.

9.1.5.1 RK512 Protocol

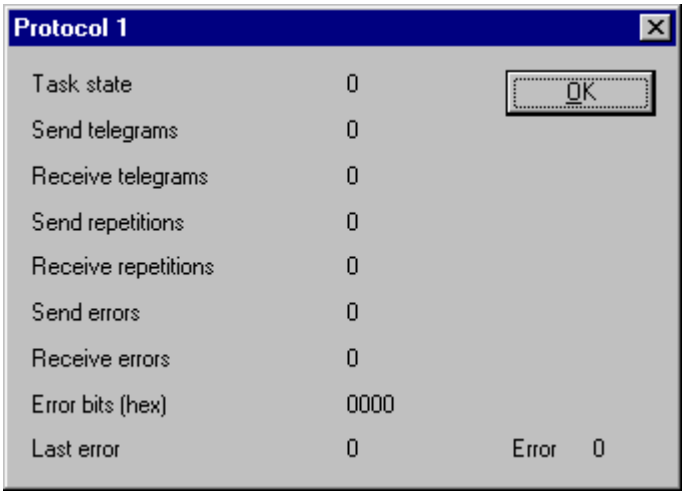

Figure 73: RK512 Protocol

Variable	Meaning
Task state	Actual state of the protocol process: 0 = not initialized 1 = basic state 2 = telegram sequence time expires 3 = sending mode 4 = receiving mode
Send telegrams	Number of faultless transmitted telegrams
Receive telegrams	Number of faultless received telegrams
Send repetitions	Number of send repetitions
Receive repetitions	Number of receive repetitions
Send errors	Number of send tasks which got lost because of Syntax or transmission errors
Receive errors	Number of errors which have occurred by receiving data
Error bits (hex)	Assigns every reported error to an error class and shows this by setting a bit. It is only displayed, if the error has executed in spite of repeat a telegram loss
Last error	Number of the last reported error. It is only displayed, if the error has occurred in spite of repeat a telegram loss The description of the error numbers see on the CD in the manual Rk_prd.pdf

Table 69: RK512 Protocol

9.1.6 Extended Device Diagnostic Modbus RTU

In this section the task states for Modbus RTU devices are described.

9.1.6.1 Modbus RTU Protocol

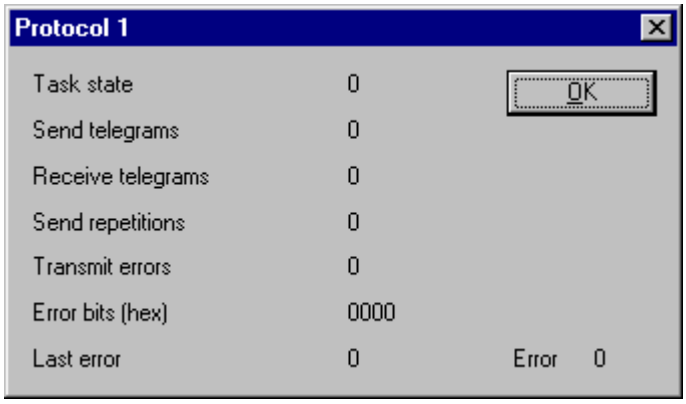

Figure 74: Modbus RTU Protocol

Variable	Meaning
Task state	Actual state of the protocol process
Send telegrams	Number of faultless transmitted telegrams
Receive telegrams	Number of faultless received telegrams
Send repetitions	Gives the number of repetitions of the send telegram before it was transmitted faultless or the maximum number of repetitions was reached
Transmit errors	Gives the number how much send task got lost in case of Syntax- or transmission errors
Error bits (hex)	Assigns every reported error to an error class and shows this by setting a bit. Assigns every reported error to an error class and shows this by setting a bit. It is only displayed, if the error has executed in spite of repeat a telegram loss
Last error	Gives the number of the last detected error. It is only displayed, if the error has executed in spite of repeat a telegram loss The description of the error numbers see on the CD in the manual Mbr_pre.pdf

Table 70: Modbus RTU Protocol

9.1.7 Extended Device Diagnostic Modbus Plus

In this section the task states for Modbus Plus devices are described.

