hilscher
COMPETENCE IN
COMMUNICATION

Operating Instruction Manual
SyConCO
System Configurator CANopen

Hilscher Gesellschaft für Systemautomation mbH
www.hilscher.com
DOC980301OI13EN | Revision 13 | English | 2012-06 | Released | Public

List of Revisions

Index	Date	Version	Chapter	Revision
7	06.05.03	2.706 CANopen.dll 2.651 SyCon.exe	4.3.1 4.3.3 4.4.2 4.4.2.5 4.4.2.11 5.2 7.5.1 10.9	Node ID Master Node ID Master Node Configuration now also possible with Heartbeat Receive PDO now also with Transmission Type 255 Heartbeat added in Configuration Error Control Protocol Bus Parameter: Node ID for Master, Heartbeat Device Table: Supervision times added Table Error Register
8	15.01.04	2.706 CANopen.dll 2.651 SyCon.exe	all 5.1 9.3	revised Section 'Device Assignment' completely revised Section 'CIF TCP/IP Error Numbers' added
9	25.02.05	2.709 CANopen.dll 2.653 SyCon.exe	4.4.2 10.5.4	Section ’Node Configuration’ revised Section ’Device Profile 405’ created Length of the EDS file name changed from 13 characters to 32 caracters
10	12.12.08	2.808 CANopen.dll 2.705 SyCon.exe	4.4.2 4.4.2.3	Node configuration with new function: Module configuration Section Module configuration added
11	19.11.09	2.81x CANopen.dll 2.71x SyCon.exe	2.1	System Requirements: Windows® 2000 / Windows® XP / Windows® Vista / Windows® 7
12	10.01.11	2.81x CANopen.dll 2.71x SyCon.exe	9.6	Error number 235 added
13	13.06.12	2.81x CANopen.dll 2.71x SyCon.exe	2.1	System Requirements: Windows® 2000 removed

Table of Contents

1 Overview SyCon

1.1 Main Functions

The main functions of the CANopen System Configurator are:

Function	**Section**	**Short Description**
Configuration	*Overview Communication Types*	Overview communication types and description of the configuration steps
Diagnostic	*Diagnostic Functions*	Diagnostic functions, e.g. Life List, Debugger, Global State Field etc.
	User Data Transfer	I/O-Monitor, I/O-Watch, Read and Write Objects, Message-Monitor, Live List
Documentation	*Project Information*	Set the project information
	Print	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, 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 a 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, e.g. live list or Slave diagnostic information on PROFIBUS. In this case a Slave 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.

1.3 CAN and CANopen

1.3.1 CAN

CAN means Controller Area Network. The CAN specification describes the physical interface, the telegram structure and the secure transmission of a CAN telegram. It describes the send and the receive of a telegram.

The CAN telegram consists (simplified) of a telegram identifier and 0 to 8 bytes of data.

The meaning of the telegram identifier and of the max. 8 bytes user data is not described, e.g. it does not say anything about the application layer.

1.3.2 CANopen

CANopen is an open standard and based on CAN. The meaning of the telegram identifier and of the 0 to 8 bytes of user data is described (specified).

CANopen is a standard application layer defined by the CIA (CAN in automation) specifications DS 301.

CANopen is network concept and determines what data and what services are to be transmitted and what is the meaning of the data for the individual device classes.

CANopen provides functions for the network initialization, the network guarding and the network configuration.

CANopen offers a big flexibility.

1.3.3 CANopen Device Model

A CANopen device can be described generally as 3 components: communication, objects and application.

Component	**Description**
Communication	The communication unit contains the mechanism for the transport of data according to the CANopen specification over the CAN.
Object dictionary	The object dictionary is the connection between the application unit and the communication unit. It contains configuration data and device information. All entries have an object index (index) and a sub index.
Application	The application unit describes the function of the CANopen device.

