



Operating Instruction Manual

# **SyConDN**

## **System Configurator DeviceNet**

Edition: 7

Language: English (EN)

### **Hilscher Gesellschaft für Systemautomation mbH**

Rheinstraße 15  
D-65795 Hattersheim  
Germany

Tel. +49 (6190) 99070  
Fax. +49 (6190) 990750

Sales: +49 (6190) 99070  
Hotline and Support: +49 (6190) 990799

Sales email: [sales@hilscher.com](mailto:sales@hilscher.com)  
Hotline and Support email: [hotline@hilscher.com](mailto:hotline@hilscher.com)

Web: [www.hilscher.com](http://www.hilscher.com)

# List of Revisions

| Index | Date     | Version                             | Chapter                    | Revision   |
|-------|----------|-------------------------------------|----------------------------|--|
| 1     | 18.03.98 | 2.000                               | all                        | created  |
| 2     | 28.05.99 | 2.600 DevNet.DLL<br>2.095 SyCon.exe | 3<br>9                     | added: Licensing<br>Error Numbers completed<br>All chapters completed  |
| 3     | 15.12.00 | 2.661 DevNet.DLL<br>2.638 SyCon.exe | all                        | New structure of chapters, Device Assignment,<br>Menu now focus dependent / focus controlled,<br>New functions: I/O Monitor, Message Monitor |
| 4     | 20.08.01 | 2.661 DevNet.DLL<br>2.638 SyCon.exe | all                        | further overview tables, diagnostic functions, user data transfer  |
| 5     | 17.12.01 | 2.672 DevNet.DLL<br>2.638 SyCon.exe | all<br>7.3.2<br>6.5.5.1    | revised<br>CSV Export<br>Change MAC-ID   |
| 6     | 09.01.04 | 2.695 DevNet.DLL<br>2.650 SyCon.exe | all<br>5.1<br>6.7.1<br>9.3 | revised<br>Section Device Assignment completely revised<br>Get / Set Device Attribute<br>Section Error Numbers TCP/IP Driver new created     |
| 7     | 26.02.04 | 2.695 DevNet.DLL<br>2.650 SyCon.exe | 4.4.3.6                    | Section 'User Parameter' created   |

Although this program has been developed with great care and intensively tested, Hilscher Gesellschaft für Systemautomation mbH cannot guarantee the suitability of this program for any purpose not confirmed by us in writing.

Guarantee claims shall be limited to the right to require rectification. Liability for any damages which may have arisen from the use of this program or its documentation shall be limited to cases of intent.

We reserve the right to modify our products and their specifications at any time in as far as this contribute to technical progress. The version of the manual supplied with the program applies.

# Table of Contents

|       |   |    |
|-------|---|----|
| 1     | OVERVIEW SYCON .....  | 7  |
| 1.1   | Main Functions.....   | 7  |
| 1.2   | Properties.....   | 8  |
| 2     | INSTALLATION AND LICENSING .....  | 9  |
| 2.1   | System Requirements .....   | 9  |
| 2.2   | Software Installation .....   | 10 |
| 2.3   | Installation of the System Configurator SyCon .....                           | 12 |
| 2.4   | Licensing.....  | 14 |
| 2.4.1 | Ordering a License for the SyCon Configurator .....                           | 14 |
| 2.4.2 | Enter the License Code .....  | 15 |
| 2.5   | Scope of functions of the basic version and unlicensed Fieldbus Modules ..... | 17 |
| 3     | GETTING STARTED - CONFIGURATION STEPS .....                                   | 19 |
| 3.1   | Overview Communication Types .....  | 19 |
| 3.2   | Configuration Hilscher DeviceNet Master to any DeviceNet Device (Slave) ..... | 20 |
| 3.3   | Configuration Hilscher DeviceNet Slave to any DeviceNet Master .....          | 22 |
| 3.4   | Configuration Hilscher DeviceNet Master to Hilscher DeviceNet Slave .....     | 23 |
| 4     | CONFIGURATION OF DEVICENET WITH SYCON .....                                   | 25 |
| 4.1   | Setting up the DeviceNet Configuration.....                                   | 25 |
| 4.2   | EDS Files (Electronic Data Sheet Files) .....                                 | 26 |
| 4.2.1 | Introduction .....  | 26 |
| 4.2.2 | EDS files and SyCon .....   | 27 |
| 4.3   | Master .....  | 28 |
| 4.3.1 | Insert Master.....  | 28 |
| 4.3.2 | Replace Master .....  | 30 |
| 4.4   | Device (Slave).....   | 31 |
| 4.4.1 | Insert Device (Slave) .....   | 31 |
| 4.4.2 | Replace Slave .....   | 32 |
| 4.4.3 | Device Configuration .....  | 33 |
| 5     | SETTINGS .....  | 41 |
| 5.1   | Device Assignment .....   | 41 |
| 5.1.1 | Driver Selection .....  | 41 |
| 5.1.2 | CIF Device Driver .....   | 43 |
| 5.1.3 | CIF Serial Driver .....   | 45 |
| 5.1.4 | CIF TCP/IP Driver.....  | 47 |
| 5.2   | Bus Parameter .....   | 51 |
| 5.3   | DeviceNet Master.....   | 52 |
| 5.3.1 | DeviceNet Master Settings .....   | 52 |

|       |  |    |
|-------|--|----|
| 5.3.2 | Auto Addressing .....  | 53 |
| 5.3.3 | Addressing Mode.....   | 54 |
| 5.4   | Device (Slave).....  | 56 |
| 5.4.1 | Device Configuration .....                                       | 56 |
| 5.4.2 | Device Settings.....   | 56 |
| 5.5   | Project Information .....  | 57 |
| 5.6   | Path .....   | 58 |
| 5.7   | Language .....   | 58 |
| 5.8   | Start Options .....  | 59 |
| 6     | ONLINE FUNCTIONS .....   | 61 |
| 6.1   | Introduction .....   | 61 |
| 6.2   | Online to the CIF .....  | 61 |
| 6.2.1 | Downloading the Configuration .....                              | 61 |
| 6.2.2 | Firmware Download.....   | 62 |
| 6.2.3 | Firmware / Reset .....   | 63 |
| 6.2.4 | Device Info.....   | 63 |
| 6.2.5 | Activate Driver - Driver Licensing .....                         | 64 |
| 6.3   | Automatic Network Scan .....                                     | 65 |
| 6.4   | Start/Stop Communication .....                                   | 69 |
| 6.5   | Diagnostic Functions .....                                       | 70 |
| 6.5.1 | Live List.....   | 71 |
| 6.5.2 | Debug Mode .....   | 72 |
| 6.5.3 | Global State Field .....   | 74 |
| 6.5.4 | Extended Device Diagnostic .....                                 | 76 |
| 6.6   | User Data Transfer.....  | 78 |
| 6.6.1 | I/O Monitor .....  | 79 |
| 6.6.2 | I/O Watch.....   | 80 |
| 6.7   | DeviceNet Services .....   | 83 |
| 6.7.1 | Get Device Attribute / Set Device Attribute .....                | 83 |
| 6.7.2 | Change MAC-ID .....  | 84 |
| 6.8   | Message Monitor.....   | 85 |
| 6.8.1 | Message Monitor for testing explicit messaging of DeviceNet..... | 87 |
| 7     | FILE, PRINT, EXPORT, EDIT AND VIEW .....                         | 89 |
| 7.1   | File .....   | 89 |
| 7.1.1 | Open .....   | 89 |
| 7.1.2 | Save and Save As .....   | 89 |
| 7.1.3 | Close.....   | 89 |
| 7.2   | Print .....  | 90 |
| 7.3   | Export Functions .....   | 91 |
| 7.3.1 | DBM Export .....   | 91 |
| 7.3.2 | CSV Export.....  | 91 |
| 7.4   | Edit.....  | 94 |
| 7.4.1 | Delete .....   | 95 |

|         |   |     |
|---------|---|-----|
| 7.4.2   | Replace.....  | 95  |
| 7.5     | View of the Configuration .....                                     | 96  |
| 7.5.1   | Device Table.....   | 96  |
| 7.5.2   | Address Table .....   | 97  |
| 7.6     | View Menu SyCon.....  | 98  |
| 7.6.1   | Logical Network View .....  | 98  |
| 7.6.2   | Toolbars.....   | 98  |
| 7.6.3   | Status Bar .....  | 98  |
| 8       | TOOLS .....   | 99  |
| 8.1     | PKV40 / PKV 50 Gateway.....   | 99  |
| 9       | ERROR NUMBERS .....   | 101 |
| 9.1     | CIF Device Driver (Dual-port memory) Error Numbers (-1 .. -49)..... | 101 |
| 9.2     | CIF Serial Driver Error Numbers (-20 .. -71).....                   | 105 |
| 9.3     | CIF TCP/IP Driver Error Numbers .....                               | 107 |
| 9.3.1   | Standard Win32 Socket API Errors .....                              | 107 |
| 9.3.2   | Specific NetIdent Errors.....                                       | 107 |
| 9.4     | RCS Error Numbers (4 .. 93).....                                    | 108 |
| 9.5     | Database Access Error Numbers (100 .. 130).....                     | 110 |
| 9.6     | Online Data Manager Error Numbers .....                             | 111 |
| 9.6.1   | Online Data Manager Error Numbers (1000 .. 1018).....               | 111 |
| 9.6.2   | Message Handler Error Numbers (2010 .. 2027).....                   | 111 |
| 9.6.3   | Driver Functions Error Numbers (2501 .. 2512).....                  | 112 |
| 9.6.4   | Online Data Manager Sub Functions Error Numbers (8001 .. 8035)..... | 112 |
| 9.7     | Data Base Functions Error Numbers (4000 .. 4199).....               | 113 |
| 9.8     | Converting Functions Error Numbers (5001 .. 5008) .....             | 117 |
| 10      | APPENDIX .....  | 119 |
| 10.1    | Extended Device Diagnostic Master.....                              | 119 |
| 10.1.1  | PLC_TASK Common Variables .....                                     | 119 |
| 10.1.2  | DNM_TASK Common Variables .....                                     | 119 |
| 10.1.3  | DNM_TASK Device Running States .....                                | 121 |
| 10.1.4  | DNM_Task Global State Field .....                                   | 121 |
| 10.1.5  | DNM_Task Communication Error .....                                  | 121 |
| 10.1.6  | DNM_Task Receive Queue.....   | 122 |
| 10.1.7  | DNM_Task Transmit Queue.....  | 122 |
| 10.1.8  | DNM_Task DeviceNet Command Counters.....                            | 123 |
| 10.1.9  | DNM_Task Timeout Counter.....                                       | 124 |
| 10.1.10 | DNM_Task Init Counter.....  | 124 |
| 10.2    | Extended Device Diagnostic Device (Slave).....                      | 125 |
| 10.2.1  | PLC_Task Common Variables (Device).....                             | 125 |
| 10.2.2  | DNS_Task Common Variables .....                                     | 125 |
| 10.2.3  | DNS_TASK Receive Queue (Device) .....                               | 127 |
| 10.2.4  | DNS_TASK Transmit Queue (Device) .....                              | 127 |

|      |                       |   |
|------|-----------------------|---|
| 11   | LISTS .....           | 129                                       |
| 11.1 | List of Figures ..... | 129                                       |
| 11.2 | List of Tables.....   | 131                                       |
| 12   | GLOSSARY.....         | <b>FEHLER! TEXTMARKE NICHT DEFINIERT.</b> |
| 13   | INDEX .....           | <b>FEHLER! TEXTMARKE NICHT DEFINIERT.</b> |

# 1 Overview SyCon

## 1.1 Main Functions

The main functions of the DeviceNet System Configurator are:

| Function      | Section                             | Short Description  |
|---------------|-------------------------------------|--|
| Configuration | <i>Overview Communication Types</i> | Overview communication types and description of the configuration steps        |
|               | <i>Automatic Network Scan</i>       | Scans the network  |
| Diagnostic    | <i>Diagnostic Functions</i>         | Diagnostic functions: for example Live List, Debugger, Global State Field etc. |
|               | <i>User Data Transfer</i>           | I/O Monitor, I/O Watch, Message Monitor, Get attribute and Set attribute       |
| Documentation | <i>Project Information</i>          | Set the project information  |
|               | <i>Print</i>                        | Print out the configuration  |

Table 1: SyCon Main Functions

## 1.2 Properties

### **SyCon is an 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 4.0/2000/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 or higher
- Windows NT: Service Pack 6 or higher
- COM/DCOM only for OPC Server
- Keyboard and Mouse

## 2.2 Software Installation

Close all application programs on the system!

Insert the CD Hilscher System software 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**.

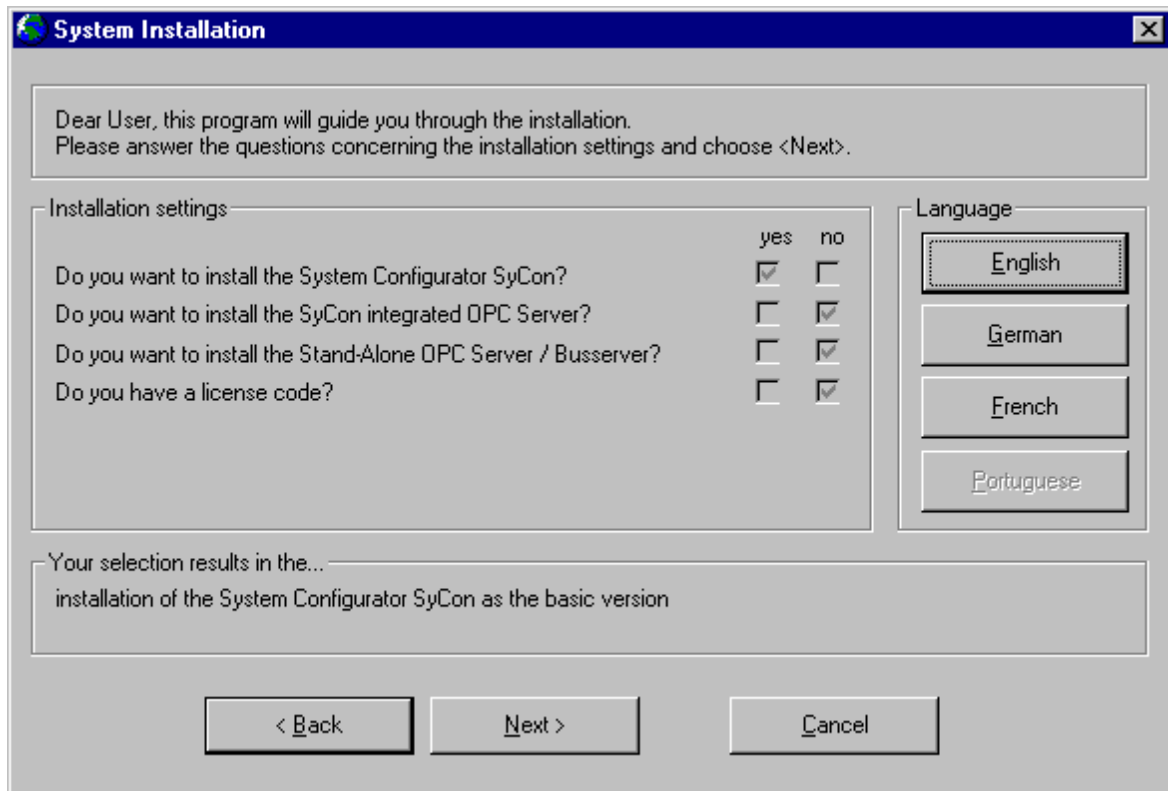


Figure 1: Selection for the Installation of the System Configurator in basic version

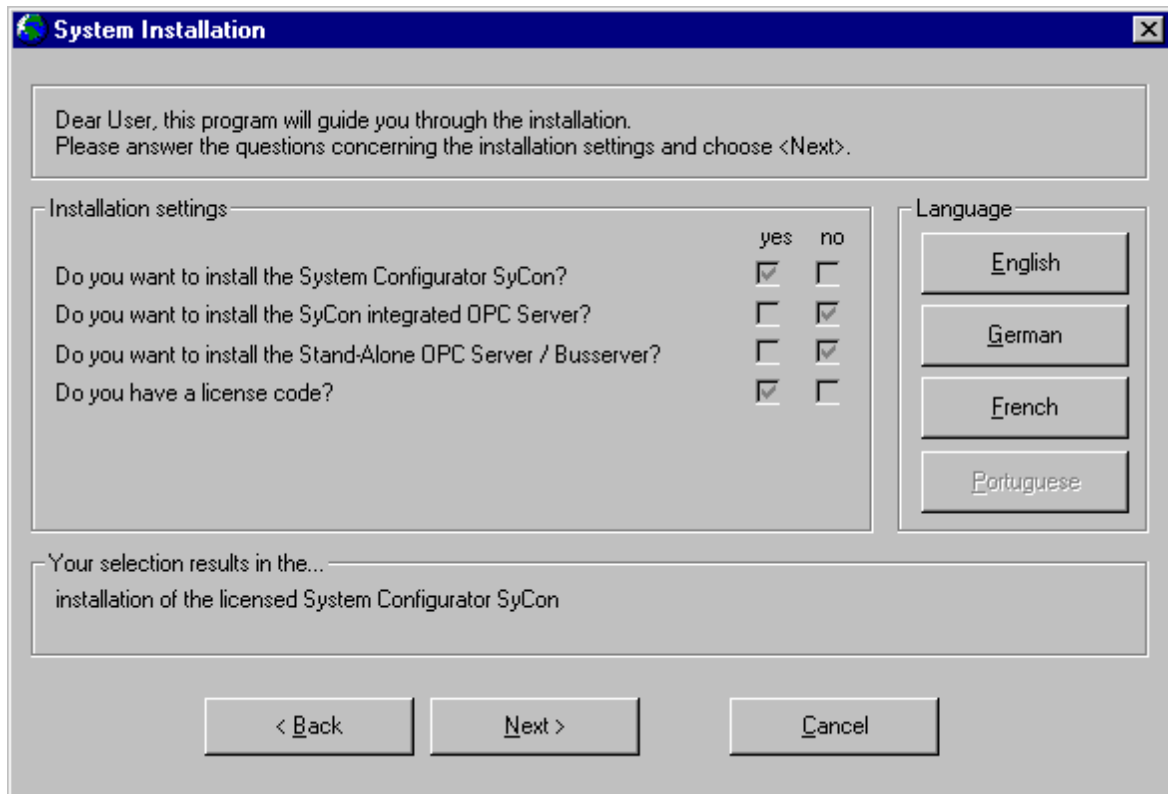


Figure 2: Selection for the Installation of the licensed System Configurator

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)

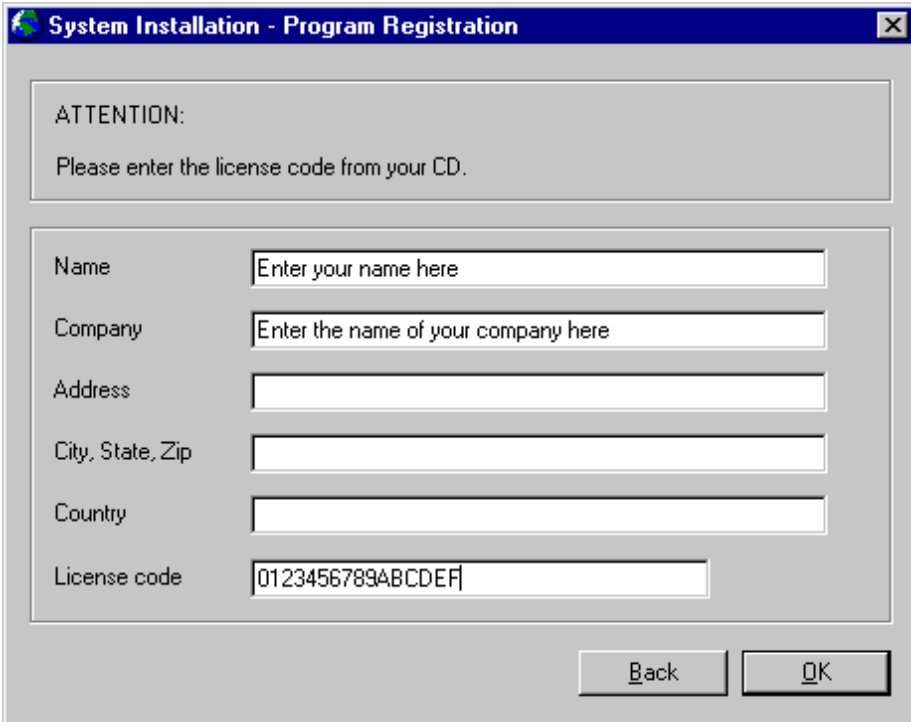
If you have a license code or it is printed on the label of the CD, then answer the question for an existing license code with yes, otherwise a basic version of the System Configurator will be installed. Enter your name and the company name.

## 2.3 Installation of the System Configurator SyCon

During the installation the user and the company name must be entered. If you have a license code or it is printed on the label of the CD, it must also be entered now. Otherwise the System Configurator will work as a basic version. In this case, all functions are available, but the configuration is limited to two devices on the network, which is sufficient for Slave devices.

A license can be ordered by filling out the order form under the menu item **Help > Licensing** and fax this order form either to the distributor or directly to us.

Follow the instructions of the installation program by selecting the fieldbus system to be installed and answer all the questions with **OK** or **NEXT**.



The screenshot shows a Windows-style dialog box titled "System Installation - Program Registration". Inside, there is a section labeled "ATTENTION:" with the text "Please enter the license code from your CD." Below this, there are several input fields: "Name" (with placeholder "Enter your name here"), "Company" (with placeholder "Enter the name of your company here"), "Address", "City, State, Zip", "Country", and "License code" (which contains the text "0123456789ABCDEF"). At the bottom right of the dialog are two buttons: "Back" and "OK".

Figure 3: Enter the Name, the Company Name and the License code

---

**Note:** The License Code 0123456789ABCDEF is no valid code and is only used for explanation.

---

It is necessary to fill in the Name and the Company Name. It is optional to fill in the Address, the City, State, Zip and Country.

The installation program offers the following selections:

| Selection         | Default Settings                          | Meaning   |
|-------------------|---|---|
| Directory         | C:\Programs\Hilscher\SyCon                | Directory for Installation of the SyCon and its Components        |
| AS-Interface      | Selected                                  | Program DLL and Components of the Fieldbus System or the Protocol |
| CANopen           | Selected                                  |   |
| ControlNet        | Selected                                  |   |
| DeviceNet         | Selected                                  |   |
| InterBus          | Selected                                  |   |
| PROFIBUS          | Selected                                  |   |
| Ethernet/Protocol | Selected                                  |   |
| SDS               | Selected                                  |   |
| CIF Device Driver | Selected<br>C:\Programs\CIF Device Driver | CIF Device Driver   |
| Program Menu      | SyCon System Configurator                 | Folder under Start > Programs                                     |

Table 2: Selection during installation

The installation program copies the program files, GSD or EDS files and Bitmaps to the PC. Finally

- System DLLs
  - The Application
  - OLE Controls
  - ODBC Components
- are entered into the Registry.

## 2.4 Licensing

This section describes the steps to license the System Configurator from the already installed basic version of the System Configurator. To license the System Configurator during installation was already described above.

Deliveries that contain a license for the System Configurator have a form with. Fill out this paper (form) and fax it to your distributor or directly to us. After you receive the license code enter it as described in section *Enter the License Code* as described below on page 15.

An order form for a license for the System Configurator can be printed out and is described in the next section.

### 2.4.1 Ordering a License for the SyCon Configurator

To order the license code for the selected fieldbus systems select the menu **Help > Licensing**. The licensing window will be opened.

Fill in your name, the company name and the address for license information into the fields.

Select one more fieldbus modules. There are three tables to do this. The first table lists the modules, which are not licensed. Double-click or select and click the **Add** button to move the desired modules into the table in the middle which is printed on the order form later. The modules, which are already licensed, are shown in the last table.

**Licensing**

Licensee Information

Name: Enter your name here

Company: Enter your company name here

Address: Enter address

City, State, Zip: Enter city, state, zip

Country: Enter country

OK Cancel

Enter License Code...

Print Order Form...

Licensing of the fieldbus systems

Not licensed

| Module   | Version    | Date       |
|----------|------------|------------|
| Canopen  | 2, 6, 6, 2 | 10/10/2000 |
| Interbus | 2, 6, 5, 2 | 10/10/2000 |

License ordered

| Module | Version    | Date       |
|--------|------------|------------|
| DEVNet | 2, 6, 6, 1 | 10/10/2000 |

Add Delete

License presented

| Module | Version | Date |
|--------|---------|------|
|--------|---------|------|

Figure 4: Example for selecting the fieldbus module DeviceNet

After selecting the modules select the button **Print Order Form** and send us this paper by fax or by mail.

## 2.4.2 Enter the License Code

This section describes the steps to license the System Configurator from the already installed basic version of the System Configurator. To license the System Configurator during installation was already described above.