9.1.7.1 Modbus Plus Parameter

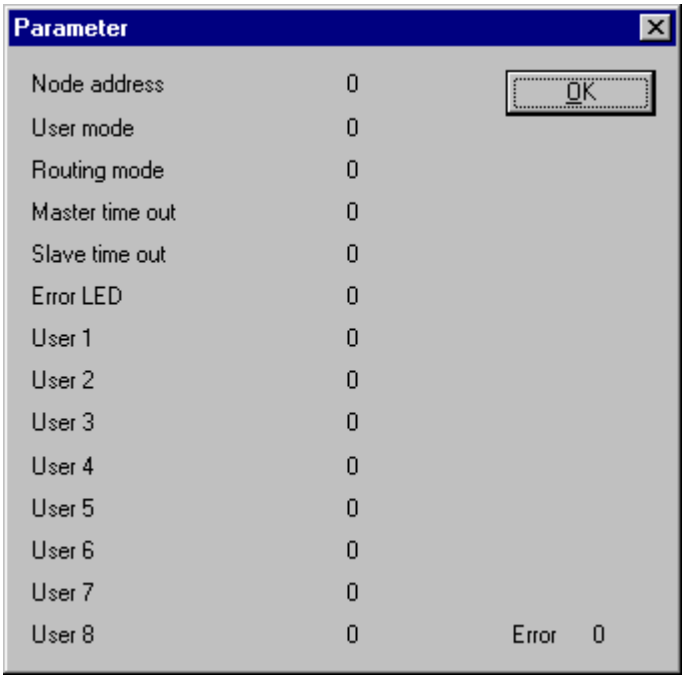

Figure 75: Modbus Plus Parameter

Variable	Meaning
Node address	Node address of the device [1..64]
User mode	User mode [0=Single-User, 1=Multi-User]
Routing mode	Routing mode [0=implicit, 1=Table]
Master time out	Task control time for Master tasks [ms]
Slave time out	Task control time for Slave tasks [ms]
Error LED	Mode of the error LED [0=set/delete, 1=set]
User 1	Use of path 1, [0=not used, 1=Client, 2=Server]
User 2	Use of path 2, [0=not used, 1=Client, 2=Server]
User 3	Use of path 3, [0=not used, 1=Client, 2=Server]
User 4	Use of path 4, [0=not used, 1=Client, 2=Server]
User 5	Use of path 5, [0=not used, 1=Client, 2=Server]
User 6	Use of path 6, [0=not used, 1=Client, 2=Server]
User 7	Use of path 7, [0=not used, 1=Client, 2=Server]
User 8	Use of path 8, [0=not used, 1=Client, 2=Server]

Table 71: Modbus Plus Parameter

9.1.7.2 Modbus Plus Interface Status

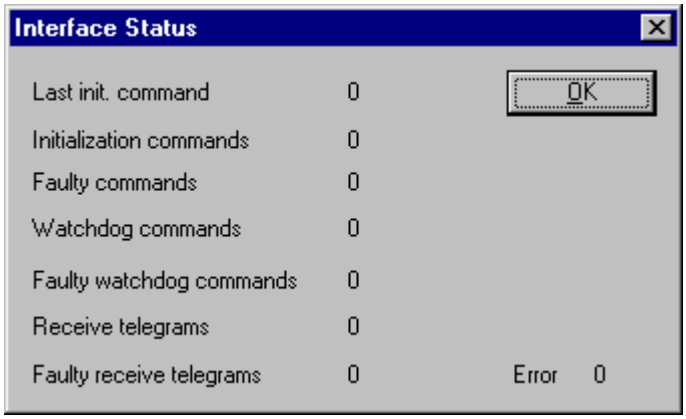

Figure 76: Modbus Plus Interface Status

Variable	Meaning
Last init. command	Identification of the last command to the interface
Initialization commands	Number of initialization commands to the interface
Faulty commands	Number of faulty commands of the interface
Watchdog commands	Number of watchdog commands to the interface
Faulty watchdog commands	Number of faulty watchdog commands
Receive telegrams	Number of received telegrams
Faulty receive telegrams	Number of faulty received telegrams