Table 2: Components of the CANopen Device Model

1.4 Legal Notes

1.4.1 Copyright

The images, photographs and texts in the accompanying material (user manual, accompanying texts, documentation, etc.) are protected by German and international copyright law as well as international trade and protection provisions. You are not authorized to duplicate these in whole or in part using technical or mechanical methods (printing, photocopying or other methods), to manipulate or transfer using electronic systems without prior written consent. You are not permitted to make changes to copyright notices, markings, trademarks or ownership declarations. The included diagrams do not take the patent situation into account. The company names and product descriptions included in this document may be trademarks or brands of the respective owners and may be trademarked or patented. Any form of further use requires the explicit consent of the respective rights owner.

1.4.2 Important Notes

The user manual, accompanying texts and the documentation were created for the use of the products by qualified experts, however, errors cannot be ruled out. For this reason, no guarantee can be made and neither juristic responsibility for erroneous information nor any liability can be assumed. Descriptions, accompanying texts and documentation included in the user manual do not present a guarantee nor any information about proper use as stipulated in the contract or a warranted feature. It cannot be ruled out that the user manual, the accompanying texts and the documentation do not correspond exactly to the described features, standards or other data of the delivered product. No warranty or guarantee regarding the correctness or accuracy of the information is assumed.

We reserve the right to change our products and their specification as well as related user manuals, accompanying texts and documentation at all times and without advance notice, without obligation to report the change. Changes will be included in future manuals and do not constitute any obligations. There is no entitlement to revisions of delivered documents. The manual delivered with the product applies.

Hilscher Gesellschaft für Systemautomation mbH is not liable under any circumstances for direct, indirect, incidental or follow-on damage or loss of earnings resulting from the use of the information contained in this publication.

1.4.3 Exclusion of Liability

The software was produced and tested with utmost care by Hilscher Gesellschaft für Systemautomation mbH and is made available as is. No warranty can be assumed for the performance and flawlessness of the software for all usage conditions and cases and for the results produced when utilized by the user. Liability for any damages that may result from the use of the hardware or software or related documents, is limited to cases of intent or grossly negligent violation of significant contractual obligations. Indemnity claims for the violation of significant contractual obligations are limited to damages that are foreseeable and typical for this type of contract.

It is strictly prohibited to use the software in the following areas:

- for military purposes or in weapon systems;
- for the design, construction, maintenance or operation of nuclear facilities;
- in air traffic control systems, air traffic or air traffic communication systems;
- in life support systems;
- in systems in which failures in the software could lead to personal injury or injuries leading to death.

We inform you that the software was not developed for use in dangerous environments requiring fail-proof control mechanisms. Use of the software in such an environment occurs at your own risk. No liability is assumed for damages or losses due to unauthorized use.

1.4.4 Warranty

Although the hardware and software was developed with utmost care and tested intensively, Hilscher Gesellschaft für Systemautomation mbH does not guarantee its suitability for any purpose not confirmed in writing. It cannot be guaranteed that the hardware and software will meet your requirements, that the use of the software operates without interruption and that the software is free of errors. No guarantee is made regarding infringements, violations of patents, rights of ownership or the freedom from interference by third parties. No additional guarantees or assurances are made regarding marketability, freedom of defect of title, integration or usability for certain purposes unless they are required in accordance with the law and cannot be limited. Warranty claims are limited to the right to claim rectification.

1.4.5 Export Regulations

The delivered product (including the technical data) is subject to export or import laws as well as the associated regulations of different counters, in particular those of Germany and the USA. The software may not be exported to countries where this is prohibited by the United States Export Administration Act and its additional provisions. You are obligated to comply with the regulations at your personal responsibility. We wish to inform you that you may require permission from state authorities to export, re-export or import the product.