Select the menu **Help > Licensing**. The licensing window will be opened.

In the table in the middle are listed the fieldbus modules that were already selected for the order form. If this is not the case then select the fieldbus modules from the upper table by double click or by select and **Add**.

Check if the name and the company name were entered exactly as printed on the fax. Observe that the spelling is the same as on the fax, especially the small and capital letters.

Then select the button **Enter License Code**. The following window appears. Enter the 16 digits of the license code.

**Note:** License codes with less than 16 digits can only be entered during the installation. In this case deinstall the System Configurator first and then restart the installation and enter the code. Also this System Configurator (license codes with less than 16 digits) expects a license in the device.

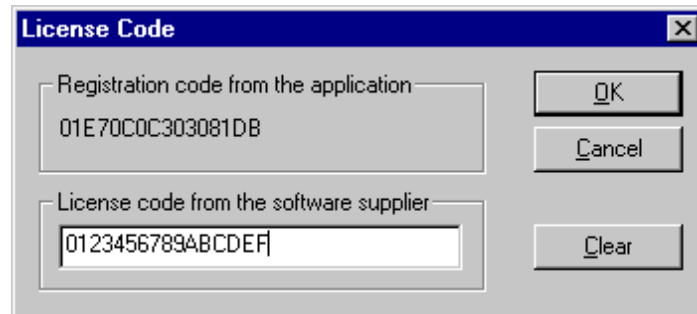


Figure 5: Enter the License code

**Note:** The license code showed above is an invalid license code and is only used for explanation.

After you have entered the license code select the button **OK**. The code is verified. If the license code is valid SyCon will ask you to exit and restart the System Configurator to activate the license. If the license code is invalid the following window appears.



Figure 6: Note license code is invalid

In this case check

- the license code with the information on the fax
- the right spelling of the name and the company name with the information on the fax. Check especially for small and capital letters.



## 2.5 Scope of functions of the basic version and unlicensed Fieldbus Modules

The basic mode and unlicensed fieldbus modules have the following functionality:

- Full functionality for configuring up to two devices. For the configuration of a Hilscher Slave device this is enough.
- All diagnostic functions
- Open and download of an existing configuration file. If the configuration file has more than two devices, a modification of this configuration is not possible.



## 3 Getting Started - Configuration Steps

### 3.1 Overview Communication Types

Select from the following the communication that you want to use. The configuration steps are described in the given chapter.

**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.

| Communication | Device                    | Device                   | Described in section   | Page |
|---------------|---------------------------|--------------------------|--|------|
| DeviceNet     | Hilscher DeviceNet Master | Any DeviceNet Device     | <i>Configuration Hilscher DeviceNet Master to any DeviceNet Device</i>     | 20   |
|               | Hilscher DeviceNet Slave  | Any DeviceNet Master     | <i>Configuration Hilscher DeviceNet Slave to any DeviceNet Master</i>      | 22   |
|               | Hilscher DeviceNet Master | Hilscher DeviceNet Slave | <i>Configuration Hilscher DeviceNet Master to Hilscher DeviceNet Slave</i> | 23   |

Table 3: Overview Communication Types

## 3.2 Configuration Hilscher DeviceNet Master to any DeviceNet Device (Slave)

The following table describes the steps to configure a Hilscher DeviceNet Master to any DeviceNet Device (Slave) as it is typical for many cases.

| #  | Action  | Menu in the System Configurator   | Detail information in section                           | Page |
|----|---|---|---|------|
| 1  | Create a new project  | <b>File &gt; New &gt; DeviceNet</b>   | <i>Setting up the DeviceNet Configuration</i>           | 25   |
| 2  | Copy EDS file of the DeviceNet device (Slave), if the device is not in the selection list | <b>File &gt; Copy EDS</b>   | <i>EDS Files (Electronic Data Sheet Files)</i>          | 26   |
| 3  | Choose Hilscher DeviceNet Master and provide MAC ID address                               | <b>Insert &gt; Master</b>   | <i>Insert Master</i>                                    | 28   |
| 4  | Choose DeviceNet device and provide MAC ID address  | <b>Insert &gt; Device</b>   | <i>Insert Device (Slave)</i>                            | 31   |
| 5  | Assign the input and output modules (*1)  | Mark the Device (left Mouse click), then<br><b>Settings &gt; Device Configuration</b>                     | <i>Device Configuration</i>                             | 33   |
| 6  | Assign the offset addresses   |   |   |      |
| 7  | Assign the Device Parameter data, if the Device needs Parameter data                      | Mark the Device (left Mouse click), then<br><b>Settings &gt; Device Configuration &gt; Parameter Data</b> | <i>Parameter Data</i>                                   | 37   |
| 8  | Set the Bus parameter   | Mark the Master (left Mouse click), then<br><b>Settings &gt; Bus Parameters</b>                           | <i>Bus Parameter</i>                                    | 51   |
| 9  | Set device assignment if no automatic assignment has occurred                             | Mark the Master (left Mouse click), then<br><b>Settings &gt; Device Assignment</b>                        | <i>Device Assignment</i>                                | 41   |
| 10 | Save project  | <b>File &gt; Save</b>   | <i>Save and Save As</i>                                 | 89   |
| 11 | Download  | Mark the Master (left Mouse click), then<br><b>Online &gt; Download</b>                                   | <i>Downloading the Configuration</i>                    | 61   |
| 12 | Live list   | Mark the Master (left Mouse click), then<br><b>Online &gt; Live List</b>                                  | <i>Live List</i>  | 71   |
| 13 | Start Debugger  | Mark the Master (left Mouse click), then<br><b>Online &gt; Start Debug Mode</b>                           | <i>Debug Mode</i>                                       | 72   |
| 14 | Device diagnostic   | Mark the Slave (left Mouse click), then<br><b>Online &gt; Device Diagnostic</b>                           | <i>Device Diagnostic</i>                                | 73   |
| 15 | Stop Debugger   | <b>Online &gt; Stop Debug Mode</b>  | <i>Debug Mode</i>                                       | 72   |
| 16 | Global Diagnostic   | Mark the Master (left Mouse click), then<br><b>Online &gt; Global State Field</b>                         | <i>Global State Field</i>                               | 74   |
| 17 | Transfer user data:   | Mark the Master (left Mouse click), then<br><b>Online &gt; I/O Monitor</b>                                | <i>I/O Monitor or<br/>alternatively: I/O Watch (*2)</i> | 79   |
|    | Write output, read input  |   |   | 80   |

Table 4: Step for Configuration Hilscher DeviceNet Master to any DeviceNet Device

Notes see next page.

**Note (\*1):** The Offset addresses assigned in the Device (Slave) configuration are always related to the Hilscher DeviceNet Master and have no meaning here.

---

**Note (\*2):** Alternatively the CIF Device Driver Test program can be used for the test. After Board Select: Data Transfer > I/O Data > DevExchangeIO.

---

### 3.3 Configuration Hilscher DeviceNet Slave to any DeviceNet Master

The following table describes the steps to configure a Hilscher DeviceNet Slave to any DeviceNet Master as it is typical for many cases.

| # | Action   | Menu in the System Configurator   | Detail information in section                           | Page     |
|---|--|---|---|----------|
| 1 | Create a new project   | <b>File &gt; New &gt; DeviceNet</b>   | <i>Setting up the DeviceNet Configuration</i>           | 25       |
| 2 | Choose DeviceNet Master (*1) and provide MAC ID address          | <b>Insert &gt; Master</b>   | <i>Insert Master</i>                                    | 28       |
| 3 | Choose Hilscher DeviceNet device (Slave) and provide Mac address | <b>Insert &gt; Device</b>   | <i>Insert Device (Slave)</i>                            | 31       |
| 4 | Assign the input and output modules (*2)                         | Mark the Device (left Mouse click), then<br><b>Settings &gt; Device Configuration</b> | <i>Device Configuration</i>                             | 33       |
| 5 | Set device assignment if no automatic assignment has occurred    | Mark the Device (left Mouse click), then<br><b>Settings &gt; Device Assignment</b>    | <i>Device Assignment</i>                                | 41       |
| 6 | Save project   | <b>File &gt; Save</b>   | <i>Save and Save As</i>                                 | 89       |
| 7 | Download   | Mark the Slave (left Mouse click), then<br><b>Online &gt; Download</b>                | <i>Downloading the Configuration</i>                    | 61       |
| 8 | Device diagnostic  | Mark the Slave (left Mouse click), then<br><b>Online &gt; Device Diagnostic</b>       | <i>Device Diagnostic</i>                                | 73       |
| 9 | Transfer user data:<br>Write output, read input                  | Mark the Device (left Mouse click), then<br><b>Online &gt; I/O Monitor</b>            | <i>I/O Monitor or (*3)<br/>alternatively: I/O Watch</i> | 79<br>80 |

Table 5: Step for Configuration Hilscher DeviceNet Slave to any DeviceNet Master

**Note (\*1):** Insert a Hilscher DeviceNet Master. This is only a placeholder (dummy) and it is not necessary that it matches the real Master.

**Note (\*2):** The Offset addresses assigned in the Device (Slave) configuration are always related to the Hilscher DeviceNet Master and have no meaning here.

**Note (\*3):** Alternatively the CIF Device Driver Test program can be used for the test. After Board Select: Data Transfer > I/O Data > DevExchangeIO.

### 3.4 Configuration Hilscher DeviceNet Master to Hilscher DeviceNet Slave

The following table describes the steps to configure a Hilscher DeviceNet Master to a Hilscher DeviceNet Slave as it is typical for many cases.

| #  | Action   | Menu in the System Configurator   | Detail information in section                 | Page |
|----|--|---|---|------|
| 1  | Create a new project   | <b>File &gt; New &gt; DeviceNet</b>   | <i>Setting up the DeviceNet Configuration</i> | 25   |
| 2  | Choose Hilscher DeviceNet Master and provide MAC ID address                          | <b>Insert &gt; Master</b>   | <i>Insert Master</i>                          | 28   |
| 3  | Choose Hilscher DeviceNet Device (Slave) and provide MAC ID address                  | <b>Insert &gt; Device</b>   | <i>Insert Device (Slave)</i>                  | 31   |
| 4  | Assign the input and output modules  | Mark the Device (left Mouse click), then<br><b>Settings &gt; Device Configuration</b> | <i>Device Configuration</i>                   | 33   |
| 5  | Assign the offset addresses (*1)   |   |   |      |
| 6  | Set the Bus parameter  | Mark the Master (left Mouse click), then<br><b>Settings &gt; Bus Parameters</b>       | <i>Bus Parameter</i>                          | 51   |
| 7  | Set device assignment for the Master if no automatic assignment has occurred         | Mark the Master (left Mouse click), then<br><b>Settings &gt; Device Assignment</b>    | <i>Device Assignment</i>                      | 41   |
| 8  | Set device assignment for the Device (Slave) if no automatic assignment has occurred | Mark the Device (left Mouse click), then<br><b>Settings &gt; Device Assignment</b>    | <i>Device Assignment</i>                      | 41   |
| 9  | Save project   | <b>File &gt; Save</b>   | <i>Save and Save As</i>                       | 89   |
| 10 | Download to the Master   | Mark the Master (left Mouse click), then<br><b>Online &gt; Download</b>               | <i>Downloading the Configuration</i>          | 61   |
| 11 | Download to the Device (Slave)   | Mark the Device (left Mouse click), then<br><b>Online &gt; Download</b>               | <i>Downloading the Configuration</i>          | 61   |
| 12 | Live list  | Mark the Master (left Mouse click), then<br><b>Online &gt; Live List</b>              | <i>Live List</i>                              | 71   |
| 13 | Start Debugger   | Mark the Master (left Mouse click), then<br><b>Online &gt; Start Debug Mode</b>       | <i>Debug Mode</i>                             | 72   |
| 14 | Device diagnostic  | Mark the Device (left Mouse click), then<br><b>Online &gt; Device Diagnostic</b>      | <i>Device Diagnostic</i>                      | 73   |
| 15 | Stop Debugger  | <b>Online &gt; Stop Debug Mode</b>  | <i>Debug Mode</i>                             | 72   |
| 16 | Global Diagnostic  | Mark the Master (left Mouse click), then<br><b>Online &gt; Global State Field</b>     | <i>Global State Field</i>                     | 74   |
| 17 | Transfer user data:<br>Write output, read input                                      | Mark the Master (left Mouse click), then<br><b>Online &gt; I/O Monitor</b>            | <i>I/O Monitor or (*2)</i>                    | 79   |
|    |  |   | <i>alternatively: I/O Watch</i>               | 80   |

Table 6: Step for Configuration Hilscher DeviceNet Master to Hilscher DeviceNet Slave

Notes see next page.

**Note (\*1):** The Offset addresses assigned in the Device (Slave) configuration are always related to the Hilscher DeviceNet Master.

---

**Note (\*2):** Alternatively the CIF Device Driver Test program can be used for the test. After Board Select: Data Transfer > I/O Data > DevExchangeIO.

---



## 4 Configuration of DeviceNet with SyCon

### 4.1 Setting up the DeviceNet Configuration

To create a new configuration, choose the **File > New** menu. This will offer a selection list of fieldbus systems. Choose the **DeviceNet**. If only a DeviceNet fieldbus system is installed with SyCon, the configuration window will open directly.

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

## 4.2 EDS Files (Electronic Data Sheet Files)

### 4.2.1 Introduction

An Electronic Data Sheet (EDS) provides information necessary to access and alter the configurable parameters of a device. An Electronic Data Sheet (EDS) is an external file that contains information about configurable attributes for the device, including object addresses of each parameter.

The application objects in a device represent the destination addresses for configuration data. These addresses are encoded in the EDS.

The figure below shows a general block diagram of a sample EDS.

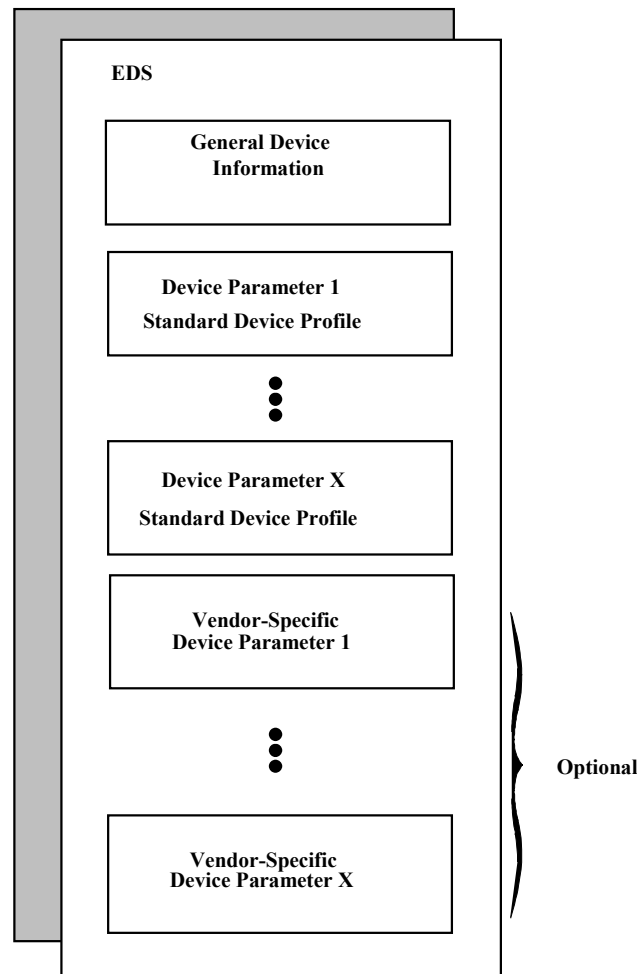


Figure 7: General block diagram of an EDS file

## 4.2.2 EDS files and SyCon

When SyCon is started, it automatically retrieves all the EDS files stored in the EDS directory. The device names are placed into an internal list. During the configuration, the device-specific data is retrieved directly from the EDS files.

If a DeviceNet device does not appear in the selection list (Insert Master or Insert Device), a corresponding EDS file can be copied into the EDS directory with **File > Copy EDS**. Another possibility is to copy the EDS file into the SyCon EDS directory with the Windows Explore and then retrieve the EDS files into the EDS directory with **Settings > Path** and **OK**.

**Hilscher devices:** The EDS files for Hilscher devices are already included and installed.

**Other devices:** The respective device manufacturer provides the EDS files for other devices.

The EDS files of some vendors are available on the DeviceNet homepage <http://www.odva.org>

or visit the homepage of the manufacturer.

The EDS directory is adjustable. In order to alter the directory from a previous setting in another directory, use the menu **Settings > Path**. All EDS files must be placed in this directory.

## 4.3 Master

### 4.3.1 Insert Master

To insert a (Hilscher) Master into the configuration, choose the **Insert > Master** menu, in order to open the selection window, or click on the symbol:

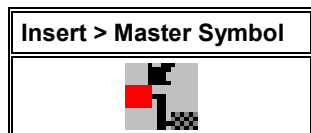


Table 7: Insert > Master Symbol

The mouse pointer automatically changes into the Insert Master pointer.

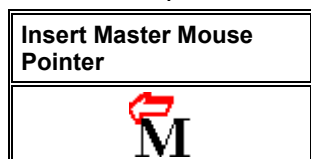


Table 8: Insert Master Mouse Pointer

Click on the position where the Master is to be inserted. The dialog box opens, from which one or more Masters can be chosen.

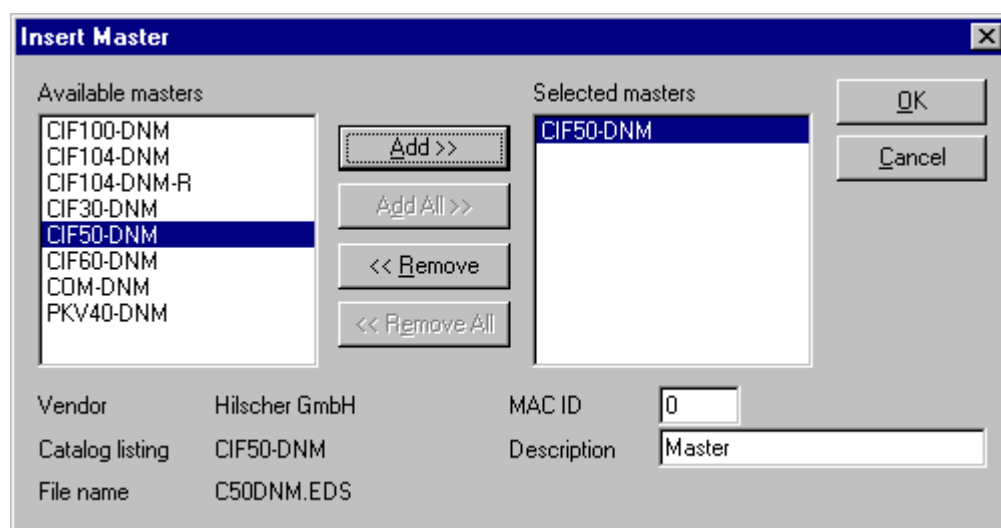


Figure 8: Insert > Master

Select a Master device in the list **Available Masters** by clicking on it. By clicking the **Add** button the Master is shown in the list **Selected Masters**. If you click the **OK** button the Master is insert in the first position of the configuration.

This example shows a CIF50-DNM. The Master gets the description Master at first. This is changeable in the field **Description**. The **MAC ID** of the Master can be changed here.

#### 4.3.1.1 Hardware Assignment

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



Figure 9: Hardware Assignment Master

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 41).

### 4.3.2 Replace Master

If a Master already exists in the configuration and should be replaced against another Master, you first have to set the focus on the Master (left mouse click at the Master) and then choose the menu **Edit > Replace**. In the opened window appears the question if the Master should be replaced.

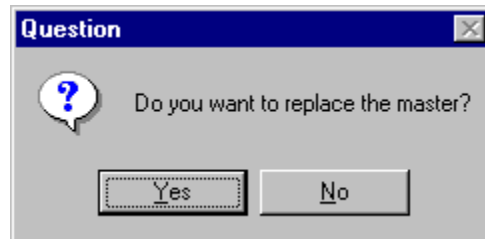


Figure 10: Security question replace Master

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

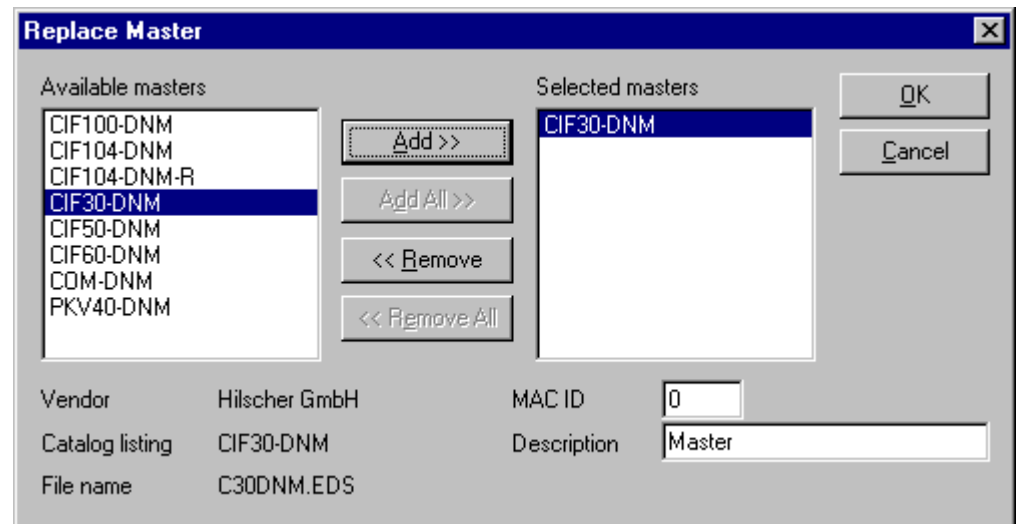


Figure 11: Edit > Replace Master

In this window you select the Master you want by clicking on it and then click the **Add** button to put the Master to **Selected devices**. With **OK** you confirm the selection and the Master will be replaced.

## 4.4 Device (Slave)

### 4.4.1 Insert Device (Slave)

To insert a new DeviceNet Slave in the configuration select the **Insert > Device** menu to open the selection window or click onto the following Icon:

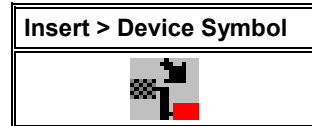


Table 9: Insert > Device Symbol

The mouse cursor changes automatically to the insert device cursor.

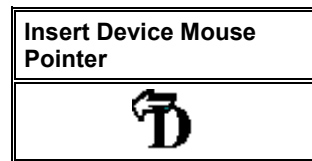


Table 10: Insert Device Mouse Pointer

Click on the position where you want to insert the new device. A dialogue box appears where you can select now one or more devices for insertion.

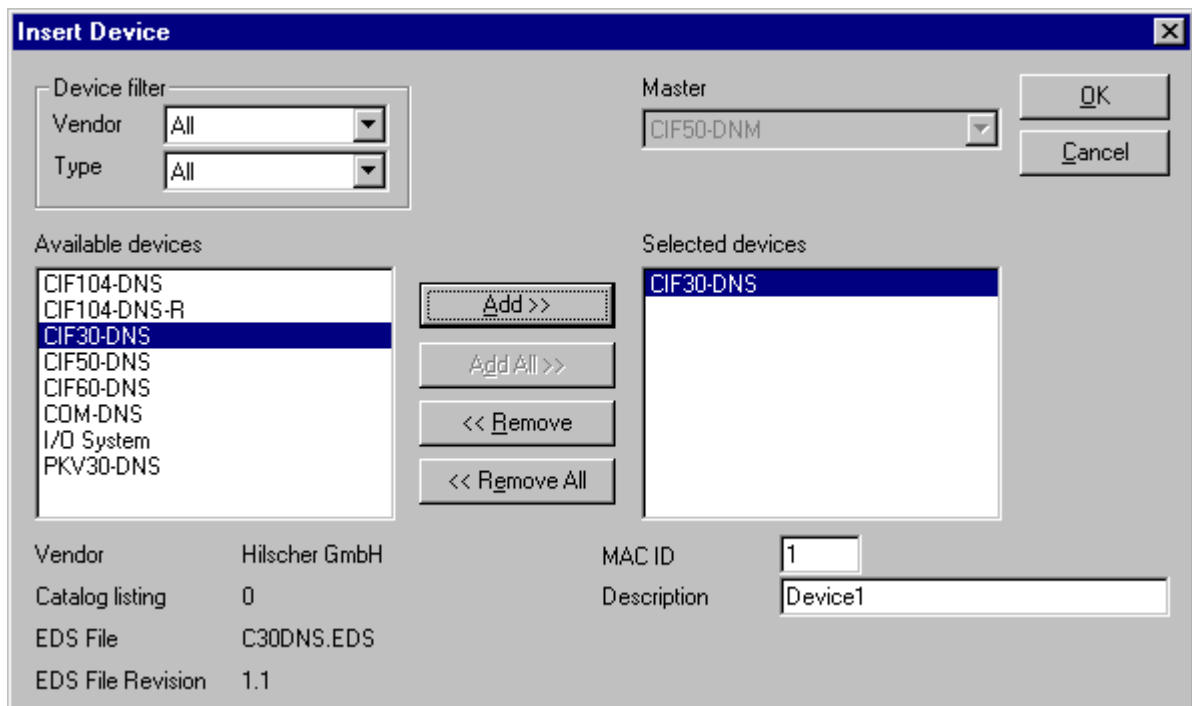


Figure 12: Insert > Device (Slave)

The list on the left displays for selection all the Slave devices whose EDS files have been put in the EDS directory. A filter can be used to limit the selection list of the manufacturer. Further information on a Slave is shown below the selection list (**Available devices**) when it is selected (one mouse click). Apart from the vendor name and the description especially the ID-Code, the I/O-Code, the file name and the file revision are given. The Slave appears on the right-hand list with a mouse click or with the **Add** button.