Table 72: Modbus Plus Interface Status

9.1.7.3 Modbus Plus Master Status

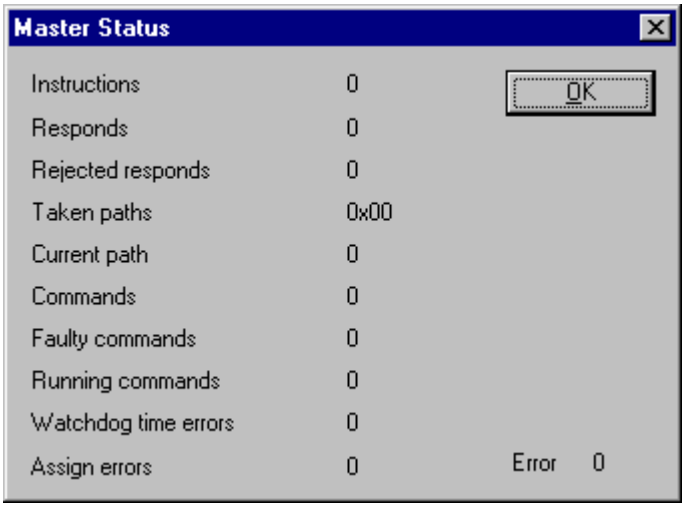

Figure 77: Modbus Plus Master Status

Variable	Meaning
Instructions	Number of instructions to the Modbus Plus Master
Responds	Number of answers to the transmitter of the command
Rejected responds	Number of rejected task because of faulty telegram data or taken interface
Taken paths	Bit string of the actual taken Master paths
Current path	Number of Master path which is actually processed
Commands	Number of commands at the Modbus Plus interface
Faulty commands	Number of receipted commands by the interface
Running commands	Number of outstanding task in the service administration
Watchdog time errors	Number of tasks which were interrupted by a timeout
Assign errors	Number of task which can not be assigned or entered in the service queue

Table 73: Modbus Plus Master Status

9.1.7.4 Modbus Plus Slave Status

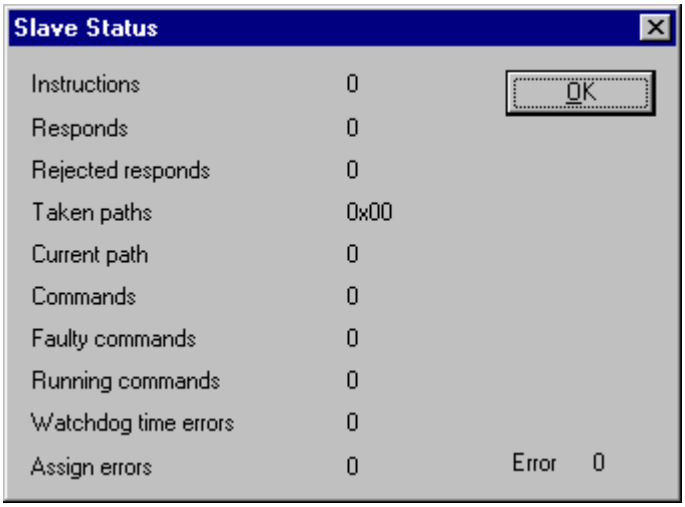

Figure 78: Modbus Plus Slave Status

Variable	Meaning
Instructions	Number of instructions to the Modbus Plus Slave
Responds	Number of answers to the transmitter of the command
Rejected responds	Number of rejected task
Taken paths	Bit string of the actual taken Slave paths
Current path	Number of the Slave path, which is actually processed
Commands	Number of commands at the Modbus Plus interface
Faulty commands	Number of receipted commands by the interface
Running commands	Number of outstanding task in the service administration
Watchdog time errors	Number of tasks which were interrupted by a timeout
Assign errors	Number of task which can not be assigned or entered in the service queue

Table 74: Modbus Plus Slave Status

9.1.8 Extended Device Diagnostic Modnet 1/N

In this section the task states for Modnet 1/N devices are described.

9.1.8.1 Modnet 1/N Protocol

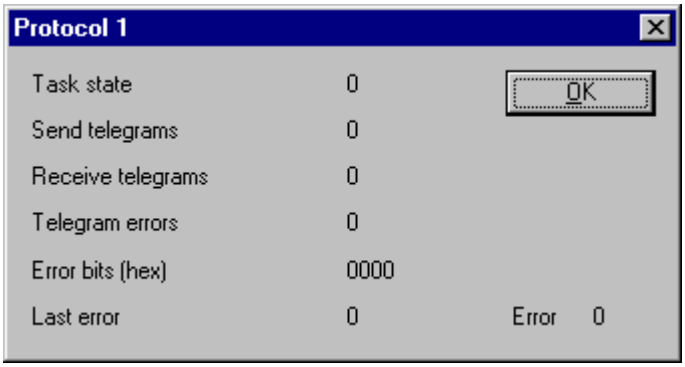

Figure 79: Modnet 1/N Protocol

Variable	Meaning
Task state	Actual state of the task process: 1 waiting for data 10 process data 11 send data 20 receive data (not for BUS-1-MA)
Send telegrams	Number of send telegrams
Receive telegrams	Number of received telegrams
Telegram errors	Number of telegrams which were not transmitted. It counts faulty telegrams in the receive and in the send direction
Error bits (hex)	Bit 0 received invalid telegram Bit 1 BCC error in the telegram Bit 2 receive buffer overflow Bit 4 Parity error Bit 5 Overrun Bit 6 Framingerror Bit 8 the Slave is not longer polled Bit 9 send telegram of the Slave was not receipted by the Master Bit 10 no message segment available Bit 11 no message segment available Bit 13 initialization not possible Bit 14 timeout of the Slave Bit 15 timeout of the Slave by the ZKA
Last error	Last occurred error The description of the error numbers see on the CD in the manual m1n_prd.pdf

Table 75: Modnet 1/N Protocol

9.1.9 Extended Device Diagnostic Modnet 1/SFB

In this section the task states for Modnet 1/SFB devices are described.