2 Installation and Licensing

2.1 System Requirements

- PC with Pentium processor or higher
- Windows® XP SP3 / Windows® Vista SP2 (32-bit) / Windows® 7 SP1 (32/64-bit)
- Free disk space: 30 - 80 MByte
- DVD ROM drive
- RAM: min. 256 MByte
- Graphic resolution: min. 800 x 600 pixel, recommended 1024 x 768
- Keyboard and Mouse

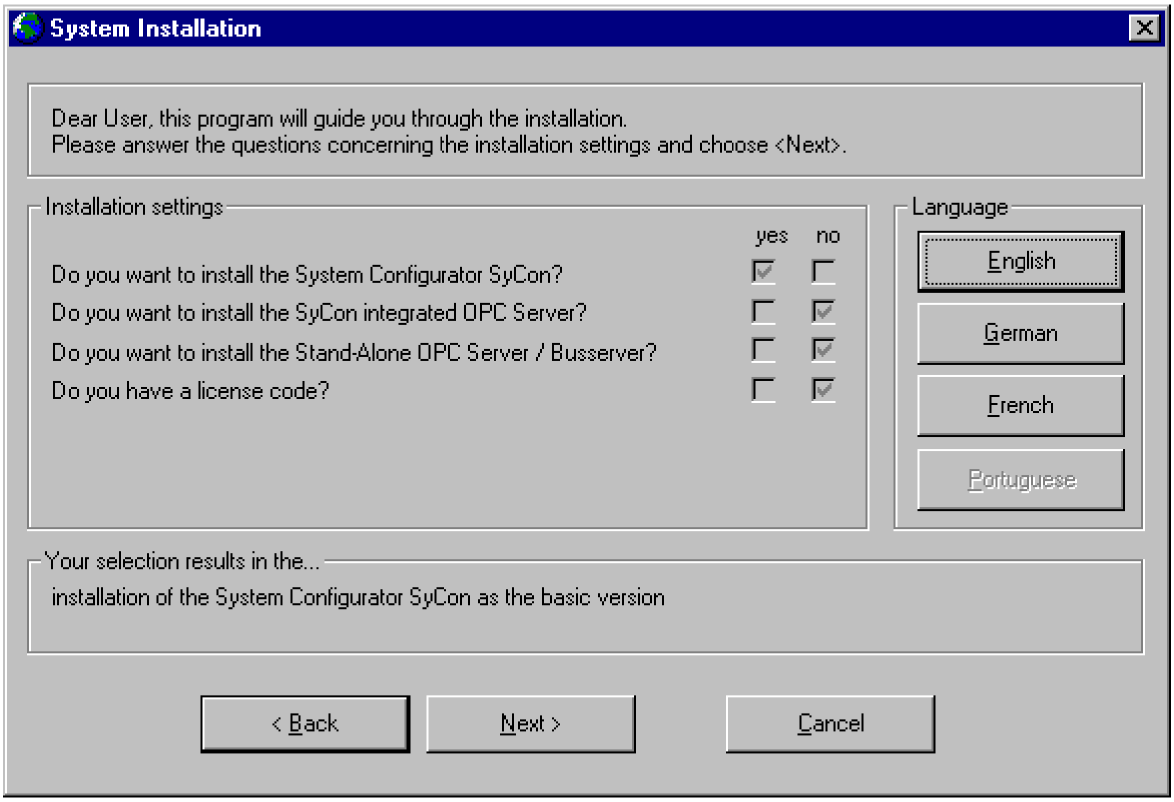
2.2 Software Installation

Close all application programs on the system!

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

Note: Administrator privileges are required on Windows® XP/Vista/7 systems for installation!