All devices in the right-hand list are assigned to the current insert point that is also shown in this window. If the Slaves in the right-hand list are chosen one after the other (a mouse click), then every Slave can be assigned a name in the **Description** field.

With each new selected device in the right list the MAC ID is raised to one more but it can be changed by the user in the field **MAC ID**.

---

**Note:** It is possible to select a device more than one time. But each device must have an own (unique) MAC ID to distinguish it in the network.

---

#### 4.4.2 Replace Slave

To replace an existing Slave device you first have to mark the device which has to be replaced. Then you have to proceed like it is described in section *Replace Master* at page 30.



### 4.4.3 Device Configuration

To enter the Device Configuration set the focus on the device (left mouse click) and select the menu **Settings > Device Configuration** or set the focus on the device (left mouse click) and use the right mouse button at the device or simply double-click on the device.

In the device configuration you have to assign the device's I/O to logical addresses in the process data image of the Master. Remember that the device offsets set here will be used in your application to read inputs and to write outputs.

**Device Configuration**

MAC ID: 1 File name: C50DNS.EDS

Description: Device1

☒ Activate device in actual configuration

Actual device: 1 / CIF50-DNS

Actual chosen IO connection: ☒ Poll ☐ Bit strobe ☐ Change of state ☐ Cyclic

☐ UCMM check Group 3

Parameter Data...

Connection Object Instance Attributes

Expected packet rate: 200 Production inhibit time: 10

Watchdog timeout action: Timeout Fragmented Timeout: 1600 ms

Produced connection size: 8 Consumed connection size: 8

Available predefined connection data types

| Data type  | Description | Data length |
|------------|-------------|-------------|
| BYTE ARRAY | Input Data  | 8           |
| BYTE ARRAY | Output Data | 8           |
|            |             |             |
|            |             |             |

Append to configured I/O data

Insert into configured I/O data

Configured I/O connection data and its offset address

| Data type  | Description | I Type | I Len. | I Addr. | O Type | O Len. | O Addr. |
|------------|-------------|--------|--------|---------|--------|--------|---------|
| BYTE ARRAY | Input_Data  | IB     | 8      | 0       |        |        |         |
| BYTE ARRAY | Output_Data |        |        |         | QB     | 8      | 0       |
|            |             |        |        |         |        |        |         |
|            |             |        |        |         |        |        |         |

Delete configured I/O data

Symbolic Names

Figure 13: Settings > Device Configuration

**Note 1:** The offset addresses set in this window are for the addressing of the input data and output data in the Master! These address settings (offsets) are not the settings in the DeviceNet device (Slave). The DeviceNet device (Slave) organizes its data itself.

**Note 2:** The input data and the output data from the bus are transferred directly to the dual-port memory in the Hilscher DeviceNet Slave. These offset addresses are related to the Master.

The EDS file which this device is using is displayed next to **File Name**.

The **Description** and **MAC ID** fields displays the entries made during the selection phase of the Slave device. Both entries can be set/changed here.

The checkbox **Activate Device in actual configuration** decides, if the Master tries to establish the communication to the Node or not. Disabling this button makes sense for example, if a Node is physically not present in the network but will be present in future. This suppresses unnecessary requests of the Master to non existing devices, but the device insertion itself reserves process data in the process data image of the Master.

#### 4.4.3.1 MAC ID (Device network address)

The network address of a device serves to distinguish itself on a DeviceNet fieldbus system from any other device or Slave on this network. This should be a unique number for each device. A valid MAC-ID address is within a range of 0 to 63 and can be re-entered and changed in the **MAC-ID** box in the **Device Configuration** Dialog.

#### 4.4.3.2 Actual chosen IO Connection

DeviceNet allows establishing several kinds of I/O connections between devices. Please note that a device has not to support all types of IO connections.

| I/O connection  |
|-----------------|
| Poll            |
| Bit Strobe      |
| Change of State |
| Cyclic          |

Table 11: Overview I/O Connections

Here is an explanation of the different connections types.

- Polled I/O Connection** - One poll command from the Master sends a number of output data to a single, specific device (point-to-point). The device receives (consumes) the poll command and processes the output data. If it has input data configured for this poll connection it reacts by sending (producing) back a number of input data and/or status information to the Master. Before a polled I/O connection is initiated by the Master, it reads the **Consumed** and **Produced Connection Size** of the data from the Slave first and compares each value with the internally configured one. If the Master detects differences the connection cannot be established. Sending a poll command can happen at any time the Master wants to and has timer or event dependencies. A device has to respond if it has consumed and understood the poll command request of the Master, even if it has no input data. Else the Master will report a timeout error. Polling data to many devices has the disadvantage that the network traffic rate is very high and most data which is transferred has not changed since the last transmission. Furthermore the higher the bus load more communication errors can occur if the bus is disturbed by external influences.

- **Bit Strobe I/O Connection** - Bit strobe command and response messages rapidly move small amounts of I/O data between the Master device and one/some/all Slave devices. The bit strobe message contains a bit string of 64 bits of output data, one output bit per possible device. Each bit in there is assigned to one device address in the network. Herewith this service has broadcast functionality that means more than one Slave device can be addressed by one command. Because all addressed Slave devices get this command at the same time, this command is normally used to synchronize data transfer to several Slave devices. A Slave device can take its corresponding output bit as a real output information to give it to the peripheral connections (e.g. an LED) and/or use the bit as a trigger to send back its input data with a poll response message. The data that can be sent back from each Slave after a bit strobe command was received is limited to 8 bytes in length. Bit strobe usage causes therefore a reduced bus loading than poll connections.
- **Change of State/Cyclic I/O Connection** - The Master device sends a number of output data to a single, specific device (point-to-point). Data production is triggered by either a determined changed value in the output data or the cyclic timer expiration. Depending on how the Slave behaviour is configured, the Slave can send back an acknowledge message, containing a number of input data and/or status information. The Slave device sends a number of input data to the Master, if the data is either changed or the cyclic timer has expired. The Master itself can acknowledge this message with output data if configured. Change of state only production of data hold down the bus load as small as possible, while data than can be transmitted as fast as possible by each device because bus conflicts are less possible. So you can get high performance data transmission with in comparison low baud rates.

#### 4.4.3.3 Connection Object Instance Attributes

The **Production Inhibit Time**, one for each connection, configures the minimum delay time between new data production in multiples of a millisecond. The timer is reloaded each time new data production through the established connection occurs. While the timer is running the device suppresses new data production until the timer has expired. This method prevents that the device is overloaded with too fast incoming requests.

The value 0 defines no inhibit time and data production can and will be done as fast as possible. If in polled mode for example a **Production Inhibit Time** of 1000dec is configured, then the poll request message to the device will be sent every second.

The **Expected Packet Rate**, one for each connection, is always transferred to the device before starting and doing the I/O transfer. The value is used by the device later to reload its 'Transmission Trigger' and 'Watchdog Timer'. The 'Transmission Trigger Timer' is used in a 'cyclic' I/O connection to control the time when the data shall be produced. Expiration of this timer then is an indication that the associated connection must transmit the corresponding I/O message. In 'change of state' connections the timer is used to avoid the watchdog timeout in this connection, when a production has not occurred since the timer was activated or reloaded.

---

**Note:** the **Production Inhibit Time** is verified against the **Expected Packet Rate**. If the Expected Packet Rate value is unequal zero, but less than the Production Inhibit Time value, then an error window is opened when pressing the OK button or changing to a wrong value.

---

The **Watchdog Timeout Action** defines the device behaviour when the watchdog timer in the device expires. The following values are defined and their functionality is closer described in the DeviceNet specification.

- **Transition to Time Out:** The connection transitions to the Timed Out state and remains in this state until it is Reset or Deleted.
- **Auto Delete:** The connection class automatically deletes the connection if it experiences an Inactivity/Watchdog timeout.
- **Auto Reset:** The connection remains in the established state and immediately restarts the Inactivity/Watchdog timer.

#### 4.4.3.4 UCMM Check

The UCMM Check box is used for modules that require the use of UCMM messaging format. Class 1, 2, and 3 are supported. Check the documentation for your Slave device to identify if this box must be checked.

#### 4.4.3.5 Fragmented Timeout

If a transmission of I/O data or explicit message is greater than 8 bytes in length, it must be transmitted on DeviceNet in a fragmented manner. The maximum time the Master will wait until a Slave has to respond during the fragmented transmissions is the fragmented timeout.

#### 4.4.3.6 Parameter Data

The button **Parameter Data** can be selected in the Device Configuration window to edit the parameter data.

If default parameters are configured in the EDS file for this Device, they are inserted automatically when the menu is chosen the first time.

Some of devices need some further parameterization data, to change for example a measurement limitation or a value range. These data is Node specific and their functionality can not be explained at this point.

The explanation can be normally found in the corresponding Node manual.

This window below shows an example of parameter data of a device.

**Parameter Data**

Available Parameters

Parameter access filter: all

| Obj. | Class | Inst. | Attr. | Type  | Access | Parameter Name             | Min  | Max  |
|------|-------|-------|-------|-------|--------|----------------------------|------|------|
| 0001 | 64    | 01    | 07    | UINT  | Ro     | Analog Output Data Length  | 0000 | FFFF |
| 0002 | 64    | 01    | 08    | UINT  | Ro     | Analog Input Data Length   | 0000 | FFFF |
| 0003 | 64    | 01    | 09    | UINT  | Ro     | Digital Output Data Length | 0000 | FFFF |
| 0004 | 64    | 01    | 0A    | UINT  | Ro     | Digital Input Data Length  | 0000 | FFFF |
| 0005 | 64    | 01    | 05    | USINT | Ro     | Coupler status             | 00   | FF   |

Help

Customized Parameters

| Class | Inst. | Attr. | Type | Parameter Name | Value |
|-------|-------|-------|------|----------------|-------|
| 01    | 0001  | 01    | UINT | new            | 00    |
|       |       |       |      |                |       |
|       |       |       |      |                |       |
|       |       |       |      |                |       |
|       |       |       |      |                |       |
|       |       |       |      |                |       |

Help

Buttons: OK, Cancel, Values, Decimal, Add, Delete, User Parameter

Figure 14: Settings > Device Configuration > Parameter Data

Two tables are available: one table with all available parameters and one table for customized parameters. These parameters can be selected from the available parameters and insert in the table Customized Parameters by clicking the **Add** button.

Selecting the button **User Parameter**, the following dialog opens:

#### 4.4.3.7 User Parameter

Own parameter data can be created in the dialog **User Parameter**.

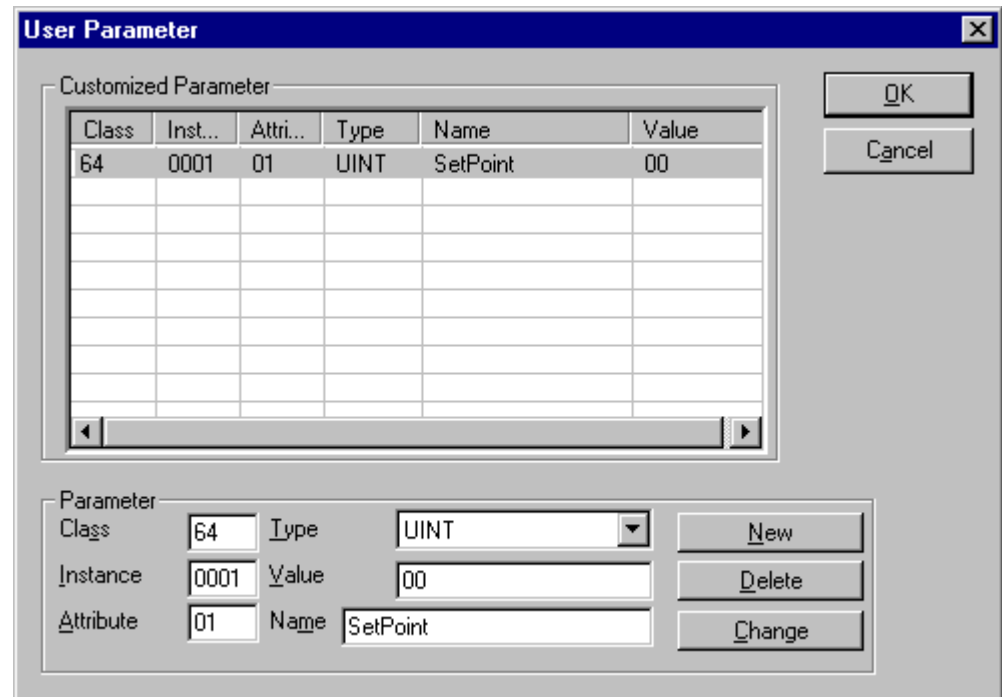


Figure 15: Parameter Data > User Parameter

To define own parameter data, the following parameter characteristics have to be set:

- **Class**
- **Instance**
- **Attribute**
- **Type**
- **Value**
- and **Name**

The parameter defined by the user are displayed in the table **Customized Parameter**.

It is possible to create further user parameter via the button **New**.

To delete user defined parameter, this parameter have to be selected in the table by clicking and via the **Delete** button these parameter data are deleted.

With the **Change** button it is possible to change still configured parameter data.

**Note:** The permissible values for user parameter are manufacturer specific. For further information about valid user parameter for the used device, please see the device description of the hardware manufacturer.

#### 4.4.3.8 Process Data Configuration

- Fixed I/O data transferred

DeviceNet handles I/O data transparent as a byte string without defining any data type in the transferred data. To be operative it defines only the number of bytes in consumed and produced direction that shall be transferred across a connection, nothing else. But SyCon and the firmware now allow assigning modular each byte or a bunch of bytes of the transparent string to different data types. A list of the supported data types of the connection can be found in the middle table of the window called **Available Predefined Connection Data Types**. This available predefined data types are in the EDS file of the respective device. If there are no predefined data types, SyCon defines the data types: Byte Array Input Data and Byte Array Output Data.

The following data types are supported:

- Bit, Byte, Word, Dword, Byte Array

If the data type **Byte Array** is chosen the number of bytes that shall be reserved for this data type can be entered in the **Data Count** column in the lower table. Any other data type has its fixed length that can not be changed. The data types are distinguished in process output and process input data in the view of the Master device.

A double-click on a predefined data type or a click in the **Add to configured I/O data** button will insert the chosen data type in the lower table called **Configured I/O connection data**. This table contains all data that shall be really transferred across the connection. SyCon will add separately the number of used bytes of each configured I/O data and forms the values **Consumed** and **Produced Connection Size** automatically. Both values indicate the sum of bytes which shall be sent by the Master as outputs (**Consumed** by the device) and received by the Master as inputs (**Produced** by the device).

- Assigning the process data offset addresses

The I/O offset addresses of each placed data type in the connection data table can be freely configured in a range of 0 to 3583 or they are set automatically by SyCon. To enable or disable free configuration use the flag **Auto Addressing** in the menu **Settings > Auto Addressing**. If enabled SyCon will place all configured I/O data space less in physical order one to each other based on the rising MAC-ID order. This is done during the download procedure. The assigned addresses can be checked then in the overview **Address Table** of the menu **View**. If the addresses are entered manually the default address 0 in the **input address** respectively the **output address** must be overwritten. Depending on the **Addressing mode** in the **DNM Master Settings** the addresses are byte addresses or word addresses. This is described in the chapter Addressing mode.

In case of manual addressing (that means auto addressing is deactivated) the configuration window looks like:

Figure 16: Settings > Device Configuration

In the column **I Addr** and **O Addr** you have to assign the addresses where to locate the data in the process image. Remember that these addresses correspond to your application on the HOST side.

If you deactivate a device in the actual configuration the device is shown like this:



To activate or deactivate a device in a configuration can be very useful for such devices which don't exist in the real physical network, but for which the I/O offset addresses shall be reserved or simply a symbolic 'missing device' wanted to be inserted as long the device is not connected.



## 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.

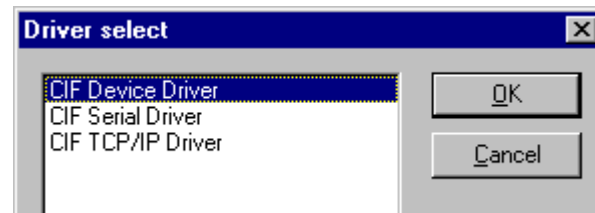


Figure 17: Driver Selection

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 that 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 | <i>CIF Device Driver</i> | 43   |
| CIF Serial Driver | <i>CIF Serial Driver</i> | 45   |
| CIF TCP/IP Driver | <i>CIF TCP/IP Driver</i> | 47   |

Table 12: 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

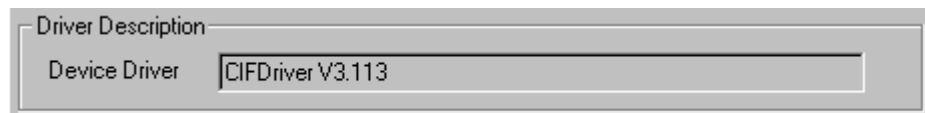


Figure 18: CIF Device Driver - Driver Description

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

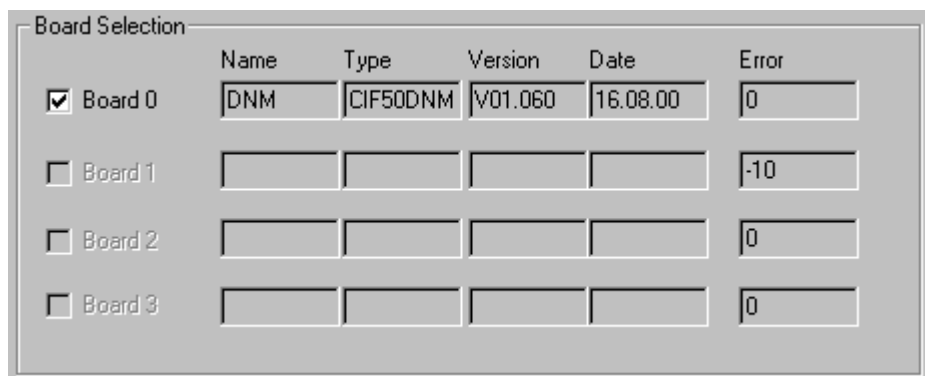


Figure 19: CIF Device Driver - Board Selection

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  |
|-------------------------------------|--|
| <input type="checkbox"/>            | Device is still not assigned and it can be selected.                                     |
| <input checked="" type="checkbox"/> | Device is assigned. The Assignment can be abrogated by deselecting.                      |
| <input type="checkbox"/>            | The assignment of the device is not possible.  |
| <input checked="" type="checkbox"/> | The device is still assigned in another open configuration and can not be selected here. |

Table 13: 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.

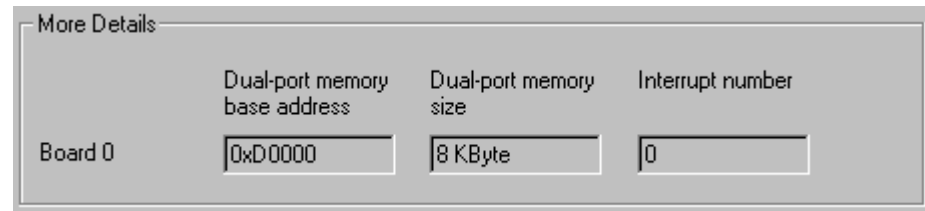


Figure 20: CIF Device Driver - More Details

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

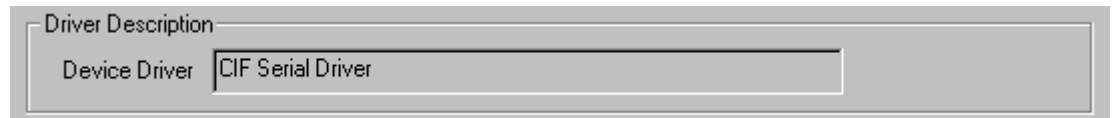


Figure 21: CIF Serial Driver - Driver Description

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

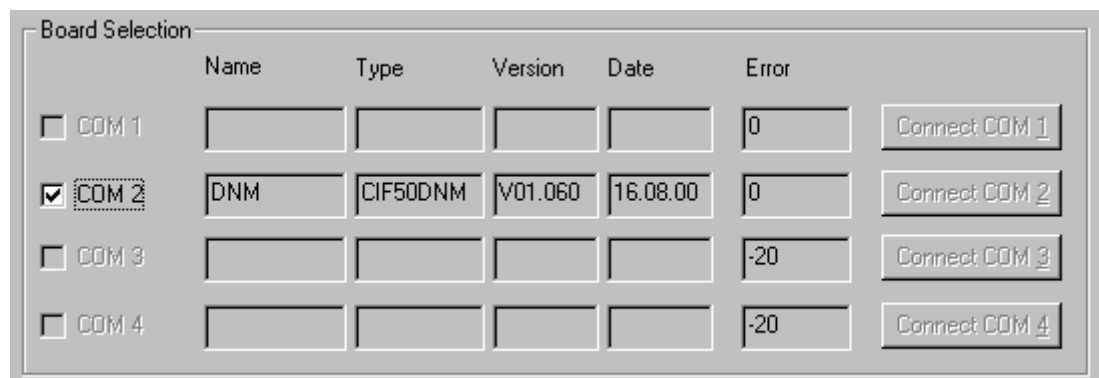


Figure 22: CIF Serial Driver - Board Selection

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  |
|-------------------------------------|--|
| <input type="checkbox"/>            | Device is still not assigned and it can be selected.                                     |
| <input checked="" type="checkbox"/> | Device is assigned. The Assignment can be abrogated by deselecting.                      |
| <input type="checkbox"/>            | The assignment of the device is not possible.  |
| <input checked="" type="checkbox"/> | The device is still assigned in another open configuration and can not be selected here. |

Table 14: 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 that 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 23: 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.

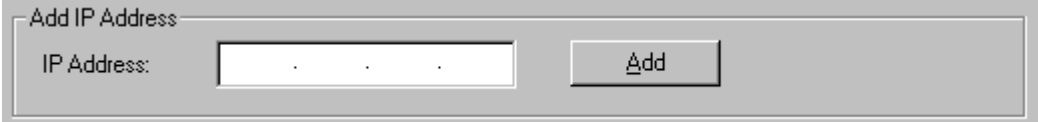


Figure 24: CIF TCP/IP Driver - Type in IP Address manually

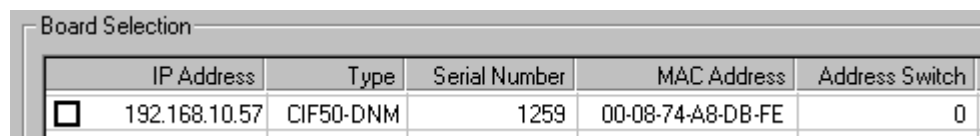
The IP address of the device to be connected need to be typed in the field **Add IP Address**. Clicking the **Add** button, it is tried to build up a CIF TCP/IP connection between the PC and the device.

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.



|                          | IP Address    | Type      | Serial Number | MAC Address       | Address Switch |
|--------------------------|---------------|-----------|---------------|-------------------|----------------|
| <input type="checkbox"/> | 192.168.10.57 | CIF50-DNM | 1259          | 00-08-74-A8-DB-FE | 0              |

Figure 25: 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 50.