9.1.9.1 Modnet 1/SFB Protocol

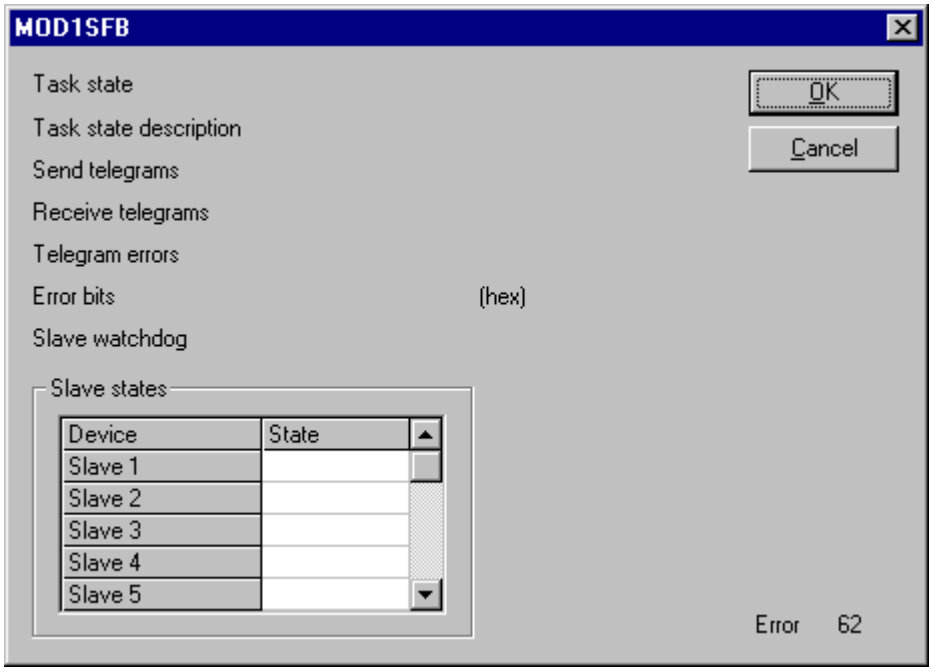

Figure 80: Mod1/SFB

Variable	Meaning
Task state	Actual state of the protocol process
Task state description	Further description of the state of the protocol process
Send telegrams	Number of faultless transmitted telegrams
Receive telegrams	Number of faultless received telegrams
Telegram errors	Gives the number how much send task got lost in case of Syntax- or transmission errors
Error bits (hex)	Assigns every reported error to an error class and shows this by setting a bit. It is only displayed, if the error has executed in spite of repeat a telegram loss
Slave watchdog	Is set, if a valid telegram arrives in the Slave mode
Slave states	State of the in the list entered Slaves

Table 76: Mod1/SFB

9.2 Full Duplex and Half Duplex

Full Duplex:
Full Duplex is the designation for a process of data transmission which allows a simultaneously transmission of the data in both directions. That means both devices can send and receive simultaneously.

Half Duplex:

In contrast to Full Duplex where two wires can be used to transmit data (one for send and one for receive) the Half Duplex works mutually with one channel. That means one the device can send and the other device can receive.

9.3 Twisted Pair and AUI

Twisted Pair:
Twisted Pair cable are used for point to point connections and they need four cores therefore two pairs of cores for separate towards and back transmission. Partly this cable has to be crossed, for example by using two repeaters. In view of faster network constellations everywhere four double wires (eight wires) with Cat 5 cable should be used.
Twisted Pair is exclusively connected with RJ45 connectors.

AUI:
The interface between an Ethernet-A Transceiver and an Ethernet-Interface are connected by a direct connection between the Transceiver and the Interface or via an AUI cable.

9.4 MAC Address

(Media Access Control-Address) This is the hardware address of a component in the network which can generate the data packet on its own. The MAC Address is given by the manufacturer who has a reserved memory location. The manufacturers guarantee that the address is unique.

10 Lists

10.1 List of Figures

10.2 List of Tables

11 Glossary

SyCon

System Configurator.

Configurations- und Diagnostic tool.