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



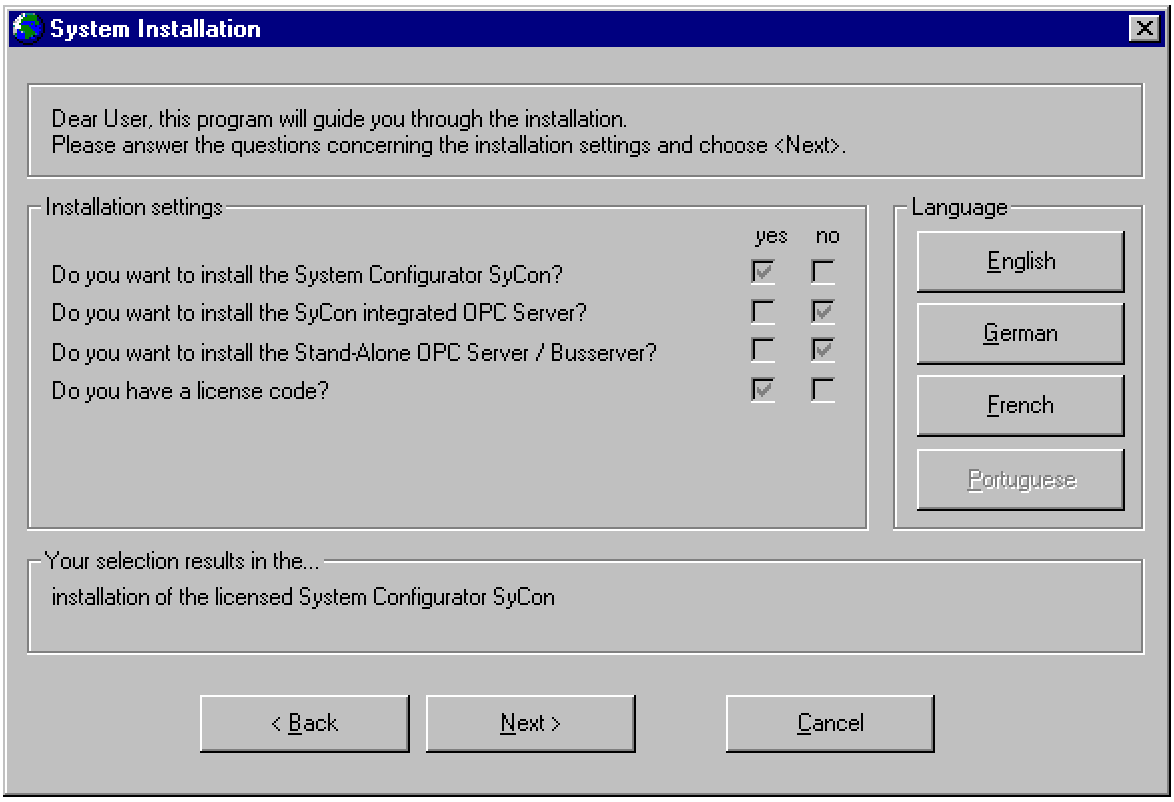
Dear User, this program will guide you through the installation.
Please answer the questions concerning the installation settings and choose <Next>.

Installation settings

	yes	no
Do you want to install the System Configurator SyCon?	☑	□
Do you want to install the SyCon integrated OPC Server?	□	☑
Do you want to install the Stand-Alone OPC Server / Busserver?	□	☑
Do you have a license code?	□	☑

Your selection results in the...
installation of the System Configurator SyCon as the basic version

Figure 1: Selection for the Installation of the System Configurator in Basic Version



System Installation

Dear User, this program will guide you through the installation.
Please answer the questions concerning the installation settings and choose <Next>.

Installation settings

	yes	no
Do you want to install the System Configurator SyCon?	☑	□
Do you want to install the SyCon integrated OPC Server?	□	☑
Do you want to install the Stand-Alone OPC Server / Busserver?	□	☑
Do you have a license code?	☑	□

Your selection results in the...
installation of the licensed System Configurator SyCon

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)