### Connect Device

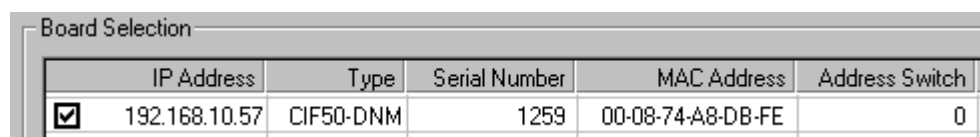
To connect a device to the PC, the checkbox of the favored device has to be selected in front of the **IP Address** field.

| Checkbox                            | Description   |
|-------------------------------------|---|
| <input type="checkbox"/>            | Device is still not assigned and it can be selected.                |
| <input checked="" type="checkbox"/> | Device is assigned. The Assignment can be abrogated by deselecting. |

Table 15: 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:



|                                     | IP Address    | Type      | Serial Number | MAC Address       | Address Switch |
|-------------------------------------|---------------|-----------|---------------|-------------------|----------------|
| <input checked="" type="checkbox"/> | 192.168.10.57 | CIF50-DNM | 1259          | 00-08-74-A8-DB-FE | 0              |

Figure 26: 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

| Filtered Device(s) |         |               |                   |                |  |
|--------------------|---------|---------------|-------------------|----------------|--|
| IP Address         | Type    | Serial Number | MAC Address       | Address Switch |  |
| 192.168.10.161     | NN40/42 | 5             | 00-02-A2-0A-00-05 | 0              |  |
| 192.168.10.155     | NL-MPI  | 13            | 00-02-A2-0C-00-0D | 0              |  |
| 192.168.10.160     | NN40/42 | 11            | 00-02-A2-0A-00-0B | 0              |  |

Figure 27: 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:

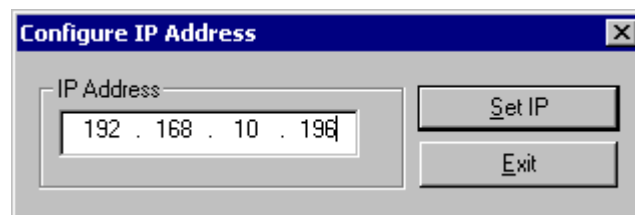


Figure 28: Set IP Address

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 Bus Parameter

In the menu **Settings > Bus Parameter** the basic settings for the DeviceNet network must be done.

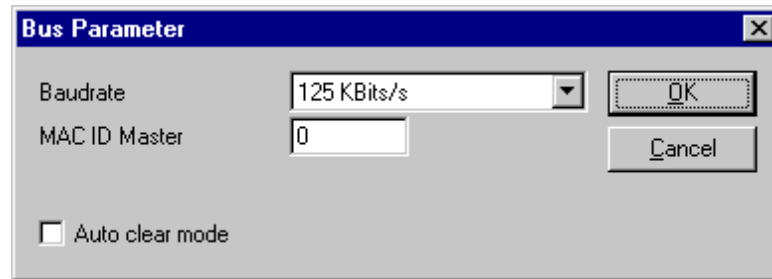


Figure 29: Settings > Bus Parameters

Mainly, this concerns the determination of the **Baudrate**. The DeviceNet board supports the baud rates 125kbit/s, 250kbit/s and 500kbit/s. Normally DeviceNet components use the auto baud detection to get the Baudrate automatically once.

Furthermore you can assign the **MAC ID** for the Master in the window Bus Parameter.

The **Auto Clear mode** feature defines the behaviour of the Master if the communication breaks down or is interrupted to a Node. If the flag **Auto clear mode ON** is activated, then the Master will also stop the communication to all further Nodes which were still responding and active. If the flag Auto clear mode is not activated, then a lost communication contact to one Node has no influence on the communication channel of the still present ones. For all the error effected Nodes the Master remains in the state to try the reestablishment of the communication again.

## 5.3 DeviceNet Master

### 5.3.1 DeviceNet Master Settings

To enter the DeviceNet Master Settings set the focus on the Master (left mouse click) and select the menu **Settings > Master Settings** or set the focus on the Master (left mouse click) and use the right mouse button at the DeviceNet Master device or simply double-click on the DeviceNet Master device.

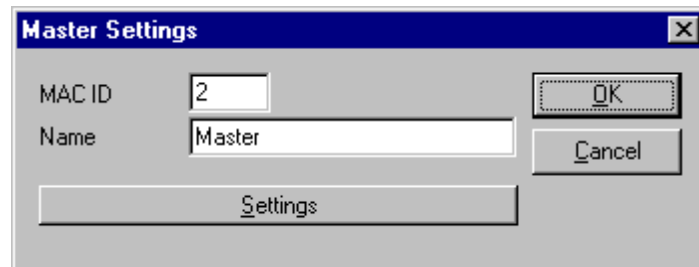


Figure 30: DeviceNet Master Settings

In the now opened dialog the **Name** and the **MAC ID** of the Master device can be changed.

Via the **Settings** button a further dialog with DeviceNet Master Settings opens. These settings contain parameters which define the behaviour of the device on its user interface which is not a part of the DeviceNet configuration directly. This menu is only valid for Hilscher devices, and is downloaded with the DeviceNet configuration.

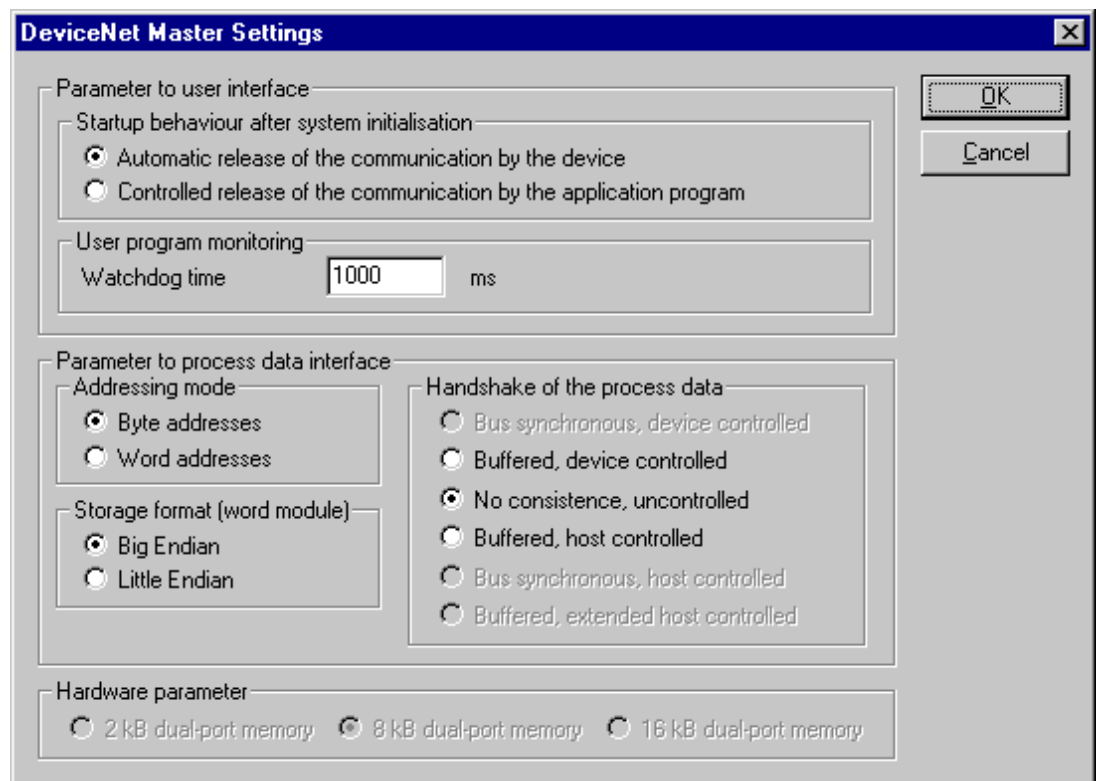


Figure 31: Settings > Master Settings

- **Startup behaviour after system initialization**

If **Automatic release of the communication by the device** is selected, the Master starts with the data transfer on the bus when initialization is finished. If **Controlled release of the communication by the application program** is selected, the user has to start the data transfer on the bus, by a defined release procedure.

- **User program monitoring**

The watchdog time appointed defines how long the device will wait for a user trigger of the software watchdog if started once, until it resets all outputs of the device to zero. This procedure must be activated by the user application and is not started automatically.

---

**Note:** This is not a special DeviceNet function.

---

An example for this can be a Soft PLC.

- **Addressing mode**

The addressing mode of the process data defines how to interpret the addresses of the process image. Possibilities are **Byte addresses** or **Word addresses**. See details next page.

- **Storage format (word module)**

The storage format fixes the format, how the data is placed and interpreted in the process images. For the data type word the **Little Endian** format and **Big Endian** format can be selected.

- **Handshake of the process data**

With these different modes the handshake of the process data is selected for the Master. The selection of this mode is important for the correct data exchange between the application and the device. Please refer to the tool kit or the device driver manual for the detailed description of these modes. If foreign user programs or drivers are used, like S5 for Windows for example, refer to the delivered manual to choose the right mode.

- **Hardware parameter**

With this parameter the size of the dual-port memory of the hardware is selected. The parameter will enlarge or reduce the possible value ranges for the I/O offsets.

In case of CANopen Master Cards the dual-port memory size is preset to 8K and cannot be changed by the user, thereby 7K are process data.

### 5.3.2 Auto Addressing

If this option is selected, SyCon allocates the process data addresses in physical order. If this option is not selected, the addressing of the process data has to take place manually. Therefore also see section *Process Data Configuration* on page 39.

### 5.3.3 Addressing Mode

The addresses in the configuration of the device define the starting point of the data in the process data image memory. This can work in a Word or Byte oriented method by means of the **Addressing mode** parameter.

| Addresses      | Meaning   |
|----------------|---|
| Byte addresses | The process depiction has a Byte structure and each Byte has its own address. |
| Word addresses | The process depiction has a Word structure and each Word has its own address. |

Table 16: Addressing Mode

This has nothing to do with the physical size of the Dual-port memory – this is always Byte-oriented! When the application makes a Word access, it is automatically divided by the PC into two sequential Byte accesses.

The following table shows the different storing of the various data types in the Byte- or Word-oriented process data image:

| IEC address in Byte mode | IEC address in Word mode | Offset address in the Dual-port memory | Data in the Process image | Output to an I/O module   |
|--------------------------|--------------------------|--|---------------------------|---|
| QB 0                     | QB 0                     | 0                                      | 0000 0000                 |   |
| QB 1                     |                          | 1                                      | 0000 0000                 |   |
| QB 2                     | QB1                      | 2                                      | 1110 0010                 | Output of QB2 / QB1 to a single Byte module:<br>D7 D6 D5 D4 D3 D2 D1 D0<br>1 1 1 0 0 0 1 0  |
| QB 3                     |                          | 3                                      | 0000 0000                 |   |
| QB 4<br>QB 5             | QB2                      | 4<br>5                                 | 1111 1000<br>0000 0111    | Output of two Bytes beginning from QB4 / QB2 to a module that is defined as a Byte module with the data count 2 (no differentiation between the two memory formats as the data are of Byte type):<br>D7 D6 D5 D4 D3 D2 D1 D0 D7 D6 D5 D4 D3 D2 D1 D0<br>1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 |
| QW 6                     | QW3                      | 6<br>7                                 | 1111 1111<br>0100 0100    | Output of QW6 / QW3 in the data format lower/higher value Byte:<br>D15 D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0<br>0 1 0 0 0 1 0 0 0 1 1 1 1 1 1 1   |

Table 17: Example for data in the process data image

The following table is meant to clarify the method of addressing:

| Byte Addressing |      |      | Word Addressing |      |      |
|-----------------|------|------|-----------------|------|------|
| Byte 0          | IB 0 | IW 0 | Word 0          | IB 0 | IW 0 |
| Byte 1          | IB 1 |      |                 | -    |      |
| Byte 2          | IB 2 | IW 2 | Word 1          | IB 1 | IW 1 |
| Byte 3          | IB 3 |      |                 | -    |      |
| Byte 4          | IB 4 | IW 4 | Word 2          | IB 2 | IW 2 |
| Byte 5          | IB 5 |      |                 | -    |      |

Figure 32: Image of the method of addressing for input

| Byte Addressing |      |      | Word Addressing |      |      |
|-----------------|------|------|-----------------|------|------|
| Byte 0          | QB 0 | QW 0 | Word 0          | QB 0 | QW 0 |
| Byte 1          | QB 1 |      |                 | -    |      |
| Byte 2          | QB 2 | QW 2 | Word 1          | QB 1 | QW 1 |
| Byte 3          | QB 3 |      |                 | -    |      |
| Byte 4          | QB 4 | QW 4 | Word 2          | QB 2 | QW 2 |
| Byte 5          | QB 5 |      |                 | -    |      |

Figure 33: Image of the method of addressing for output

## 5.4 Device (Slave)

### 5.4.1 Device Configuration

To call up the menu Device Configuration, set the focus at the device (left mouse click) and select the menu **Settings > Device Configuration**

or

Make a double-click on the device icon will open the **Device Configuration** window. The section *Device Configuration* on page 33 describes the configuration.

### 5.4.2 Device Settings

The DeviceNet Slave Settings contain parameters which defines the behaviour of the Slave device on its user interface which is not a part of the DeviceNet configuration. This menu is only valid for Hilscher devices, which can be directly downloaded with the DeviceNet configuration.

To enter the DeviceNet Slave settings set the focus on the Slave and open the window with the menu **Settings > Device Settings**.

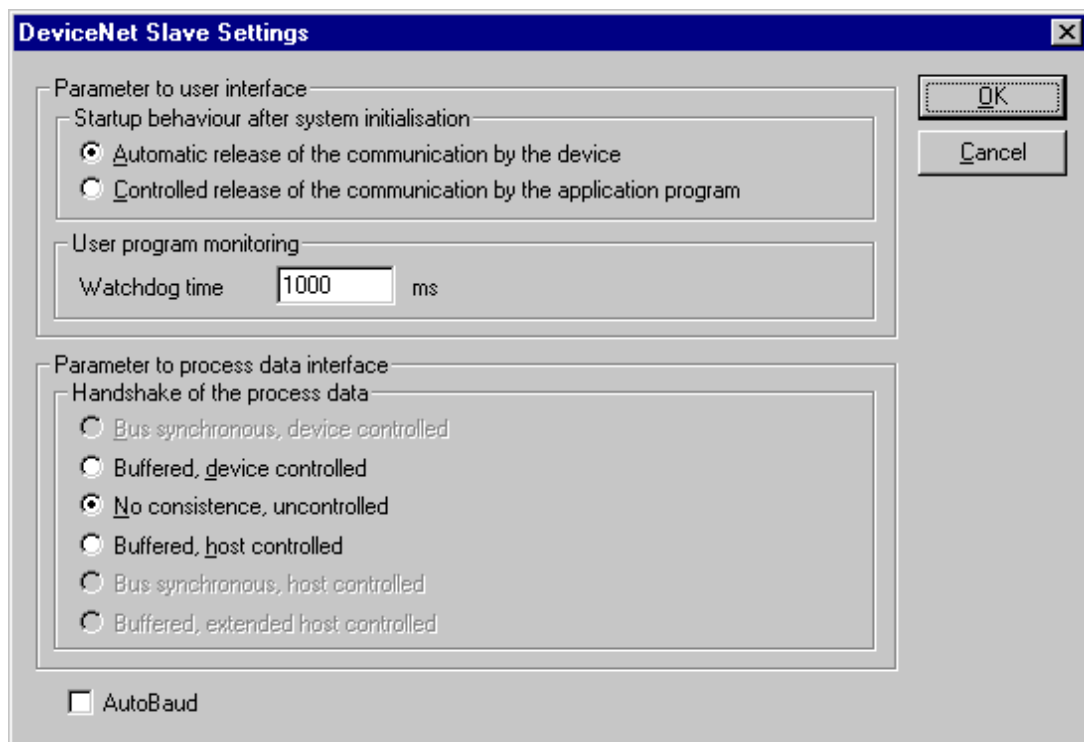


Figure 34: Settings > Device Settings

- **Startup behaviour after system initialization**

If **Automatic release of the communication by the device** is selected, the Slave is ready to communicate with the Master when started. If **Controlled release of the communication by the application program** is selected, the user has to release communication by a defined release procedure.

- **User program monitoring**

This function is not supported by the DeviceNet Slave.



- **Handshake of the process data**

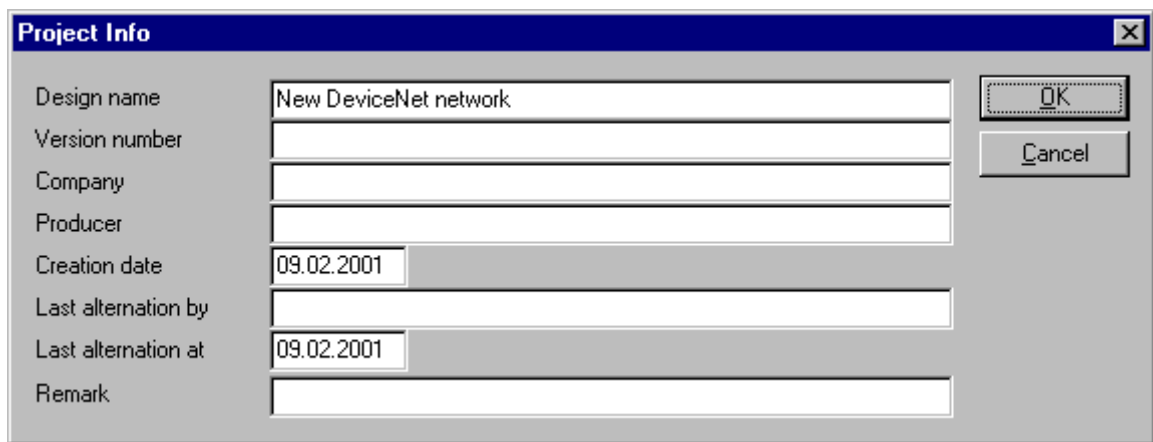
With these different modes the handshake of the process data is selected for the Slave. The selection of this mode is important for the correct data exchange between the application and the device. Please refer to the tool kit or the device driver manual for the detailed description of these modes.

- **Autobaud On**

Activates or deactivates the automatic baudrate detection of the Hilscher DeviceNet Slave.

## 5.5 Project Information

If the user creates its own 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.



| Project Info        |                       |
|---------------------|-----------------------|
| Design name         | New DeviceNet network |
| Version number      |                       |
| Company             |                       |
| Producer            |                       |
| Creation date       | 09.02.2001            |
| Last alternation by |                       |
| Last alternation at | 09.02.2001            |
| Remark              |                       |

Figure 35: Settings > Project Information

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

## 5.6 Path

In the menu **Settings > Path** the path directory of the EDS files is shown (EDS File directory). The default value is C:\Programme\Hilscher\SyCon\Fieldbus\DEVNet\EDS

The path for Project directory defines the path where the project specific files are stored.

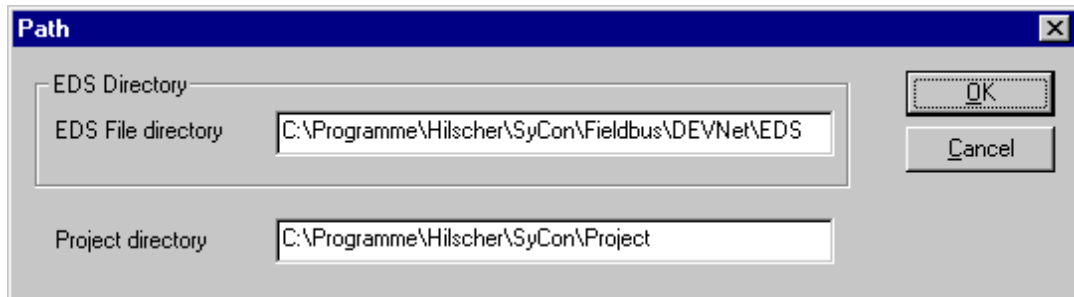


Figure 36: Settings > Path

If you click the **OK** button all EDS files are read in.

## 5.7 Language

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

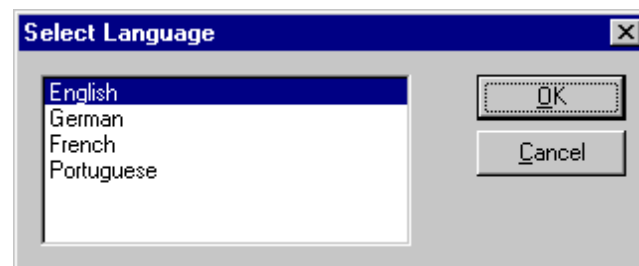


Figure 37: Settings > Language

Here one is in a position 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 fieldbuses!

---

## 5.8 Start Options

After activating the **Settings > Start Options** menu point in the network mode (you can change to the network mode by choosing **Window > Logical Network View**), the following dialog will appear.

Here it is possible to set the various starting options or modes. Some are of importance only for the OPC-Server operation.

The important ones are given below.

| Start Options  |  |                |      |                 |  |                 |  |                 |  |                 |  |
|--|--|----------------|------|-----------------|--|-----------------|--|-----------------|--|-----------------|--|
| <input type="checkbox"/> Simulation mode ON/OFF  | <input checked="" type="checkbox"/> Auto connect ON/OFF            |                |      |                 |  |                 |  |                 |  |                 |  |
| <input type="checkbox"/> Start SyCon hidden if started via OPC   | <input checked="" type="checkbox"/> Send Message only when changed |                |      |                 |  |                 |  |                 |  |                 |  |
| <input type="checkbox"/> Start SyCon next Time with last Configuration   | <input type="checkbox"/> Message transfer synchronous              |                |      |                 |  |                 |  |                 |  |                 |  |
| <input type="checkbox"/> Logical Network View visible  |  |                |      |                 |  |                 |  |                 |  |                 |  |
| Fast start options   |  |                |      |                 |  |                 |  |                 |  |                 |  |
| <input type="checkbox"/> Fast start ON/OFF   |  |                |      |                 |  |                 |  |                 |  |                 |  |
| MSG tracer options   |  |                |      |                 |  |                 |  |                 |  |                 |  |
| <input type="checkbox"/> MSG tracing ON/OFF  |  |                |      |                 |  |                 |  |                 |  |                 |  |
| <input checked="" type="checkbox"/> Start with multiple configurations   |  |                |      |                 |  |                 |  |                 |  |                 |  |
| <table border="1"> <thead> <tr> <th>Product</th> <th>Code</th> </tr> </thead> <tbody> <tr> <td>License</td> <td>----</td> </tr> </tbody> </table>  |  | Product        | Code | License         | ----   |                 |  |                 |  |                 |  |
| Product  | Code   |                |      |                 |  |                 |  |                 |  |                 |  |
| License  | ----   |                |      |                 |  |                 |  |                 |  |                 |  |
| <table border="1"> <thead> <tr> <th>Configurations</th> <th></th> </tr> </thead> <tbody> <tr> <td>Configuration 1</td> <td>e:\programme\hilscher\sycon\project\devicenet.dn</td> </tr> <tr> <td>Configuration 2</td> <td></td> </tr> <tr> <td>Configuration 3</td> <td></td> </tr> <tr> <td>Configuration 4</td> <td></td> </tr> </tbody> </table> |  | Configurations |      | Configuration 1 | e:\programme\hilscher\sycon\project\devicenet.dn | Configuration 2 |  | Configuration 3 |  | Configuration 4 |  |
| Configurations   |  |                |      |                 |  |                 |  |                 |  |                 |  |
| Configuration 1  | e:\programme\hilscher\sycon\project\devicenet.dn                   |                |      |                 |  |                 |  |                 |  |                 |  |
| Configuration 2  |  |                |      |                 |  |                 |  |                 |  |                 |  |
| Configuration 3  |  |                |      |                 |  |                 |  |                 |  |                 |  |
| Configuration 4  |  |                |      |                 |  |                 |  |                 |  |                 |  |

Figure 38: 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 selected the last saved configuration in the SyCon is automatically loaded when the SyCon is started again.

- **Logic Network View visible**  
When this is selected 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.
- **MSG 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.
- **Send Message only when changed**  
Only valid for the OPC Server.
- **Message transfer synchronous**  
Only valid for the OPC Server.
- **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 are described that directly affect the Hilscher DeviceNet device, e.g. CIF 30-DNM, CIF 30-DNS, CIF 104-DNM.

**Note:** Please note that this will also permit interruption of a running communication or that outputs can be switched ON or OFF.

### 6.2 Online to the CIF

#### 6.2.1 Downloading the Configuration

First, the desired 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/COM/PKV devices must be carried out on the **Online > Download** menu. A warning will appear that the communication on the DeviceNet will be interrupted. This warning must be confirmed.

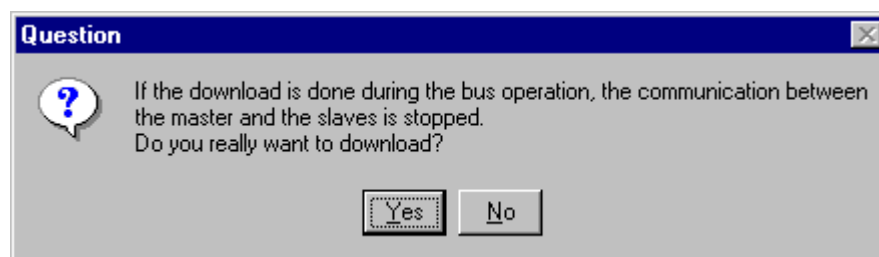


Figure 39: Security question before Download

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

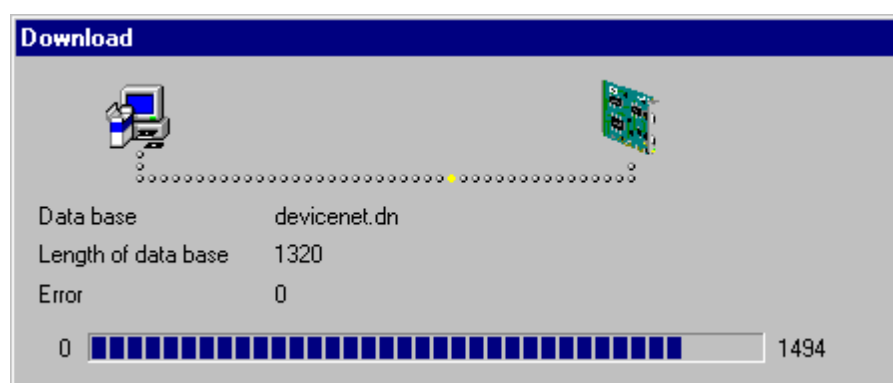


Figure 40: Online > Download

Before the download is executed, the configuration is tested by the Configurator. The most common cause of error is overlapping of addresses in the process data image. This can be checked by calling up the address table with the **View > Address Table** menu point.

If the issue of addresses in the process data image should be carried out automatically, then the **Auto Addressing** button in the **Master Configuration** window must be activated.

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

After the download, the device carries out an internal restart and begins with the communication if in **DeviceNet Master Settings** the **Automatic Release of Communication by the Device** menu point has been set.

## 6.2.2 Firmware Download

If a Firmware download is to be carried out, proceed as follows: first the desired device for Firmware downloading must be chosen in that the symbol of the device is selected with a left mouse click. Then, call up the **Online > Firmware Download** menu. Select the new Firmware and retrieve it with **Download** into the device. The Firmware is now retrieved.

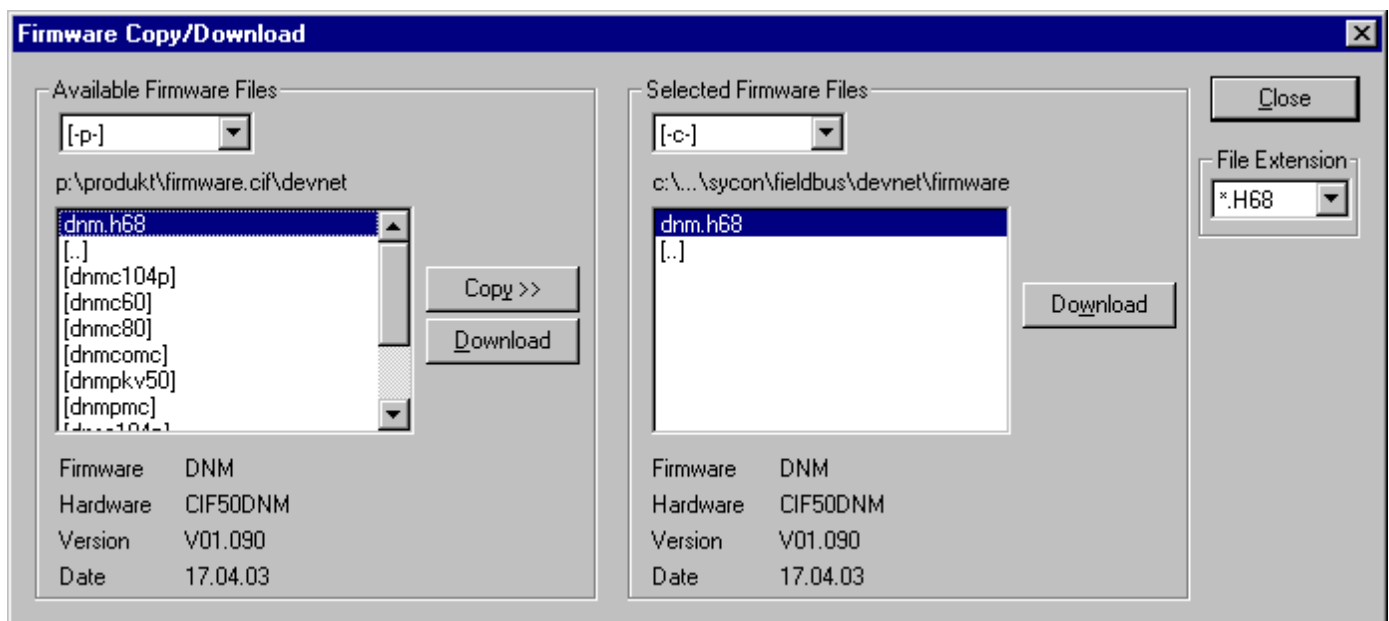


Figure 41: Online > Firmware Download

### 6.2.3 Firmware / Reset

First the desired 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 the version of the Firmware are displayed.

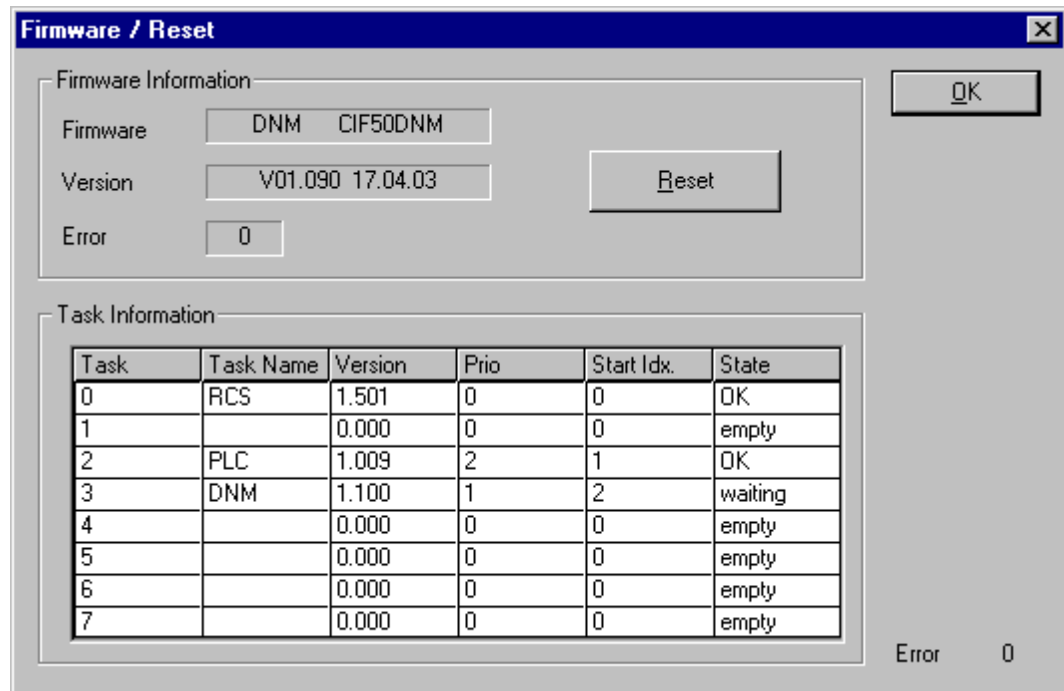


Figure 42: Online > Firmware / Reset

Click the button **Reset** to reset the device.

### 6.2.4 Device Info

First the desired 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, the device number and the serial number of the device is retrieved and shown.

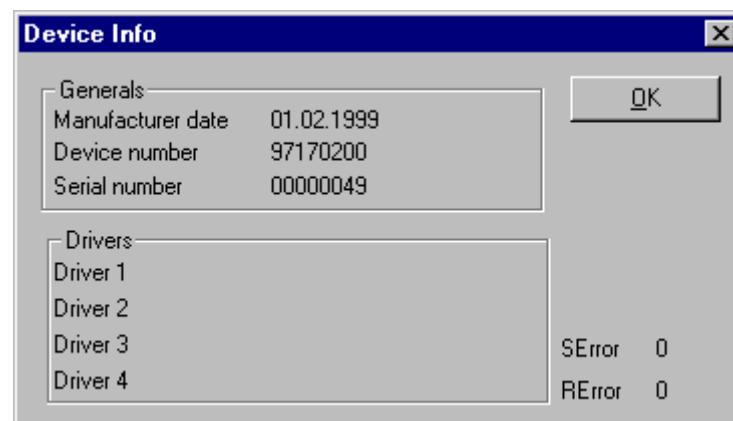


Figure 43: Online > Device Info

## 6.2.5 Activate Driver - Driver Licensing

First the desired device must be chosen with a left mouse click on the symbol of the device. Then select the **Online > Activate Driver** menu.

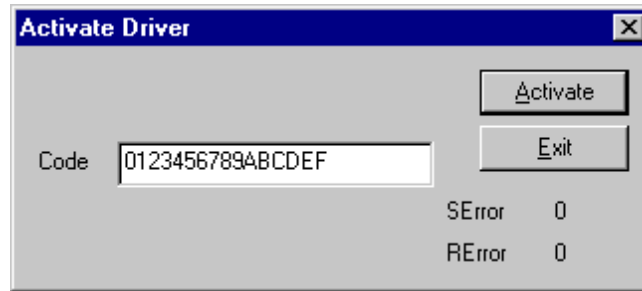


Figure 44: Online > Activate Driver

---

**Note:** The code 01234567890ABCDEF is not a valid code and is only shown as an example.

---



## 6.3 Automatic Network Scan

After the Master device is configured, it is possible to scan the DeviceNet network for other devices (automatic network scan). This allows a very fast configuration, and you also have the possibility to change detailed parameters for these devices later.

To start an automatic network scan, please proceed as followed:

1. Create a new project: Select the menu **File > New** and **DeviceNet**.
2. Select a Master: The Master is selected with the menu **Insert > Master**
3. Click on **Settings > Bus parameters** and select the baudrate and the MAC ID from the Master (explained in section *Bus Parameter* on Page 51)
4. Select **Online > Download** to load these settings into the DeviceNet Master.
5. Save: With the menu **File > Save** the so far set settings are saved.
6. Click on the Master and choose **Online > Automatic Network Scan**

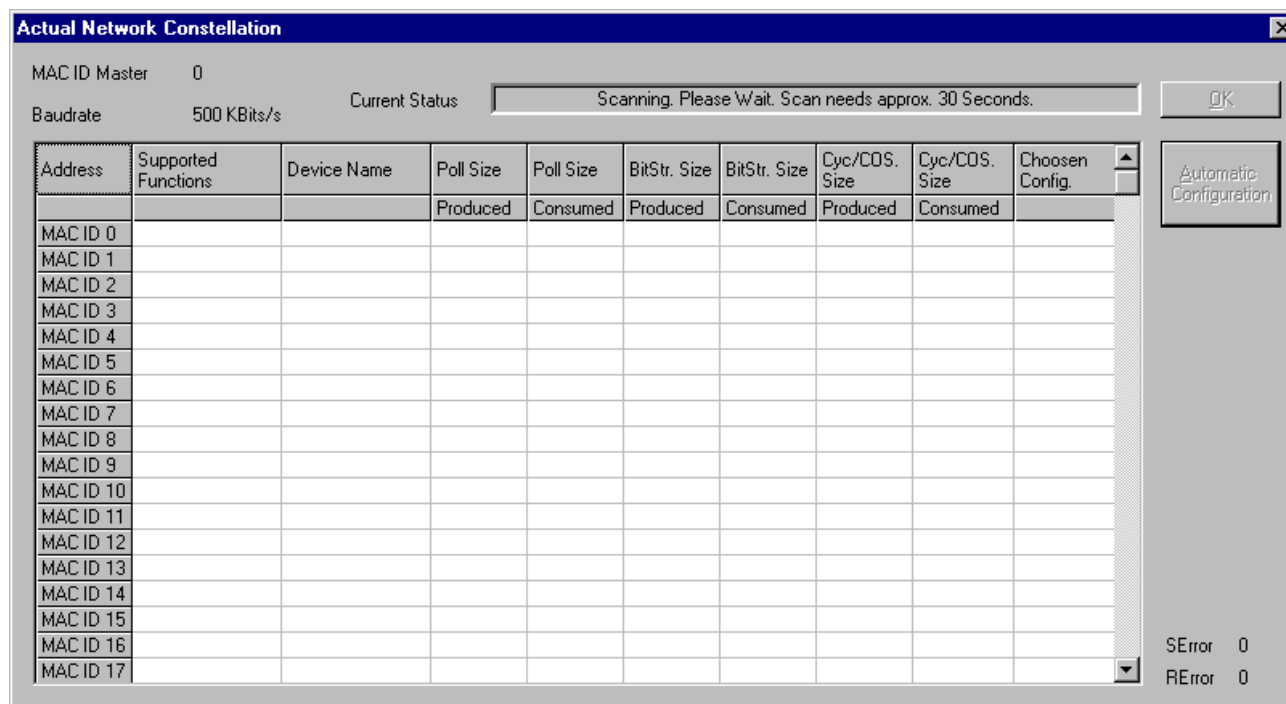


Figure 45: Online &gt; Automatic Network Scan (during scan)

The network scan will take approx. 30 seconds. The network scan is still in progress and could not be interrupted until the status, shown in the field “Current Status”, is “Ready!”. When the scan is done, you see the devices that were found at the corresponding MAC ID address in the table.

The master reads the following objects from the DeviceNet device (Slave) during the scan:

| Element                | Class.Instance.Attribute |
|------------------------|--------------------------|
| Device Name            | 1.1.7                    |
| Poll Size Produced     | 5.2.7                    |
| Poll Size Consumed     | 5.2.8                    |
| BitStr. Size Produced  | 5.3.7                    |
| BitStr. Size Consumed  | 5.3.8                    |
| Cyc/COS. Size Produced | 5.4.7                    |
| Cyc/COS. Size Consumed | 5.4.8                    |

Table 18: Read Classes.Instance.Attribute during the network scan

Example:

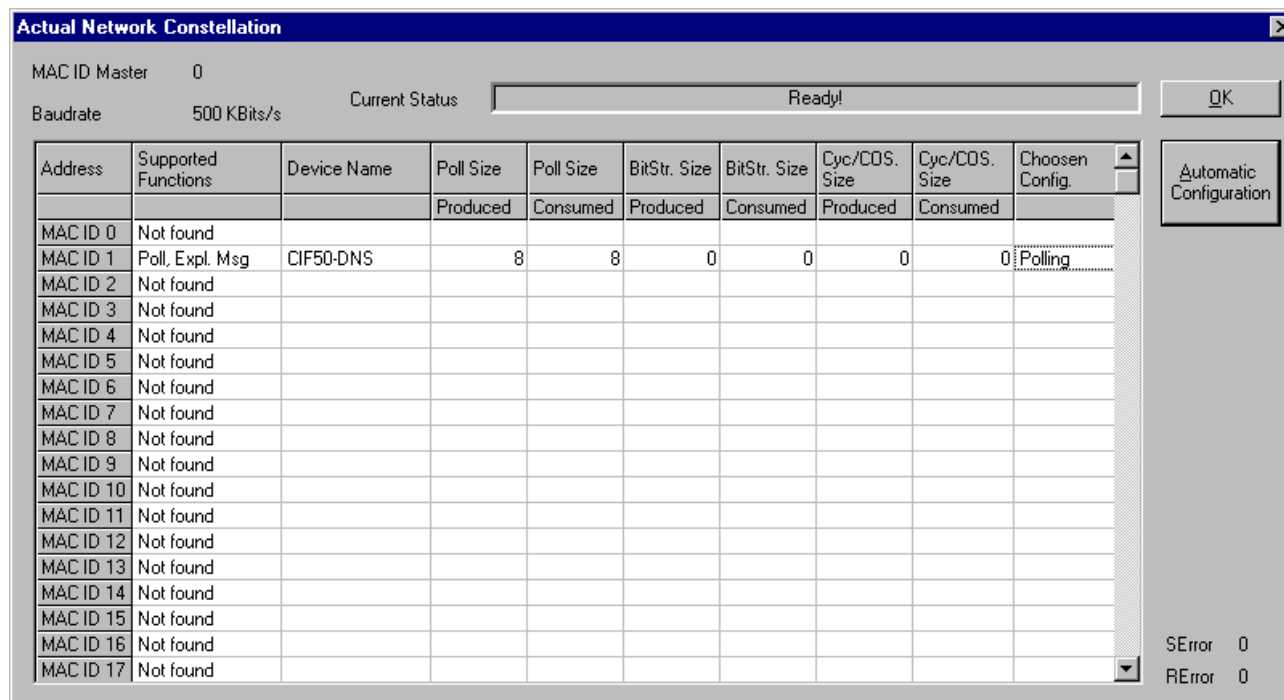


Figure 46: Online > Automatic Network Scan (after scan)

In our example in the figure above the automatic network scan detected a Hilscher device at MAC ID 1.

Here is an explanation of the different columns:

| Variable               | Meaning   |
|------------------------|---|
| Supported functions    | Functions supported by the device, could be polled, bit strobe or cyclic/change of state (see explanation in <i>Actual chosen IO Connection</i> , page 34)  |
| Device Name            | Name of the device, result from network scan  |
| Poll Size Produced     | Number of data for poll connection (input)  |
| Poll Size Consumed     | Number of data for poll connection (output)   |
| BitStr. Size Produced  | Number of data for bit strobe connection (input)  |
| BitStr. Size Consumed  | Number of data for bit strobe connection (output)   |
| Cyc/COS. Size Produced | Number of data for cyclic/COS connection (input)  |
| Cyc/COS. Size Consumed | Number of data for cyclic/COS connection (output)   |
| Chosen Config          | Configuration chosen by the user could be Change of State, Cyclic, Polling, Bit strobed or explicit only and depends on the functions supported by the device. Click on the cell to change the configuration. |

Table 19: Meaning of the columns in the automatic network scan

A double click on the first or second column of the corresponding row of the device shows information of the device

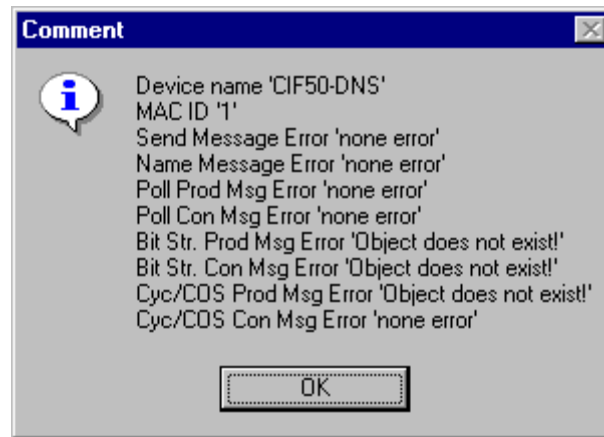


Figure 47: Information on a device in the automatic scan window

If you want to use this configuration as your configuration, click on **Automatic Configuration** and select **Yes** when prompted. Afterwards you can close the Automatic Configuration Window by clicking on **Close**. If you do not want the devices that were found in your configuration, just click **Close**.

If you want to insert your devices manually, please go on with section Insert Device (Slave) on page 31.

## 6.4 Start/Stop Communication

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

## 6.5 Diagnostic Functions

The following table shows Diagnostic Functions and their using for

- Hilscher DeviceNet Master devices
- Hilscher DeviceNet Slaves

| Diagnostic Functions              | Using  | Usable for Hilscher DeviceNet Masters devices | Usable for Hilscher DeviceNet Slave devices |
|-----------------------------------|--|---|---|
| <i>Live List</i>                  | Detects, which devices are connected to the Hilscher DeviceNet Master          | Yes   | No, only for Hilscher Master devices        |
| <i>Debug Mode</i>                 | Detects, to which DeviceNet Slaves the Master has a connection                 | Yes   | No, only for Hilscher Master devices        |
| <i>Global State Field</i>         | State information of the Hilscher DeviceNet Master                             | Yes   | No, only for Hilscher Master devices        |
| <i>Extended Device Diagnostic</i> | Statistic information and state information from the Hilscher DeviceNet device | Yes   | Yes   |

Table 20: Overview Diagnostic Functions

## 6.5.1 Live List

If you select the menu **Online > Live List** you will get an overview of all devices physically present in the actual network constellation. Present devices are drawn in black; all other non present devices are drawn in grey. The live list works online. If you connect or disconnect one station you will see the result as soon as SyCon collects the latest live list from the Master board. Remember that all devices on DeviceNet have to proceed the autobaud detection phase first to get wholly run. This can take up to some milliseconds.



Figure 48: Online > Live List

### 6.5.1.1 Change MAC-ID

The window to change a MAC-ID opens with a double click on the address (MAC-ID) of a DeviceNet device (slave).

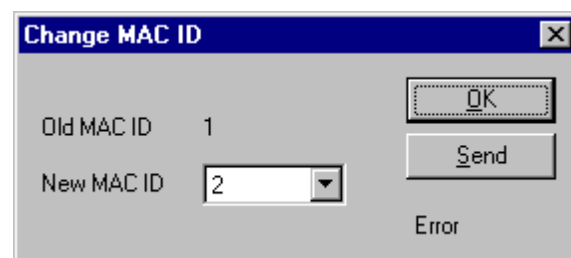


Figure 49: Online > Live List > Change MAC-ID

Select the new MAC-ID in the **New MAC-ID** field and click on the **Send** button to send it to the DeviceNet device (slave).

**Note:** The DeviceNet device (slave) has to support this function if this function should be used.

## 6.5.2 Debug Mode

First you have to select the Master device by clicking on it. Then select the menu item **Online > Start Debug Mode**. The System Configurator cyclically interrogates the status of the network communication on the Hilscher device and the individual condition of the devices.

To end the Debug Mode select the menu **Online > Stop Debug Mode**.

### 6.5.2.1 The Debug Window

When started the debug session the configuration window changes into the debug window. The devices and the line between them are displayed in green or red colour depending on the established network communication.

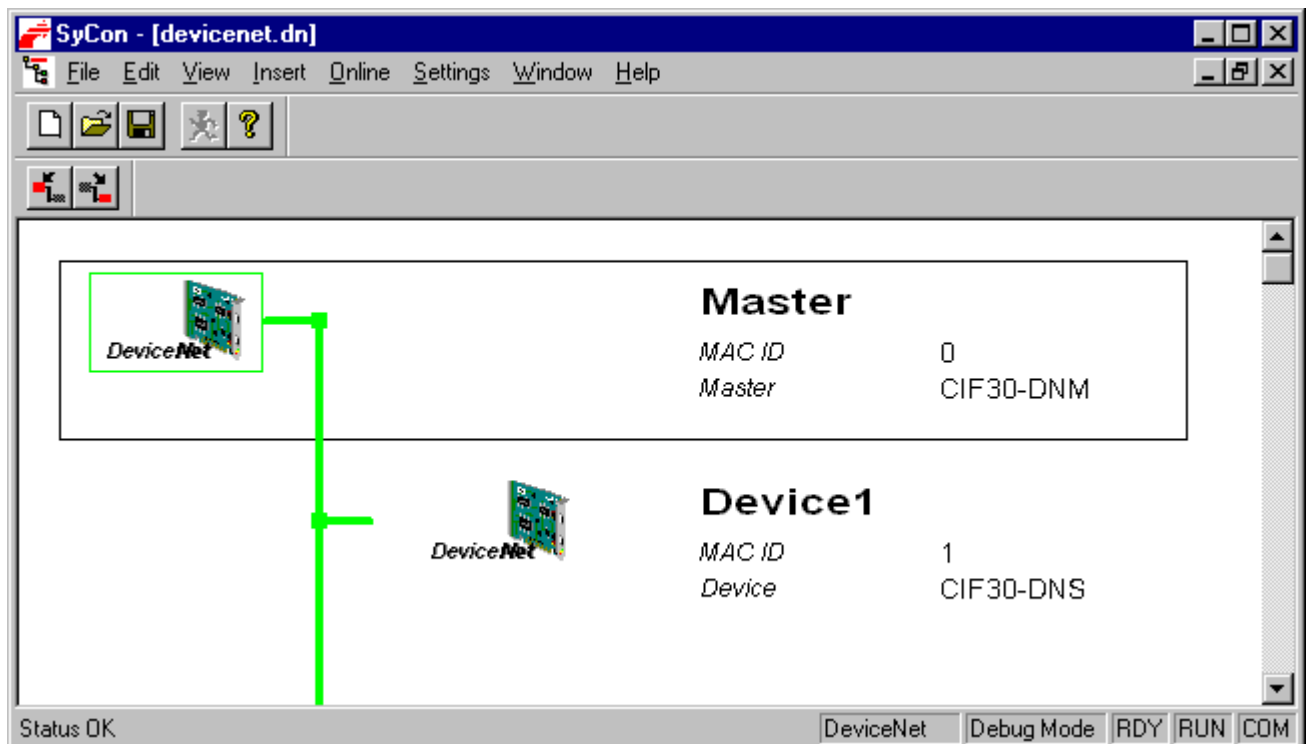


Figure 50: The Debug window

If diagnostic information is available for a specific device, next to the device icon the text **Diag** appears in red. To get further device specific diagnostic information then double-click on the device itself or set the focus to the device and select **Online > Device Diagnostic**.