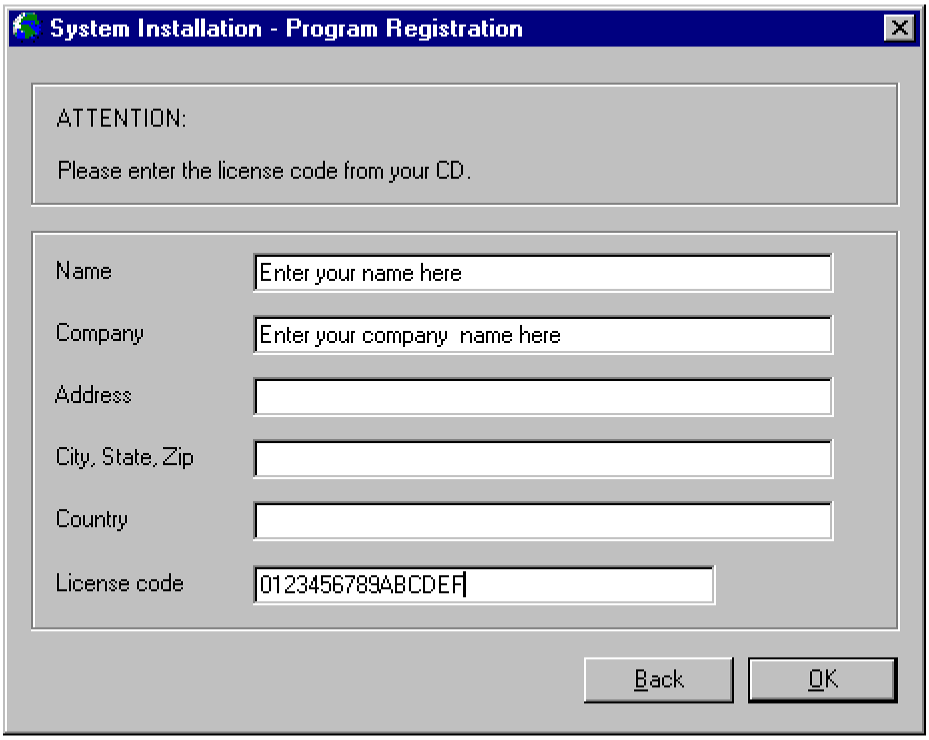
If you have a license code or it is printed on the label of the DVD, then answer the question for an existing license code with yes, otherwise a basic version of the System Configurator will be installed. Enter your name and the company name.

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 DVD, 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**.



System Installation - Program Registration

ATTENTION:

Please enter the license code from your CD.

Name: Enter your name here

Company: Enter your company name here

Address:

City, State, Zip:

Country:

License code: 0123456789ABCDEF

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 System Configurator and its Components
AS-Interface	Selected	Program DLL and Components of the Fieldbus System or the Protocol
CANopen	Selected	
DeviceNet	Selected	
InterBus	Selected	
PROFIBUS	Selected	
Ethernet / Protocol	Selected	
CIF Device Driver	Selected C:\Programs\CIF Device Driver	CIF Device Driver
Program Menu	SyCon System Configurator	Folder under **Start > Programs**

Table 3: 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 formulary with. Fill out this paper (formulary) 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 *20*.