### 6.5.2.2 Device Diagnostic

After the debug started from this time SyCon requests the status of all devices from the Master. If there is an error on a device the bus line to this Slave is drawn in red colour otherwise it is green. SyCon also displays the letters **Diag**, if the device signals diagnostic information. This information is displayed closer if you click on the device with the mouse. To activate the debug mode select the menu **Online > Start Debug Mode**. The menu **Online > Device Diagnostic** activates the DeviceNet device diagnostic. To end the Debug Mode select the menu **Online > Stop Debug Mode**.

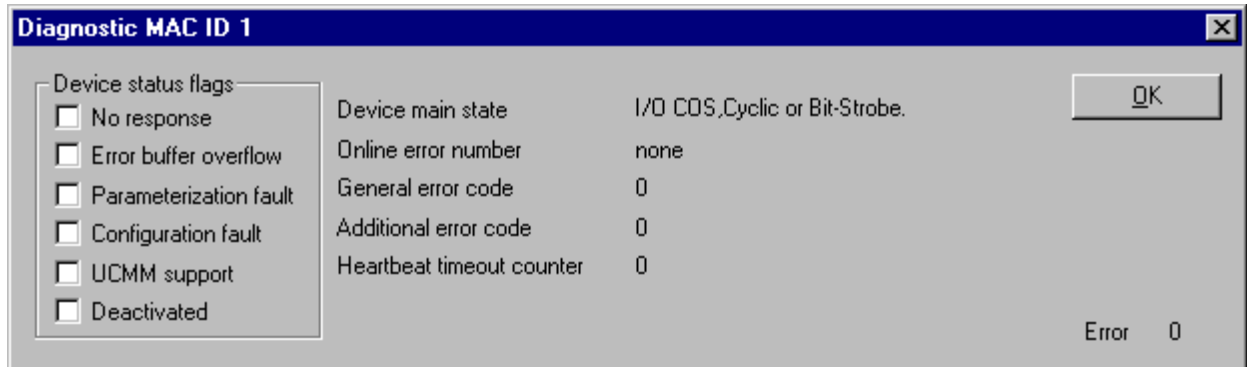


Figure 51: Online > Device Diagnostic

The individual bits in the **Device Diagnostic** have the following meaning:

| Device status flags in the Device Diagnostic | Meaning  |
|--|--|
| No response                                  | The Device is configured but is not present in the network. Please check the physical connection between the Master and this Node. Check also the chosen baudrate and if this baudrate is supported by this device.  |
| Error Buffer Overflow                        | DeviceNet defines a special reserved error channel for each Slave with high priority to give each Node the possibility to report emergency messages triggered by the occurrence of a device internal fatal error situation. The emergency message of each Node is collected in an internal buffer of a limited size. In this case the buffer overflow event is reported. |
| Parameterization Fault                       | The Master compares the configured Device <b>Profile</b> and the corresponding <b>Device Type</b> value of the <b>Device Configuration</b> window with the real physically present ones in the Slave by reading out the Slave configuration object. If the Master detects differences between the values it will report the Parameterization Fault.                      |
| Configuration Fault                          | A configuration fault will occur if a difference is seen between the configured Produced/Consumed data size and the actual Slave Produced/Consumed data size.  |
| UCMM Support                                 | The box will be checked if the Slave device requires UCMM support.   |
| Deactivated                                  | This bit is set by the Master automatically, if the Node state was configured to <b>Deactivate Device in actual configuration</b> in the <b>Device Configuration</b> window.   |

Table 21: Meaning of the bits in the Device Diagnostic

### 6.5.3 Global State Field

First you have to select the Master device by clicking on it. With the menu option **Online > Global State Field** opens a window in that cyclically statistic about the bus status and attached devices is put out.

**Global state field**

Online master main state    OPERATE

Collective status bits    PDUP DMAC NRDY EVE FAT NEXC ACLR CTRL

Collective online error location and corresponding error

Error at remote address    0    dec

Corresponding error event    (none)

Statistic bus information

Counter of detected bus off reports    2    dec

Counter of rejected telegram transmissions    0    dec

Device specific status bits

Parameterized Devices    Activated Devices    Device with Diagnostic

|    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 |
| 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 |
| 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 |
| 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 |    |    |    |    |    |    |

Error    0

Figure 52: Online > Global State Field

The first row displays the main state of the Master. It can have the state **OPERATE** or **STOP**.

The next row display individual bus errors. A pending error is displayed with a red field. The meaning of the individual abbreviations is described in the following.

| Status Bits | Meaning   |
|-------------|---|
| PDUP        | The device is involved in the duplicate MAC-ID check procedure, to check if other devices with the same address are connected to the network. The duplicate MAC-ID check will be finished, if at least one DeviceNet device could have been found connected to the network. |
| DMAC        | The DEVICE has stopped the duplicate MAC-ID check procedure and found another device having the same MAC-ID address. Change the DEVICE address to avoid this failure.   |
| NRDY        | HOST-NOT-READY-NOTIFICATION<br>indicates if the host program has set its state to operative or not. If this bit is set the host program is not ready to communicate.  |
| EVE         | EVENT-ERROR<br>The used CAN chip has detected transmission errors. The number of detected events is counted in the bus off reports and the error warning limit counter. The bit will be set when the first event was detected and will not be deleted any more.             |
| FAT         | FATAL-ERROR<br>Because of severe bus error, no further bus communication is possible.   |
| NEXC        | NON-EXCHANGE-ERROR<br>At least one Node has not reached the data exchange state and no process data are exchange with it.   |
| ACLR        | AUTO-CLEAR-ERROR<br>The device stopped the communication to all Nodes and reached the auto-clear end state.   |
| CTRL        | CONTROL-ERROR<br>Master parameterization error detected.  |

Table 22: Meaning of collecting status bits in the Global State Field

Further Information is:

**Collective online error location and corresponding error** gives the address of the faulty station and the error in words.

**Statistical bus information** gives the number of detected bus errors and the rejected telegrams.

Device specific status bits: **Parameterized Devices**, **Activated Devices** and **Devices with Diagnostic** are shown if you click at that button. The activated addresses are white numbers.

You can see the diagnostic by double clicking at a highlighted station address (MAC-ID) of a device.

This application is updated cyclically.

## 6.5.4 Extended Device Diagnostic

The Extended Device Diagnostic helps to find bus and configuration errors when the SyCon menu functions are of no further help.

First the desired device must be chosen with a left mouse click on the symbol of the device. Then select the menu **Online > Extended Device Diagnostic**.

This menu opens a list of diagnostic structures. These contain online counters, statuses and parameters:

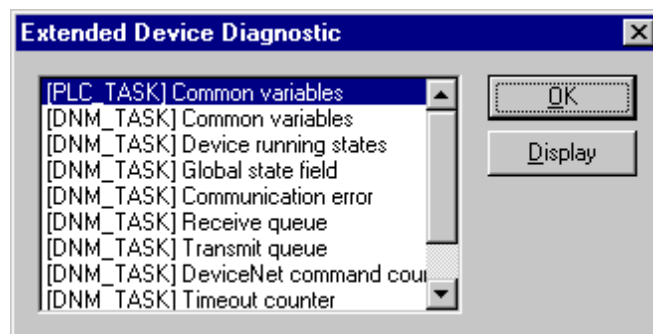


Figure 53: Online > Extended Device Diagnostic (Master)

### 6.5.4.1 Extended Device Diagnostic Master

PLC\_TASK

DNM\_TASK

| Task / Task State                   | Page |
|-------------------------------------|------|
| PLC_TASK Common Variables           | 119  |
| DNM_TASK Common Variables           | 119  |
| DNM_TASK Device Running States      | 121  |
| DNM_Task Global State Field         | 121  |
| DNM_Task Communication Error        | 121  |
| DNM_Task Receive Queue              | 122  |
| DNM_Task Transmit Queue             | 122  |
| DNM_Task DeviceNet Command Counters | 123  |
| DNM_Task Timeout Counter            | 124  |
| DNM_Task Init Counter               | 124  |

Table 23: Extended Device Diagnostic Master

### 6.5.4.2 Extended Device Diagnostic Device (Slave)

PLC\_TASK

DNS\_TASK

| Task / Task State                         | Page |
|---|------|
| <i>PLC_Task Common Variables (Device)</i> | 125  |
| <i>DNS_Task Common Variables</i>          | 125  |
| <i>DNS_TASK Receive Queue (Device)</i>    | 127  |
| <i>DNS_TASK Transmit Queue (Device)</i>   | 127  |

Table 24: Extended Device Diagnostic Device (Slave)

## 6.6 User Data Transfer

The following table shows test functions with user data transfer and the usability for

- Hilscher DeviceNet Master devices
- Hilscher DeviceNet Slaves

| User data transfer function                        | Usage   | Usable with Hilscher DeviceNet Master devices | Usable with Hilscher DeviceNet Master devices |
|--|---|---|---|
| <i>I/O Monitor</i>                                 | Read input data and set output data. (cyclic I/O data exchange) | Yes   | Yes   |
| <i>I/O Watch</i>                                   | Read input data and set output data. (cyclic I/O data exchange) | Yes   | Yes   |
| <i>Get Device Attribute / Set Device Attribute</i> | Get and Set attribute   | Yes   | No, only for Hilscher Master devices          |

Table 25: Overview User Data Transfer

Further test functions with user data transfer are available for

- Explicit Messaging for DeviceNet Master in section *Message Monitor for testing explicit messaging of DeviceNet* on page 87.

### 6.6.1 I/O Monitor

This is an easy way of viewing and changing the first 32 Bytes of the process data image.

The screenshot shows the 'I/O Monitor' window with two main sections: 'Input data' and 'Output data'. Each section contains a table with 11 columns (labeled 0-9) and 8 rows (labeled dec 0-7). The 'Input data' table shows values for dec 0 (0, 0, 127, 255, 127, 255, 127, 0, 0, 0) and dec 1 (0, 0, 0, 0, 0, 0, 0, 0, 0, 0). The 'Output data' table shows values for dec 0 (255, 0, 0, 0, 0, 0, 0, 0, 0, 0) and dec 1 (0, 0, 0, 0, 0, 0, 0, 0, 0, 0). To the right of the tables are buttons for 'OK', 'DEC/HEX', and 'Update'. At the bottom right, it says 'Error 0'.

| dec | 0 | 1 | 2   | 3   | 4   | 5   | 6   | 7 | 8 | 9 |
|-----|---|---|-----|-----|-----|-----|-----|---|---|---|
| 0   | 0 | 0 | 127 | 255 | 127 | 255 | 127 | 0 | 0 | 0 |
| 1   | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 |
| 2   | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 |
| 3   | 0 | 0 |     |     |     |     |     |   |   |   |
| 4   |   |   |     |     |     |     |     |   |   |   |
| 5   |   |   |     |     |     |     |     |   |   |   |
| 6   |   |   |     |     |     |     |     |   |   |   |
| 7   |   |   |     |     |     |     |     |   |   |   |

| dec | 0   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|-----|-----|---|---|---|---|---|---|---|---|---|
| 0   | 255 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3   | 0   | 0 |   |   |   |   |   |   |   |   |
| 4   |     |   |   |   |   |   |   |   |   |   |
| 5   |     |   |   |   |   |   |   |   |   |   |
| 6   |     |   |   |   |   |   |   |   |   |   |
| 7   |     |   |   |   |   |   |   |   |   |   |

Buttons: OK, DEC/HEX, Update

Error 0

Figure 54: Online > IO Monitor

**DEC/HEX** converts the display of the input data. The output data are always in the decimal form.

Enter the output value and then press **Update**.

Always the first 32 input and output Bytes of the process description are shown, also when these Bytes have not been occupied by the configuration.

The display is always in a byte manner.

The I/O Watch Monitor described in the next section allows a more comfortable display.

## 6.6.2 I/O Watch

The I/O Watch monitor can be used in place of the I/O Monitor and offers more functionality.

- Various data formats: Hex, Unsigned Decimal, Signed Decimal, Bit
- The I/O Watch monitor works symbol oriented
- It is not necessary to know the offset addresses

The following firmware supports the I/O Watch monitor function:

| Fieldbus           | From Version  |
|--------------------|---|
| PROFIBUS-DP Master | 1.040 (Combimaster) and respectively. 1.140 (DP-Master) |
| InterBus Master    | 2.040   |
| CANopen Master     | 1.040   |
| DeviceNet Master   | 1.058   |

Table 26: Firmware for I/O Watch functions

The following table lists the typical steps to use the I/O Watch monitor.

Preconditions:

- The project/configuration already exists, containing a DeviceNet Master and a DeviceNet device (or Slave) as described in chapter *Getting Started - Configuration Steps* starting on page 19.
- The Configuration has been downloaded into the DeviceNet Master using **Online > Download**
- Running bus system



1. Open the existing project using **File > Open**.
2. Open the Windows dropdown menu and select **Window > Logical Network View** to change the window. A window with three sections opens

| Left Window            | Center Window | Right Window |
|------------------------|---------------|--------------|
| Project Tree structure | Tag / Symbol  | IO Watch     |

3. Open the tree structure in the left window to reach the I/O module of the device desired:

Project > Master > Device > Connection Type > Date Type

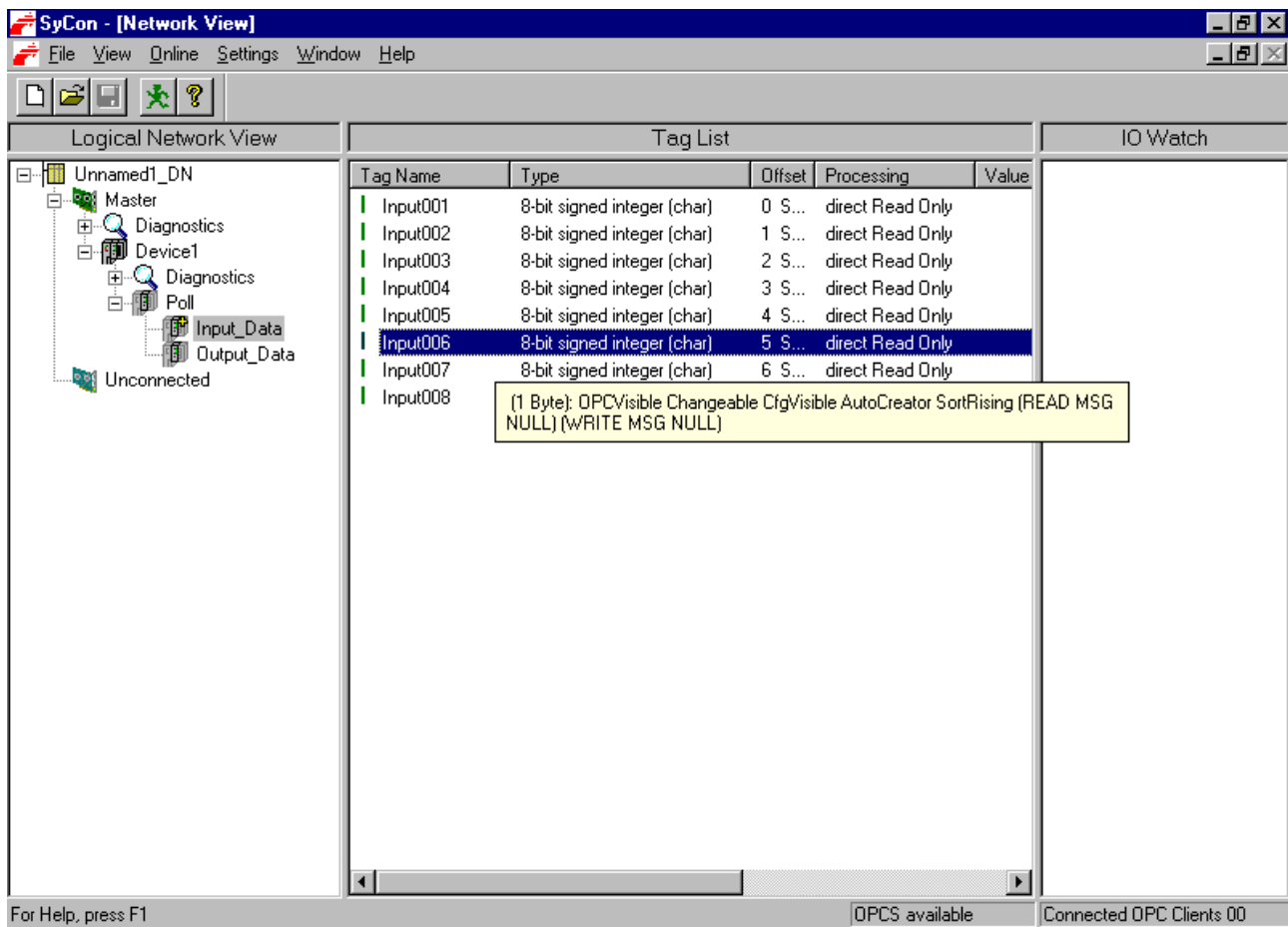
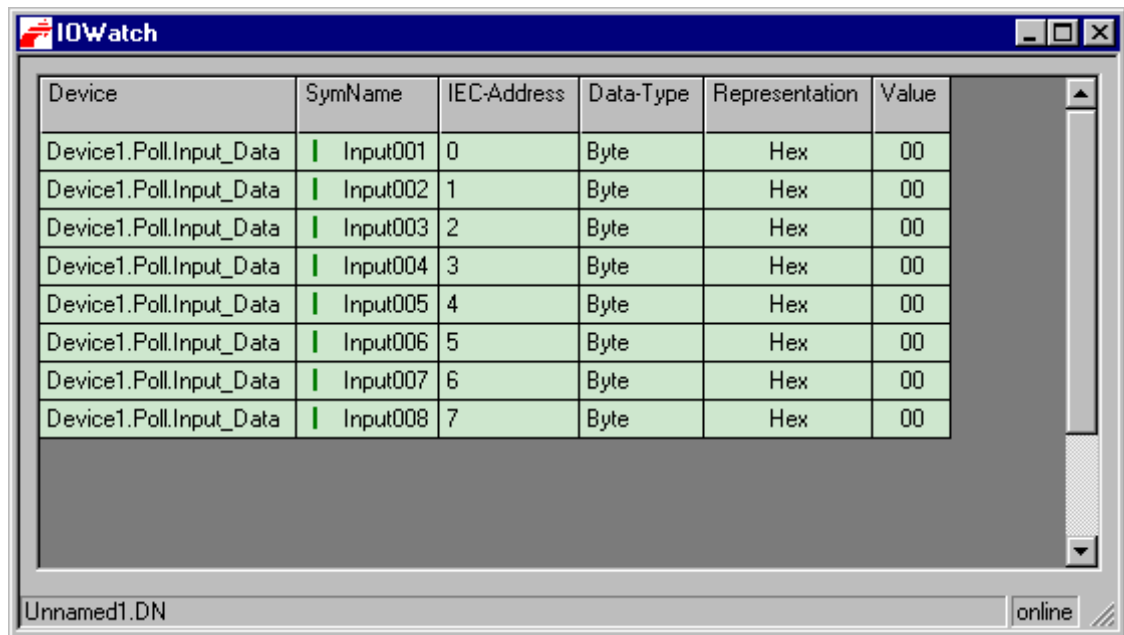


Figure 55: Logical Network View and I/O Watch

4. Left click on the module desired and the tags (I/Os) will be displayed in the center window of the Logical Network View.
5. Select with the left mouse button the tag/symbol desired and drag and drop them in the right window of the Logical Network View (IO Watch).
6. In the right window select the desired tag with the left mouse click to highlight it then right mouse click to open a menu. Select **Start**. A new window called IO Watch appears.
7. A table shows the Device, Symbolic Name, IEC Address (Offset), Data type Representation and Value. Select the line with the desired information. Click on **Hex** under Representation and select the way the values are to be displayed. Choices are Hex, Decimal unsigned, Decimal signed, Bit pattern.
8. Input data are displayed and can't be changed. Output data can be entered into the value column.



| Device                  | SymName  | IEC-Address | Data-Type | Representation | Value |
|-------------------------|----------|-------------|-----------|----------------|-------|
| Device1.Poll.Input_Data | Input001 | 0           | Byte      | Hex            | 00    |
| Device1.Poll.Input_Data | Input002 | 1           | Byte      | Hex            | 00    |
| Device1.Poll.Input_Data | Input003 | 2           | Byte      | Hex            | 00    |
| Device1.Poll.Input_Data | Input004 | 3           | Byte      | Hex            | 00    |
| Device1.Poll.Input_Data | Input005 | 4           | Byte      | Hex            | 00    |
| Device1.Poll.Input_Data | Input006 | 5           | Byte      | Hex            | 00    |
| Device1.Poll.Input_Data | Input007 | 6           | Byte      | Hex            | 00    |
| Device1.Poll.Input_Data | Input008 | 7           | Byte      | Hex            | 00    |

Unnamed1.DN online

Figure 56: I/O Watch Window

To close this windows use Alt-F4 or click the icon in the upper left corner of the window select Exit.

## 6.7 DeviceNet Services

### 6.7.1 Get Device Attribute / Set Device Attribute

This menu selection enables the user to get/receive attribute related information from a Slave device. The user should be familiar with the supported classes, instances, and attributes entrees for the Slave device. These entries should be available within the suppliers' data sheet for the Slave product.

#### **Read**

Clicking the **Get** button will get/read the value from the device. The return value will be represented in Hexadecimal. Clicking the **ASCII** button will change this value to ASCII text display.

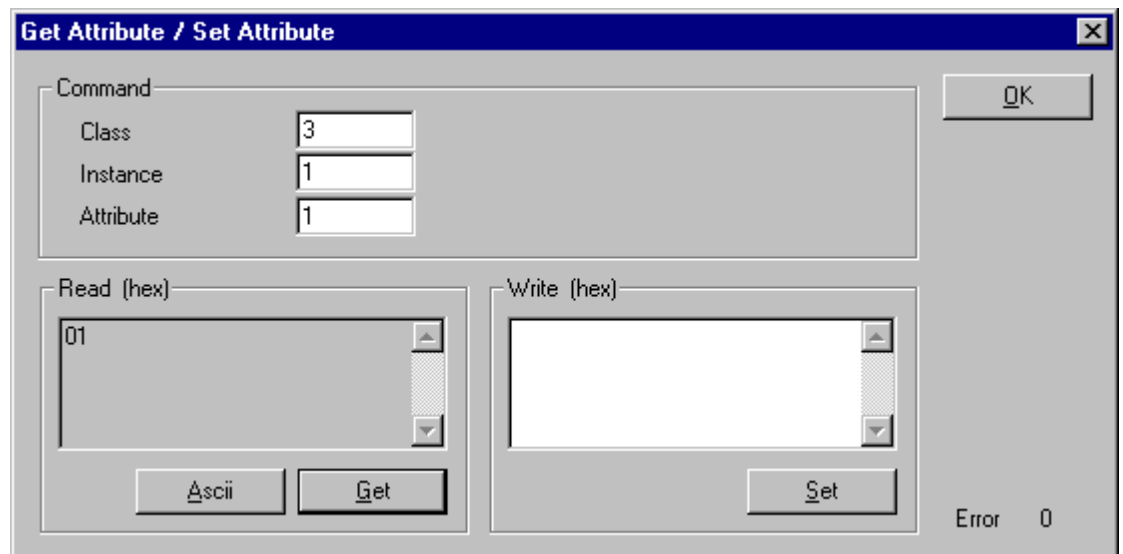


Figure 57: Get Attribute Window

### Write

The value has to be entered in hexadecimal. Clicking the **Set** button will write the information to the Slave device.

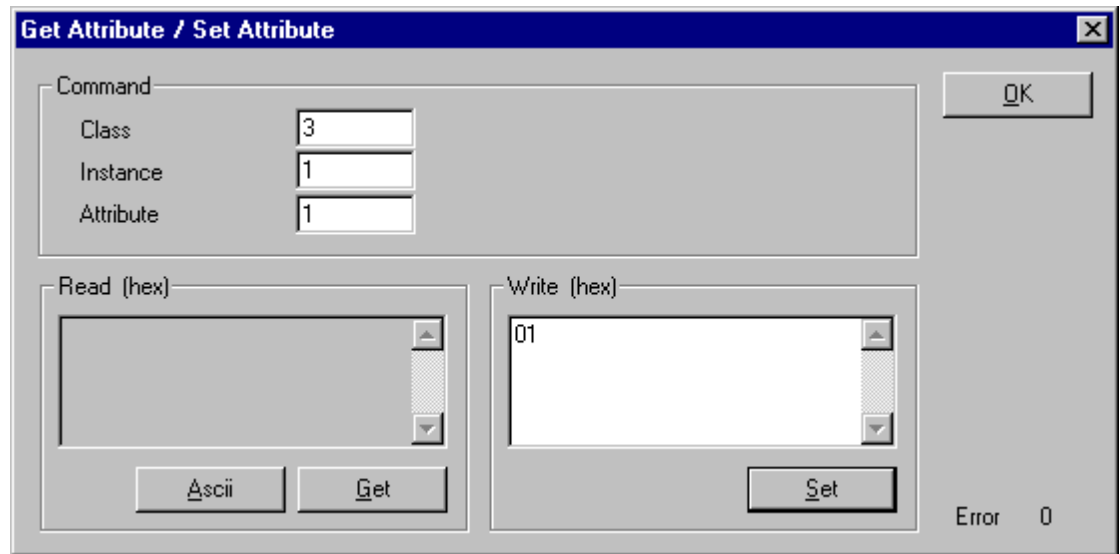


Figure 58: Set Attribute Window

## 6.7.2 Change MAC-ID

To change a MAC-ID of a DeviceNet device (slave) you have to open the Live List. This is described in section *Change MAC-ID* on page 71.

## 6.8 Message Monitor

The Message Monitor permits access to the Mailbox of the CIF. The usage of the Message Monitor assumes advanced knowledge from the user.

First the Hilscher device must be chosen with a left mouse click on the symbol of the Hilscher device. Then call up the **Online > Message Monitor** menu.

**Message Monitor**

File Edit View

**MESSAGE OUTPUT** Counter 0

Message Header

|    |   |    |   |
|----|---|----|---|
| RX | 0 | TX | 0 |
| LN | 0 | NR | 0 |
| A  | 0 | F  | 0 |
| B  | 0 | E  | 0 |

Telegram Header

|             |                                 |
|-------------|---------------------------------|
| Device Adr. | Data Area                       |
| Data Adr.   | Data Idx.                       |
| Data Count  | Data Type                       |
| Function    | <input type="checkbox"/> enable |

Receive data

|    | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|---|---|---|---|---|---|---|---|---|---|
| 0  |   |   |   |   |   |   |   |   |   |   |
| 10 |   |   |   |   |   |   |   |   |   |   |
| 20 |   |   |   |   |   |   |   |   |   |   |
| 30 |   |   |   |   |   |   |   |   |   |   |
| 40 |   |   |   |   |   |   |   |   |   |   |
| 50 |   |   |   |   |   |   |   |   |   |   |
| 60 |   |   |   |   |   |   |   |   |   |   |
| 70 |   |   |   |   |   |   |   |   |   |   |

**MESSAGE INPUT** Counter 0

Message Header

|    |   |                                  |      |
|----|---|----------------------------------|------|
| RX | 0 | TX                               | 255  |
| LN | 0 | Auto NR <input type="checkbox"/> | NR 0 |
| A  | 0 | F                                | 0    |
| B  | 0 | E                                | 0    |

Telegram Header

|             |                                 |
|-------------|---------------------------------|
| Device Adr. | Data Area                       |
| Data Adr.   | Data Idx.                       |
| Data Count  | Data Type                       |
| Function    | <input type="checkbox"/> enable |

Send data

|    | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|---|---|---|---|---|---|---|---|---|---|
| 0  |   |   |   |   |   |   |   |   |   |   |
| 10 |   |   |   |   |   |   |   |   |   |   |
| 20 |   |   |   |   |   |   |   |   |   |   |
| 30 |   |   |   |   |   |   |   |   |   |   |
| 40 |   |   |   |   |   |   |   |   |   |   |
| 50 |   |   |   |   |   |   |   |   |   |   |
| 60 |   |   |   |   |   |   |   |   |   |   |
| 70 |   |   |   |   |   |   |   |   |   |   |

☐ Put cyclic PutMessage

OK

Figure 59: 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 > Review the received data:** all received data is shown

**View > Review the send data:** all the send data is shown

**View > Number of receipt errors:** the number of the receipt errors is 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 60: Save a Message

## 6.8.1 Message Monitor for testing explicit messaging of DeviceNet

In the following, the Message Monitor for reading and writing data via DeviceNet at the Master is described.

The following must be entered in the Message Monitor in order to read or write data via DeviceNet from a Device:

| Message header    |   |              |
|-------------------|---|--------------|
| Rx = 3 (always)   | Tx = 255  |              |
| Ln = (calculated) | Nr = 0 .. 255   |              |
| A = 0             | F = 0   |              |
| B = 17            | E = 0   |              |
| Telegram header   | Meaning for DeviceNet   | Value range  |
| Device Addr       | MAC ID of the device  | 0 .. 63      |
| Data Area         | Class ID  | 0 .. 32      |
| Data Address      | Instance ID   | 0 .. 255     |
| Data Index        | Attribute ID  | 0 .. 255     |
| Data Count        | Data Count<br>Unused in read access<br>length of attribute data in write access | 0<br>1 – 240 |
| Data Type         | Data Type, unused   | 0            |
| Function          | Read  | 1            |
|                   | Write   | 2            |

Table 27: Message Monitor – Example DeviceNet





## 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.



Figure 61: File > Print

The base setting prints information on one sheet only for one device.

**Topology** prints the topology of the Bus system.

**Bus parameters** prints the Bus parameters of the Bus system.

**Address table** prints the address table of the Master.

**Device table** prints the device table.

The scope can be given with the **Device Selection** menu point. The following can be chosen:

- All
- From Station address to Station address
- Selection of a device by means of its description

If no option is selected and the **OK** button is pressed nothing will be printed out. It is like clicking the **Cancel** button.

## 7.3 Export Functions

### 7.3.1 DBM Export

Select the **File > Export > DBM** menu in order to save the previously saved project file (\*.DN 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 extension \*.DBM.

---

**Attention:** The file name can have max. 8 characters.

---

### 7.3.2 CSV Export

With the menu **File > Export > CSV** the configuration data of the connected Slaves can be exported into a table.

Requirement is, that the configuration was saved before the export is executed. The exported file has the ending .CSV (comma separated value) and is taken off in the same directory as the configuration, but with the ending \*.CSV.

The CSV file can be read with a table program like for example Excel.

The CSV Export saves only the text and the values of the configured Slaves. The meaning of the individual values can be shown in the table.

Here is the description of the text and values:

| Value                  | Meaning   |
|------------------------|---|
| <b>Station address</b> | The Station address is the clear device address of the Slave on the bus.  |
| <b>RecordType</b>      | The RecordType defines the arrangement of the structure and is always default 2.  |
| <b>IdentNumber</b>     | This number is the clear device number of the Slave (if available).   |
| <b>VendorNumber</b>    | The VendorNumber is the clear number of the vendor (if available).  |
| <b>VendorName</b>      | Here the name of the vendor is shown.   |
| <b>Device</b>          | Name of the device.   |
| <b>Description</b>     | This is the description of the device, which is set by the user.  |
| <b>MasterAddress</b>   | This is the number of the Master Address, where the devices are connected to (if available).  |
| <b>Settings</b>        | The Settings of the I/O Data is shown here.   |
| <b>Reserved</b>        | reserved  |
| <b>ModulCount</b>      | The ModulCount gives the number of the actual modules. Depending on this there are the DataType, DataSize, DataPosition and the Address from 0 to 59. |
| <b>DataType_0</b>      | The DataType, which is used in the configuration. The code for this you find below this table in section <i>DataType Code</i> .                       |
| <b>DataSize_0</b>      | Number of bytes, which were used by the module.   |
| <b>DataPosition_0</b>  | The byte DataPosition, which is used in the configuration. The code for this you find below this table in section <i>DataPosition Code</i> .          |
| <b>Address_0</b>       | Offset Address in the Dual-port memory  |

Table 28: CSV Export - Meaning of the values

### 7.3.2.1 DataType Code

| D7      | D6                                   | D5 | D4 | D3  | D2 | D1 | D0 |
|---------|--------------------------------------|----|----|---|----|----|----|
| SubFlag | Data Direction                       |    |    | Data Format   |    |    |    |
|         |                                      |    |    | according EN standard<br>0 blank space<br>1 Boolean<br>2 Integer 8<br>3 Integer 16<br>4 Integer 32<br>5 Unsigned Integer 8<br>6 Unsigned Integer 16<br>7 Unsigned Integer 32<br>8 Float<br>9 ASCII<br>10 String<br>14 Bit |    |    |    |
|         | 0 empty space<br>1 input<br>2 output |    |    |   |    |    |    |
|         | 0 start of a module<br>1 sub module  |    |    |   |    |    |    |

Table 29: CSV Export > DataType Code

### 7.3.2.2 DataPosition Code

| D7            | D6 | D5 | D4 | D3                                 | D2 | D1 | D0 |
|---------------|----|----|----|------------------------------------|----|----|----|
| Reserved Area |    |    |    | Bit Position                       |    |    |    |
| reserved      |    |    |    | Bit Position of the Offset Address |    |    |    |

Table 30: CSV Export - DataPosition Code

Example of a CSV file which was exported in Excel:

| A | B | C | D | E | F             | G         | H       | I   | J | K              | L | M | N  | O | P  | Q | R  | S | T  |   |
|---|---|---|---|---|---------------|-----------|---------|-----|---|----------------|---|---|----|---|----|---|----|---|----|---|
| 1 | 1 | 2 | 0 | 0 | Hilscher GmbH | CIF30-DNS | Device1 | 255 | 0 | ***reserved*** | 2 | 8 | 26 | 0 | 15 | 8 | 42 | 0 | 44 | 0 |

Table 31: Example of a CSV file in Excel format

The DeviceNet Salve device has the **Station address** 1 (A1).

The **RecordType** is default 2 (B1).

The **IdentNumber** of the Slave is 0 (C1).

No **VendorNumber** is available, so this is 0 (D1).

The **VendorName** is Hilscher GmbH (E1).

The **Device** has the description CIF30-DNS (F1).

The **Description** of the Device is Device1 (G1).

The **Master address** is 255 (H1).

There are no **Settings** of the I/O data, so this is 0, too (I1).

This field is **Reserved**, because of "reserved" is displayed (J1).

The number of the actual modules is 2, so in the field **ModulCount** a 2 is shown (K1).

The **DataType** of module 1 is 2 (K1).

The **DataSize** of the first module is 8 (L1).

The **DataPosition** of the first module is 26 (M1).

In the following field the DataType, DataSize and DataPosition of the following modules are shown.

---

**Note:** If two or more Slave devices are connected to the Master, these are displayed in the next lines of the table.

---

## 7.4 Edit

With **Edit > Cut** and **Edit > Copy** you put the cut/copied Slave device with its settings and configuration (only not the description of the device) in the Clipboard and with **Edit > Paste** it can be inserting.

The difference between **Cut** and **Copy** is:

With the menu option **Edit > Cut** you move a Slave device from one point in the configuration to another. With the menu option **Edit > Copy** you duplicate an existing Slave device.

If you select **Edit > Cut** a security question appears.

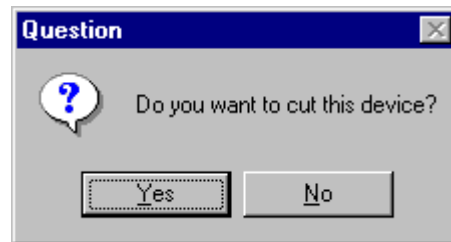


Figure 62: Security question cut device

If you answer this question with **Yes** the Slave device is cut and stays in the clipboard.

When you select **Edit > Paste** the device can be insert again at the position you want.

The mouse pointer changes into the "insert device mouse pointer". Click at the position where the device should be insert. A window opens, where the cutted/copied device can be selected.

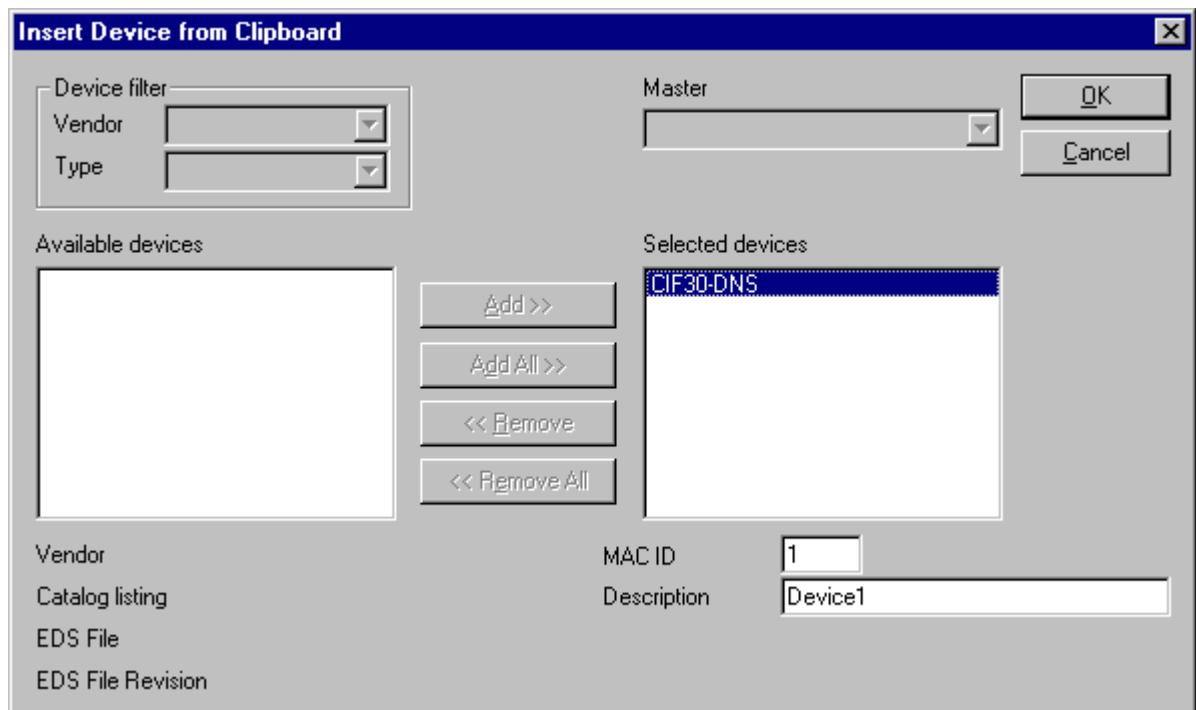


Figure 63: Edit > Paste cutted/copied device

With the **OK** button the device will be insert.

### 7.4.1 Delete

To delete a Slave device you have to have to mark this device and then select the menu **Edit > Delete**. Before SyCon deletes the Slave a security question appears.

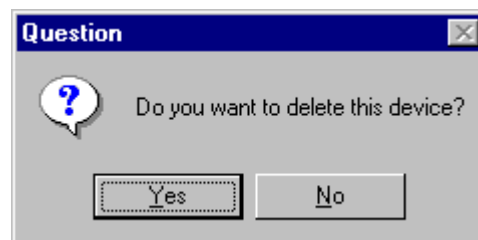


Figure 64: Security question delete device

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

### 7.4.2 Replace

With the menu **Edit > Replace** the Master or a Slave device can be replaced. How to replace the Master look in section *Replace Master* at page 30. If you want to replace a Slave device look in section *Replace Slave* at page 32.







## 7.6 View Menu SyCon

### 7.6.1 Logical Network View

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

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

### 7.6.2 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.

### 7.6.3 Status Bar

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

## 8 Tools

### 8.1 PKV40 / PKV 50 Gateway

The **Tools** menu for the PKV40 and respectively PKV50 is described in an own operating manual.



## 9 Error Numbers

### 9.1 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<br>The communication board is not initialized by the driver.<br>No or wrong configuration found for the given board, check the driver configuration. 0<br>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<br>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)<br>Dual-ported RAM (DPM) not accessible / no hardware found.<br>This error occurs, when the driver is not able to read or write to the Dual-port memory.<br>Check the BIOS setting of the PC Memory address conflict with other PC components.<br>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)<br>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)<br>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<br>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 to the device is loaded.  |

Table 32: 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          | <p>Device: Function PutMessage timeout</p> <p>No message could be send during the timeout period given in the DevPutMessage() function.</p> <p>If you use an interrupt, 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.</p> <p>If you use polling mode, 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!</p> <p>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().</p> <p>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.</p> |
| -18          | <p>Device: Function GetMessage timeout</p> <p>No message received during the timeout period given in the DevGetMessage() function.</p> <p>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.</p> <p>If you use polling mode, 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!</p> <p>The used protocol on the device needs longer than the timeout period given in the DevGetMessage() function.</p>   |
| -19          | Device: No message available   |

Table 33: CIF Device Driver Error Numbers (-15..-19)

| Error Number | Description   |
|--------------|---|
| -20          | <p>Device: Reset command timeout</p> <p>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.</p> <p>The device needs longer than the timeout period given in the DevReset() function. Using device interrupts. The timeout period can differ between fieldbus protocols.</p> <p>If you use an interrupt, 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.</p> <p>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!</p> |
| -21          | <p>Device: COM flag not set</p> <p>The device can not reach communication state. Device not connected to the fieldbus. No station found on the fieldbus. Wrong configuration on the device.</p>   |
| -22          | Device: IO data exchange failed   |
| -23          | <p>Device: IO data exchange timeout</p> <p>The device needs longer than the timeout period given in the DevExchangeIO() function.</p> <p>If you use an interrupt, 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.</p> <p>If you use polling mode, 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!</p>   |
| -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 34: CIF Device Driver Error Numbers (-20..-27)

| Error Number | Description  |
|--------------|--|
| -30          | User: Driver not opened (device driver not loaded)<br>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: IOCTL function failed<br>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: Message buffer 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 35: 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 36: CIF Device Driver Error Numbers (1000)



## 9.2 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 zero                          |
| -46          | User: Buffer too short                                |
| -47          | User: Setup error                                     |

Table 37: CIF Serial Driver Error Numbers (-20..-47)

| Error Number | Description  |
|--------------|--|
| -50          | User: Send message, timeout error  |
| -51          | User: Could not send a message<br>Cable not connected.<br>Wrong cable.<br>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 38: CIF Serial Driver Error Numbers (-20..-47)

## 9.3 CIF TCP/IP Driver Error Numbers

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

### 9.3.1 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 39: CIF TCP/IP Driver Error Numbers - Standard Win32 Socket API errors

### 9.3.2 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 40: CIF TCP/IP Driver Error Numbers - Specific NetIdent Errors

## 9.4 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 existed  |
| 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 41: 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 42: RCS error numbers (answer message) (40..93)

## 9.5 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 43: Database Access Error Numbers (100..130)

## 9.6 Online Data Manager Error Numbers

### 9.6.1 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 44: Online Data Manager Error numbers (1000..1018)

### 9.6.2 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: Message buffer empty        |
| 2011         | Message handler: Message buffer 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 45: Error Numbers of the Message Handler of the Online Data Manager (2010..2027)

### 9.6.3 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 46: Error Numbers of the Driver Functions of the Online Data Manager (2501..2512)

### 9.6.4 Online Data Manager Sub Functions Error Numbers (8001 .. 8035)

The following table lists the error numbers of the Sub Functions 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 47: Sub function Error Numbers of the Driver Functions of the Online Data Manager (8001..8035)



## 9.7 Data Base Functions Error Numbers (4000 .. 4199)

The following table lists the error numbers of the converting functions.

| 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 48: 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 Dbmlnit has assigned a Zero pointer during RCS initialization                                  |
| 4040         | Printer not ready   |
| 4041         | The data base is used from another 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  |
| 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 49: 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 50: 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 CreateTable not found  |
| 4092         | Error in structure of table CreateTable  |
| 4093         | No entry in element szSourceTable  |
| 4094         | No entry in element szDestTable  |
| 4095         | Entry in iSourceType of table CreateTable is wrong   |
| 4096         | Entry in iTranslate of table CreateTable 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 51: Error numbers of converting functions (4082..4199)

## 9.8 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 52: Error Numbers of data base functions (5000 .. 5012)



## 10 Appendix

### 10.1 Extended Device Diagnostic Master

On the following pages the task state structures of the DeviceNet Master are described.

#### 10.1.1 PLC\_TASK Common Variables

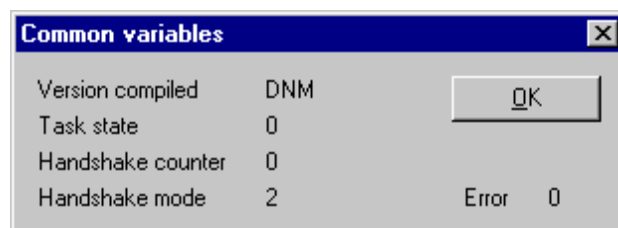


Figure 67: PLC\_TASK Common Variables

| Variable          | Meaning  |
|-------------------|--|
| Version Compiled  | Hardware   |
| Task State        | Task State   |
| Handshake Counter | Counter for the performed process data handshakes  |
| Handshake Mode    | This value represents the actual handshake mode between application and CIF.<br>0 = Bus synchronous, Device Controlled<br>1 = Buffered, Device Controlled<br>2 = No consistence, Uncontrolled<br>3 = Buffered, Host Controlled<br>4 = Bus synchronous, Host Controlled<br>5 = Buffered, extended host controlled |

Table 53: PLC\_TASK Common Variables

#### 10.1.2 DNM\_TASK Common Variables

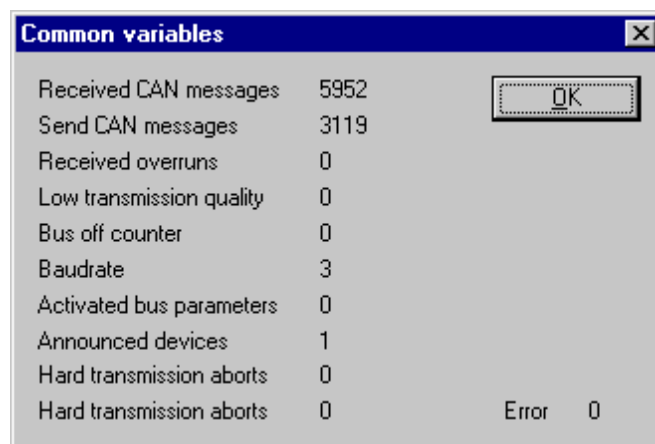


Figure 68: DNM\_TASK Common Variables

| Variable                 | Meaning  |
|--------------------------|--|
| Received message         | number of received CAN-Messages  |
| Sent messages            | number of sent CAN-Messages  |
| Receive overruns         | this counter is incrementing when to many incoming CAN messages overload the Master. An incremented counter will always cause lost CAN message data, so it should normally contain the value 0                       |
| Received Overruns        | our DeviceNet controller has two internal error frame counter for detected   |
| Low Transmission Quality | if the internal DeviceNet controller error frame counter overstep a defined limit  |
| Bus Off Counter          | This number will increment when the bus is off or not powered during bus cycles  |
| Baudrate                 | this value shows numeric the actual baudrate the Master is working with (, 1 = 500kBaud, 2 = 250Kbaud, 3 = 125kBaud)   |
| Activated bus parameters | value 0, the Master has found a configuration data base coming from SyCon, value 1, the Master device isn't configured and need to be configured via SyCon   |
| Announced Nodes          | this value represents the number of found device data sets in the download database  |
| Wrong parameters         | this value indicates, if the Master has detected errors in a device data set which was a containment of the actual downloaded database. For each Slave device that has a wrong entry the counter is incremented by 1 |
| Hard Transmission Aborts | this value indicates transmission aborts by the Master   |

Table 54: DNM\_TASK Common Variables



### 10.1.3 DNM\_Task Device Running States

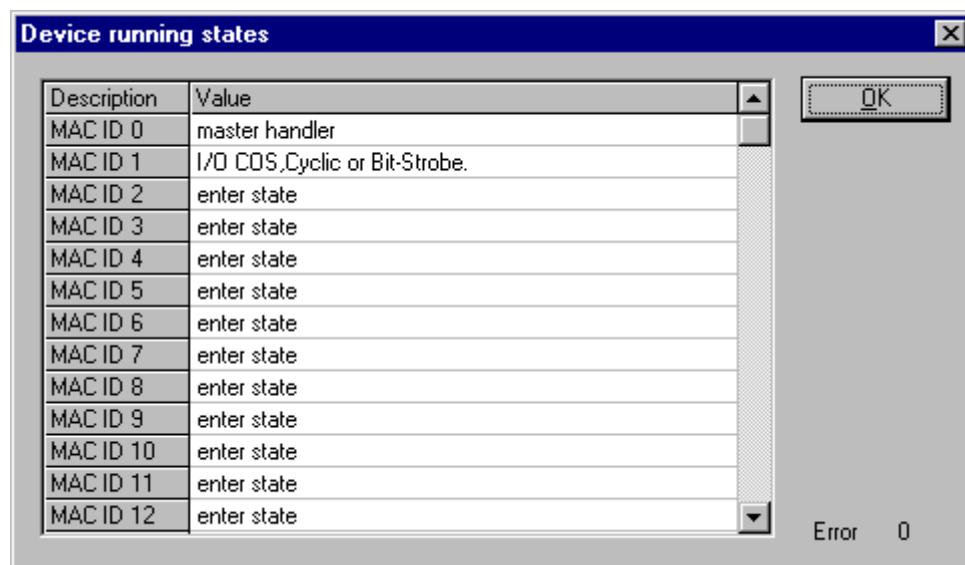


Figure 69: DNM\_Task Device Running States

To handle the Slave devices in their different states the Master device has a Slave device handler running, where each Slave device has its own actual running state. SyCon interprets what the actual state of each Slave and enters these states on the screen in textual form.