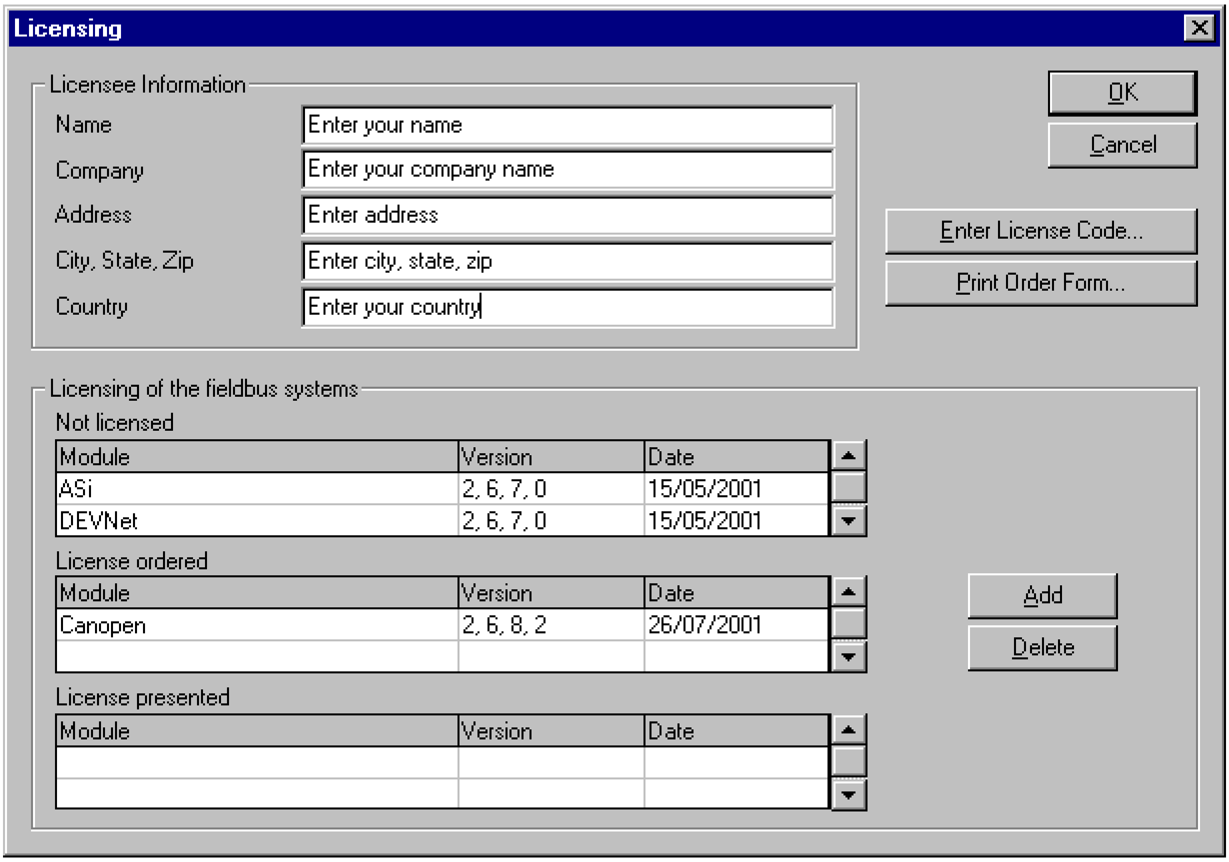
An order form for a license for the System Configurator can be printed out and is described in the next section.

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 list the modules, that are not licensed. Doubleclick or select and click the **Add** button to move the desired modules into the table in the middle that are printed on the order form later. The modules, which are already licensed, are shown in the last table.



Licensee Information

Name	Enter your name
Company	Enter your company name
Address	Enter address
City, State, Zip	Enter city, state, zip
Country	Enter your country

Licensing of the fieldbus systems

Not licensed

Module	Version	Date
ASi	2, 6, 7, 0	15/05/2001
DEVNet	2, 6, 7, 0	15/05/2001

License ordered

Module	Version	Date
Canopen	2, 6, 8, 2	26/07/2001

License presented

Module	Version	Date

Figure 4: Example for Selection of the Fieldbus Module CANopen

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.

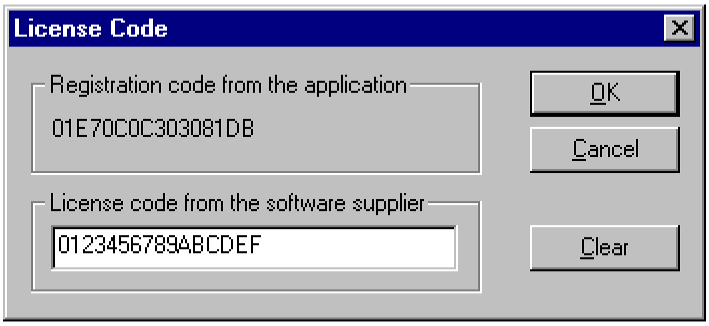
Select the menu **Help > Licensing**. The licensing window will be opened.

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 was 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 windows 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 the System Configurator (license code with less than 16 digits) expects a license in the device.



License Code

Registration code from the application

01E70C0C303081DB

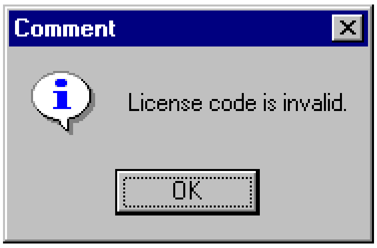
License code from the software supplier

0123456789ABCDEF

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.



Comment

License code is invalid.

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

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

CANopen offers the following communication possibilities:

Communication	**Overview in section**	**Page**
PDO (CANopen)	*Configuration for PDO Communication (CANopen)*	*24*
SDO (CANopen)	*Configuration for SDO Communication (CANopen)*	*24*
Send/Receive Transparent (CAN)	*Configuration for Send/Receive transparent (CAN)*	*24*

Table 4: Overview Communication Types CANopen

3.1.1 Configuration for PDO Communication (CANopen)

Communication	Device	Device	Described in section	Page
PDO (CANopen)	Hilscher CANopen Master	Any CANopen Node	*Configuration Hilscher CANopen Master to any CANopen Node*	*25*
	Any CANopen Master	Hilscher CANopen Node	*Configuration Hilscher CANopen Node to any CANopen Master*	*27*
	Hilscher CANopen Master	Hilscher CANopen Node	*Configuration Hilscher CANopen Master to a Hilscher CANopen Node*	*28*

Table 5: Overview Communication Types PDO Communication

3.1.2 Configuration for SDO Communication (CANopen)

Communication	Device	Device	Described in section	Page
SDO (CANopen)	Hilscher CANopen Master	Any CANopen Node	*Configuration Hilscher CANopen Master to any CANopen Node*	*30*
	Any CANopen Master	Hilscher CANopen Node	*Configuration Hilscher CANopen Node to any CANopen Master*	*31*
	Hilscher CANopen Master	Hilscher CANopen Node	*Configuration Hilscher CANopen Master to a Hilscher CANopen Node*	*32*

Table 6: Overview Communication Types SDO Communication

3.1.3 Configuration for Send/Receive transparent (CAN)

Communication	Device	Device	Described in section	Page
Send / Receive transparently (CAN)	Hilscher CANopen Master	Any CAN device	*Configuration Hilscher CANopen Master to any CAN Device for Send/Receive transparent (CAN)*	*33*
	Hilscher CANopen Node	Any CAN device	*Configuration Hilscher CANopen Node to any CAN Device for Send/Receive transparently (CAN)*	*34*

Table 7: Overview Communication Types CAN send/receive transparent

3.2 Configuration for PDO Communication

3.2.1 Configuration Hilscher CANopen Master to any CANopen Node (PDO)

The following table describes the steps to configure a Hilscher CANopen Master to any CANopen Node for PDO communication, as it is typical for many cases.