### 10.1.4 DNM\_Task Global State Field

See section *Global State Field* at page 74.

### 10.1.5 DNM\_Task Communication Error

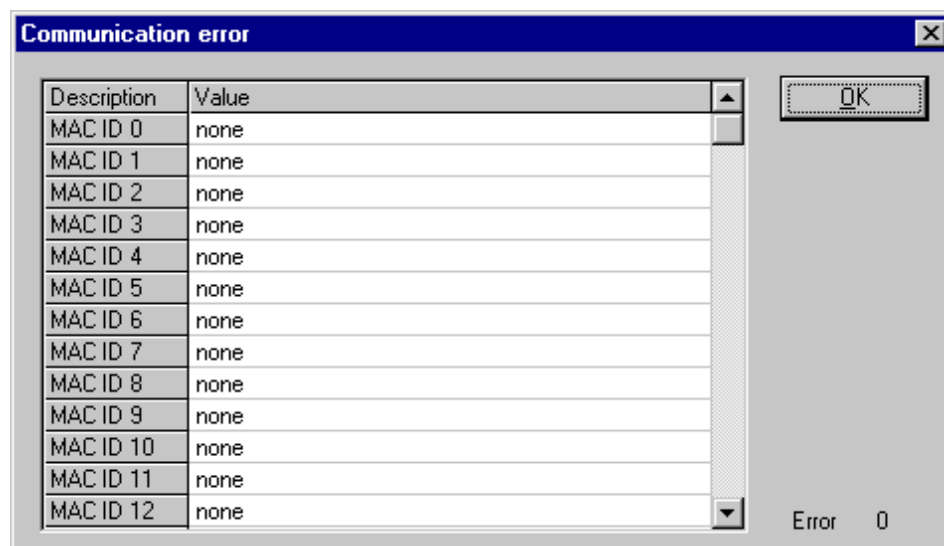


Figure 70: DNM\_Task Communication Error

For each Slave device the Master has an internal online error buffer. SyCon interprets the actual error condition and prints it on the screen in textual form.

### 10.1.6 DNM\_Task Receive Queue

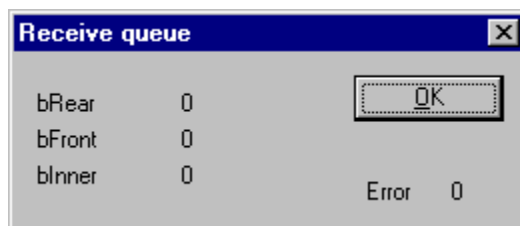


Figure 71: DNM\_Task Receive Queue

The Receive Queue is used to monitor the receive transmission queue of the internal CAN controller.

| Variable | Meaning  |
|----------|--|
| bRear    | A pointer to where the next message will be dequeued from the queue body |
| bFront   | A pointer to where the next message will be stored                       |
| blInner  | The actual number of stored messages                                     |

Table 55: Receive Queue

### 10.1.7 DNM\_Task Transmit Queue

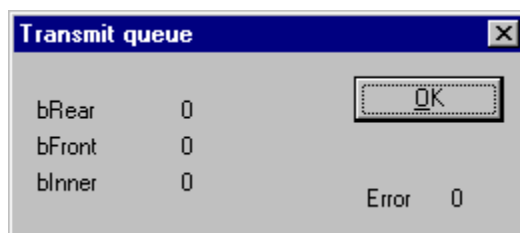


Figure 72: DNM\_Task Transmit Queue

The Transmit Queue is used to monitor the transmission queue of the internal CAN controller.

| Variable | Meaning  |
|----------|--|
| bRear    | A pointer to where the next message will be dequeued from the queue body |
| bFront   | A pointer to where the next message will be stored                       |
| blInner  | The actual number of stored messages                                     |
| fnUser   | not documented   |

Table 56: DNM\_Task Transmit Queue

### 10.1.8 DNM\_Task DeviceNet Command Counters

| DeviceNet command counters |       |                           |       | OK      |
|----------------------------|-------|---------------------------|-------|---------|
| Command                    | Count | Command                   | Count |         |
| OpenUnconnRequest          | 0     | AllocIOCyclicRequest      | 0     |         |
| OpenUnconnAckPos           | 0     | AllocIOCyclicAckPos       | 0     |         |
| OpenUnconnAckNeg           | 0     | AllocIOCyclicAckNeg       | 0     |         |
| CloseUnconnRequest         | 0     | ReleaseIOPollRequest      | 0     |         |
| CloseUnconnAckPos          | 0     | ReleaseIOPollAckPos       | 0     |         |
| CloseUnconnAckNeg          | 0     | ReleaseIOPollAckNeg       | 0     |         |
| AllocateExplicitRequest    | 4     | ReleaseIOBitStrobeRequest | 0     |         |
| AllocateExplicitAckPos     | 2     | ReleaseIOBitStrobeAckPos  | 0     |         |
| AllocateExplicitAckNeg     | 2     | ReleaseIOBitStrobeAckNeg  | 0     |         |
| ReleaseExplicitRequest     | 2     | ReleaseIOCosRequest       | 0     |         |
| ReleaseExplicitAckPos      | 2     | ReleaseIOCosAckPos        | 0     |         |
| ReleaseExplicitAckNeg      | 0     | ReleaseIOCosAckNeg        | 0     |         |
| AllocIOPollRequest         | 0     | ReleaseIOCyclicRequest    | 0     |         |
| AllocIOPollAckPos          | 0     | ReleaseIOCyclicAckPos     | 0     |         |
| AllocIOPollAckNeg          | 0     | ReleaseIOCyclicAckNeg     | 0     |         |
| AllocIOBitStrobeRequest    | 0     | GetAttributeSingleRequest | 8     |         |
| AllocIOBitStrobeAckPos     | 0     | GetAttributeSingleAckPos  | 8     |         |
| AllocIOBitStrobeAckNeg     | 0     | GetAttributeSingleAckNeg  | 0     |         |
| AllocIOCosRequest          | 4     | SetAttributeSingleRequest | 4     |         |
| AllocIOCosAckPos           | 2     | SetAttributeSingleAckPos  | 6     |         |
| AllocIOCosAckNeg           | 2     | SetAttributeSingleAckNeg  | 0     | Error 0 |

Figure 73: DNM\_Task DeviceNet Command Counters

The DeviceNet Command Counters dialog box shows a listing of the DeviceNet specific commands used by the controller and there associated usage count.

### 10.1.9 DNM\_Task Timeout Counter

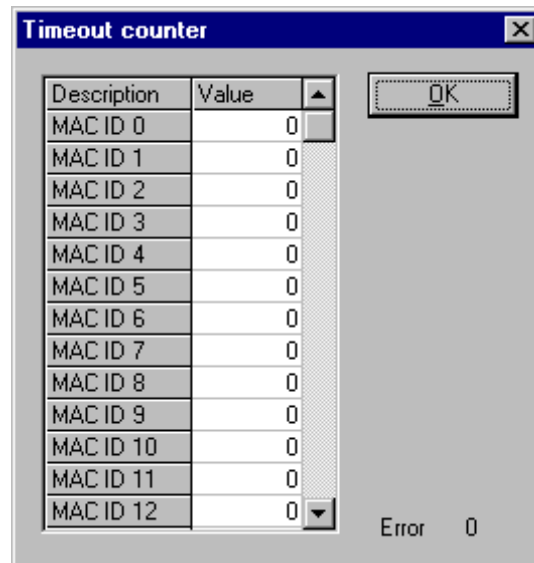


Figure 74: DNM\_Task Timeout Counter

The Timeout Counter shows the number of timeouts for each Slave device configured in the DeviceNet bus system.

### 10.1.10 DNM\_Task Init Counter

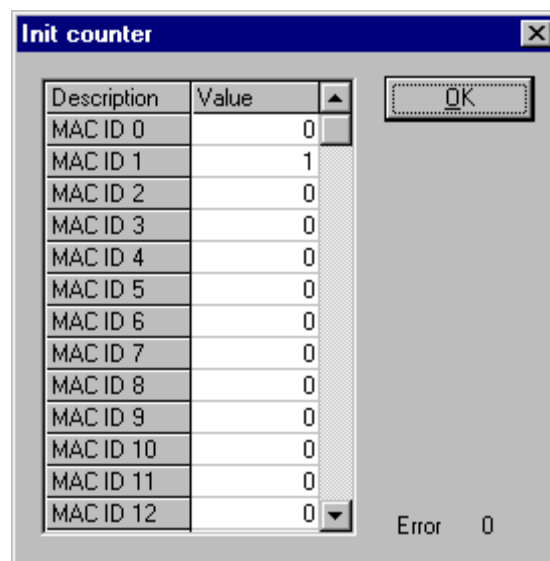


Figure 75: DNM\_Task Init Counter

The Device init counter is incremented whenever the Slave device is initialized. Normally the counter must show the value 1 for each configured Slave, but if a Slave is detected as inactive during the diagnostic procedure, then the Master tries to reinitialize the Slave again. If this happens the Slave init counter is incremented by a value of 1. So values larger than 1 are an indication for communication error to the corresponding Slave.

## 10.2 Extended Device Diagnostic Device (Slave)

On the following pages the task state structures of DeviceNet Slaves are described.

### 10.2.1 PLC\_Task Common Variables (Device)

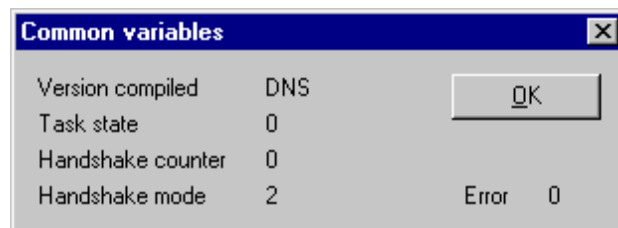


Figure 76: PLC\_Task Common Variables (Slave)

| Variable          | Meaning  |
|-------------------|--|
| Version Compiled  | Hardware   |
| Task State        | Task State   |
| Handshake Counter | Counter for the performed process data handshakes  |
| Handshake Mode    | This value represents the actual handshake mode between application and CIF.<br>0 = Bus synchronous, Device Controlled<br>1 = Buffered, Device Controlled<br>2 = No consistence, Uncontrolled<br>3 = Buffered, Host Controlled<br>4 = Bus synchronous, Host Controlled<br>5 = Buffered, extended host controlled |

Table 57: PLC\_Task Common Variables (Slave)

### 10.2.2 DNS\_Task Common Variables

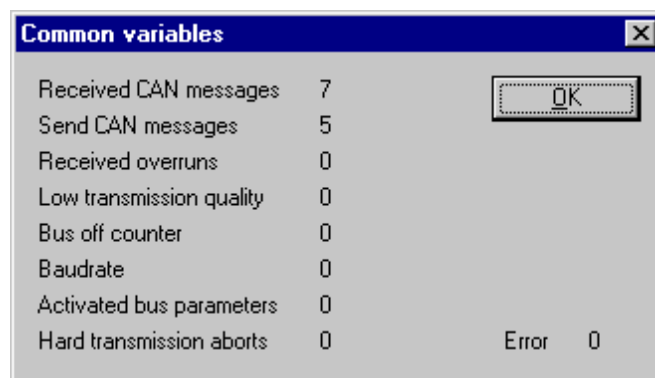


Figure 77: DNS\_Task Common Variables

| Variable                 | Meaning  |
|--------------------------|--|
| Received message         | number of received CAN-Messages  |
| Sent messages            | number of sent CAN-Messages  |
| Received Overruns        | our DeviceNet controller has two internal error frame counter for detected   |
| Low Transmission Quality | if the internal DeviceNet Slave error frame counter overstep a defined limit   |
| Bus Off Counter          | This number will increment when the bus is off or not powered during bus cycles  |
| Baudrate                 | this value shows numeric the actual baudrate the Master is working with(255 = Auto, 0 = 500kBaud, 1 = 250Kbaud, 2 = 125kBaud)                              |
| Activated bus parameters | value 0, the Master has found a configuration data base coming from SyCon, value 1, the Master device isn't configured and need to be configured via SyCon |
| Hard Transmission Aborts | this value indicates transmission aborts by the Slave  |

Table 58: DNS\_Task Common Variables

### 10.2.3 DNS\_TASK Receive Queue (Device)

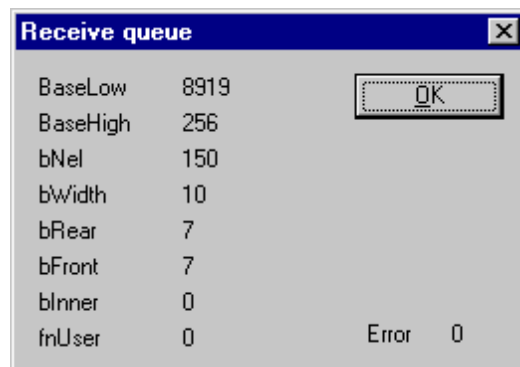


Figure 78: DNS\_TASK Receive Queue (Device)

The Receive Queue is used to monitor the receive transmission queue of the internal CAN controller.

| Variable | Meaning   |
|----------|---|
| BaseLow  | undocumented  |
| BaseHigh | undocumented  |
| BNel     | undocumented  |
| bWidth   | undocumented  |
| bRear    | A pointer to where the next message will be dequeue from the queue body |
| bFront   | A pointer to where the next message will be stored                      |
| blInner  | The actual number of stored messages                                    |
| bfnUser  | undocumented  |

Table 59: DNS\_TASK Receive Queue (Device)

### 10.2.4 DNS\_TASK Transmit Queue (Device)

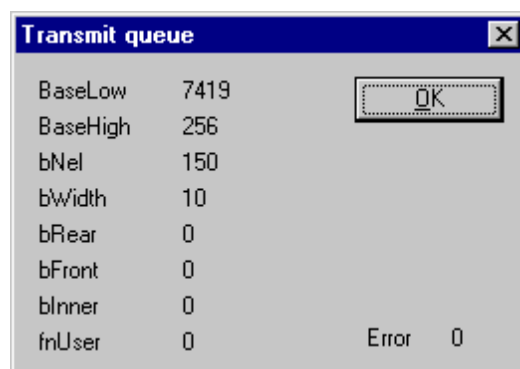


Figure 79: DNS\_TASK Transmit Queue

| Variable | Meaning   |
|----------|---|
| BaseLow  | undocumented  |
| BaseHigh | undocumented  |
| BNel     | undocumented  |
| bWidth   | undocumented  |
| bRear    | A pointer to where the next message will be dequeue from the queue body |
| bFront   | A pointer to where the next message will be stored                      |
| bInner   | The actual number of stored messages                                    |
| bfnUser  | undocumented  |

*Table 60: Transmit Queue (Device)*



# 11 Lists

## 11.1 List of Figures

|  |    |
|--|----|
| Figure 1: Selection for the Installation of the System Configurator in basic version | 10 |
| Figure 2: Selection for the Installation of the licensed System Configurator         | 11 |
| Figure 3: Enter the Name, the Company Name and the License code                      | 12 |
| Figure 4: Example for selecting the fieldbus module DeviceNet                        | 15 |
| Figure 5: Enter the License code   | 16 |
| Figure 6: Note license code is invalid   | 16 |
| Figure 7: General block diagram of an EDS file                                       | 26 |
| Figure 8: Insert > Master  | 28 |
| Figure 9: Hardware Assignment Master   | 29 |
| Figure 10: Security question replace Master  | 30 |
| Figure 11: Edit > Replace Master   | 30 |
| Figure 12: Insert > Device (Slave)   | 31 |
| Figure 13: Settings > Device Configuration   | 33 |
| Figure 14: Settings > Device Configuration > Parameter Data                          | 37 |
| Figure 15: Parameter Data > User Parameter   | 38 |
| Figure 16: Settings > Device Configuration   | 40 |
| Figure 17: Driver Selection  | 41 |
| Figure 18: CIF Device Driver - Driver Description                                    | 43 |
| Figure 19: CIF Device Driver - Board Selection                                       | 43 |
| Figure 20: CIF Device Driver - More Details  | 44 |
| Figure 21: CIF Serial Driver - Driver Description                                    | 45 |
| Figure 22: CIF Serial Driver - Board Selection                                       | 45 |
| Figure 23: CIF TCP/IP Driver - Driver Description                                    | 47 |
| Figure 24: CIF TCP/IP Driver - Type in IP Address manually                           | 48 |
| Figure 25: CIF TCP/IP Driver - Board Selection - Found Device                        | 49 |
| Figure 26: CIF TCP/IP Driver - Board Selection - Assigned Device                     | 49 |
| Figure 27: CIF TCP/IP Driver - Filtered Devices                                      | 50 |
| Figure 28: Set IP Address  | 50 |
| Figure 29: Settings > Bus Parameters   | 51 |
| Figure 30: DeviceNet Master Settings   | 52 |
| Figure 31: Settings > Master Settings  | 52 |
| Figure 32: Image of the method of addressing for input                               | 55 |
| Figure 33: Image of the method of addressing for output                              | 55 |
| Figure 34: Settings > Device Settings  | 56 |
| Figure 35: Settings > Project Information  | 57 |
| Figure 36: Settings > Path   | 58 |
| Figure 37: Settings > Language   | 58 |
| Figure 38: Settings > Start Options  | 59 |
| Figure 39: Security question before Download   | 61 |
| Figure 40: Online > Download   | 61 |
| Figure 41: Online > Firmware Download  | 62 |
| Figure 42: Online > Firmware / Reset   | 63 |
| Figure 43: Online > Device Info  | 63 |
| Figure 44: Online > Activate Driver  | 64 |
| Figure 45: Online > Automatic Network Scan (during scan)                             | 66 |
| Figure 46: Online > Automatic Network Scan (after scan)                              | 67 |
| Figure 47: Information on a device in the automatic scan window                      | 68 |
| Figure 48: Online > Live List  | 71 |
| Figure 49: Online > Live List > Change MAC-ID  | 71 |

|   |     |
|---|-----|
| Figure 50: The Debug window                             | 72  |
| Figure 51: Online > Device Diagnostic                   | 73  |
| Figure 52: Online > Global State Field                  | 74  |
| Figure 53: Online > Extended Device Diagnostic (Master) | 76  |
| Figure 54: Online > IO Monitor                          | 79  |
| Figure 55: Logical Network View and I/O Watch           | 81  |
| Figure 56: I/O Watch Window                             | 82  |
| Figure 57: Get Attribute Window                         | 83  |
| Figure 58: Set Attribute Window                         | 84  |
| Figure 59: Online > Message Monitor                     | 85  |
| Figure 60: Save a Message                               | 86  |
| Figure 61: File > Print                                 | 90  |
| Figure 62: Security question cut device                 | 94  |
| Figure 63: Edit > Paste cutted/copied device            | 95  |
| Figure 64: Security question delete device              | 95  |
| Figure 65: View > Device Table                          | 96  |
| Figure 66: View > Address Table                         | 97  |
| Figure 67: PLC_TASK Common Variables                    | 119 |
| Figure 68: DNM_TASK Common Variables                    | 119 |
| Figure 69: DNM_Task Device Running States               | 121 |
| Figure 70: DNM_Task Communication Error                 | 121 |
| Figure 71: DNM_Task Receive Queue                       | 122 |
| Figure 72: DNM_Task Transmit Queue                      | 122 |
| Figure 73: DNM_Task DeviceNet Command Counters          | 123 |
| Figure 74: DNM_Task Timeout Counter                     | 124 |
| Figure 75: DNM_Task Init Counter                        | 124 |
| Figure 76: PLC_Task Common Variables (Slave)            | 125 |
| Figure 77: DNS_Task Common Variables                    | 125 |
| Figure 78: DNS_TASK Receive Queue (Device)              | 127 |
| Figure 79: DNS_TASK Transmit Queue                      | 127 |

## 11.2 List of Tables

|   |     |
|---|-----|
| Table 1: SyCon Main Functions   | 7   |
| Table 2: Selection during installation  | 13  |
| Table 3: Overview Communication Types   | 19  |
| Table 4: Step for Configuration Hilscher DeviceNet Master to any DeviceNet Device                     | 20  |
| Table 5: Step for Configuration Hilscher DeviceNet Slave to any DeviceNet Master                      | 22  |
| Table 6: Step for Configuration Hilscher DeviceNet Master to Hilscher DeviceNet Slave                 | 23  |
| Table 7: Insert > Master Symbol   | 28  |
| Table 8: Insert Master Mouse Pointer  | 28  |
| Table 9: Insert > Device Symbol   | 31  |
| Table 10: Insert Device Mouse Pointer   | 31  |
| Table 11: Overview I/O Connections  | 34  |
| Table 12: Driver Selection  | 42  |
| Table 13: Device Assignment - Checkboxes of the CIF Device Driver                                     | 43  |
| Table 14: Device Assignment - Checkboxes of the CIF Serial Driver                                     | 46  |
| Table 15: Device Assignment - Checkboxes of the CIF TCP/IP Driver                                     | 49  |
| Table 16: Addressing Mode   | 54  |
| Table 17: Example for data in the process data image  | 54  |
| Table 18: Read Classes.Instance.Attribute during the network scan                                     | 66  |
| Table 19: Meaning of the columns in the automatic network scan  | 67  |
| Table 20: Overview Diagnostic Functions   | 70  |
| Table 21: Meaning of the bits in the Device Diagnostic  | 73  |
| Table 22: Meaning of collecting status bits in the Global State Field                                 | 75  |
| Table 23: Extended Device Diagnostic Master   | 76  |
| Table 24: Extended Device Diagnostic Device (Slave)   | 77  |
| Table 25: Overview User Data Transfer   | 78  |
| Table 26: Firmware for I/O Watch functions  | 80  |
| Table 27: Message Monitor – Example DeviceNet   | 87  |
| Table 28: CSV Export - Meaning of the values  | 92  |
| Table 29: CSV Export > DataType Code  | 92  |
| Table 30: CSV Export - DataPosition Code  | 93  |
| Table 31: Example of a CSV file in Excel format   | 93  |
| Table 32: CIF Device Driver Error Numbers (-1...-14)  | 101 |
| Table 33: CIF Device Driver Error Numbers (-15...-19)   | 102 |
| Table 34: CIF Device Driver Error Numbers (-20...-27)   | 103 |
| Table 35: CIF Device Driver Error Numbers (-30...-49)   | 104 |
| Table 36: CIF Device Driver Error Numbers (1000)  | 104 |
| Table 37: CIF Serial Driver Error Numbers (-20...-47)   | 105 |
| Table 38: CIF Serial Driver Error Numbers (-20...-47)   | 106 |
| Table 39: CIF TCP/IP Driver Error Numbers - Standard Win32 Socket API errors                          | 107 |
| Table 40: CIF TCP/IP Driver Error Numbers - Specific NetIdent Errors                                  | 107 |
| Table 41: RCS error numbers (answer message) (4...39)   | 108 |
| Table 42: RCS error numbers (answer message) (40...93)  | 109 |
| Table 43: Database Access Error Numbers (100...130)   | 110 |
| Table 44: Online Data Manager Error numbers (1000...1018)   | 111 |
| Table 45: Error Numbers of the Message Handler of the Online Data Manager (2010...2027)               | 111 |
| Table 46: Error Numbers of the Driver Functions of the Online Data Manager (2501...2512)              | 112 |
| Table 47: Sub function Error Numbers of the Driver Functions of the Online Data Manager (8001...8035) | 112 |
| Table 48: Error numbers of converting functions (4000...4029)   | 113 |
| Table 49: Error numbers of converting functions (4030...4060)   | 114 |
| Table 50: Error numbers of converting functions (4061...4075)   | 115 |
| Table 51: Error numbers of converting functions (4082...4199)   | 116 |

|   |     |
|---|-----|
| Table 52: Error Numbers of data base functions (5000 .. 5012) | 117 |
| Table 53: PLC_TASK Common Variables                           | 119 |
| Table 54: DNM_TASK Common Variables                           | 120 |
| Table 55: Receive Queue                                       | 122 |
| Table 56: DNM_Task Transmit Queue                             | 122 |
| Table 57: PLC_Task Common Variables (Slave)                   | 125 |
| Table 58: DNS_Task Common Variables                           | 126 |
| Table 59: DNS_TASK Receive Queue (Device)                     | 127 |
| Table 60: Transmit Queue (Device)                             | 128 |