#	Action	Menu in the System Configurator	Detail information in section	Page
1	Create a new project	**File > New > CANopen**	*Setting up the CANopen Configuration*	*35*
2	Copy EDS file of CANopen Node, if Node is not available yet	**File > Copy EDS**	*EDS files*	*35*
3	Select Hilscher CANopen Master	**Insert > Master**	*Insert Master*	*36*
4	Select CANopen Node and set Node address	**Insert > Node**	*Insert Node*	*40*
5	Set PDO	Left mouse click at the Node, then **Settings > Node Configuration**	*Node Configuration*	*42*
6	Set Offset address **(*1)**			
7	Set Bus Parameter	Left mouse click at the Master, then **Settings > Bus Parameter**	*Bus Parameter*	*77*
8	Set Device Assignment, if no automatic assignment has occurred	Left mouse click at the Master, then **Settings > Device Assignment**	*Device Assignment*	*67*
9	Save project	**File > Save**	*Save and Save As*	*126*
10	Download	Left mouse click at the Master, then **Online > Download**	*Downloading the Configuration*	*92*
11	Live List	Left mouse click at the Master, then **Online > Live List**	*Live List*	*98*
12	Start Debugger	Left mouse click at the Master, then **Online > Start Debug Mode**	*Debugmode (CANopen)*	*99*
13	Device Diagnostic	Left mouse click at the Node, then **Online > Device Diagnostic**	*CANopen Node specific Diagnostic*	*100*
14	Stop Debugger	**Online > Stop Debug Mode**	*Debugmode (CANopen)*	*99*
15	Global Diagnostic	Left mouse click at the Master, then **Online > Global State Field**	*Global State Field*	*103*
16	Transfer user data: Send data, Receive data	Left mouse click at the Master, then **Online > I/O Monitor**	*I/O-Monitor* or **(*2)** alternatively: *I/O Watch*	*108* *109*

Table 8: Configuration Hilscher CANopen Master to any CANopen Node (PDO)

Notes see next page.

Note (*1): The Offset addresses assigned in the Node configuration are always related to the Hilscher DP 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.

3.2.2 Configuration Hilscher CANopen Node to any CANopen Master (PDO)

The following table describes the steps to configure a Hilscher CANopen Node to any CANopen Master for PDO communication, as it is typical for many cases.

#	Action	Menu in the System Configurator	Detail information in section	Page
1	Create a new project	**File > New > CANopen**	*Setting up the CANopen Configuration*	*35*
2	Select Hilscher CANopen Master **(*1)**	**Insert > Master**	*Insert Master*	*36*
3	Select Hilscher CANopen Node and set Node address	**Insert > Node**	*Insert Node*	*40*
4	Set Bus Parameter	Left mouse click at the Master, then **Settings > Bus Parameter**	*Bus Parameter*	*77*
5	Set Device Assignment, if no automatic assignment has occurred	Left mouse click at the Node, then **Settings > Device Assignment**	*Device Assignment*	*67*
6	Save project	**File > Save**	*Save and Save As*	*126*
7	Download	Left mouse click at the Node, then **Online > Download**	*Downloading the Configuration*	*92*
8	PDO diagnostic	Left mouse click at the Node, then **Online > Extended Device Diagnostic > COS_TASK PDO Transfer**	*COS_TASK PDO Transfer*	*172*
9	Transfer user data: Send data, Receive data	Left mouse click at the Master, then **Online > I/O Monitor**	*I/O-Monitor* or **(*2)** alternatively: *I/O Watch*	*108* *109*

Table 9: Configuration Hilscher CANopen Node to any CANopen Master (PDO)

Note: The Hilscher CANopen Node is configured via the CANopen Bus by means of SDO download by a configuration master. Without a configuration master the Hilscher CANopen Node provides two send- and two receive-PDOs with a default mapping for the communication.

Note (*1): Insert a Hilscher CANopen Master into the configuration. It serves as dummy and it does not have to agree with the connected 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.

3.2.3 Configuration Hilscher CANopen Master to a Hilscher CANopen Node (PDO)

The following table describes the steps to configure a Hilscher CANopen Master to a Hilscher CANopen Node for PDO communication, as it is typical for many cases.

#	Action	Menu in the System Configurator	Detail information in section	Page
1	Create a new project	**File > New > CANopen**	*Setting up the CANopen Configuration*	*35*
2	Select Hilscher CANopen Master	**Insert > Master**	*Insert Master*	*36*
3	Select Hilscher CANopen Node and set Node address	**Insert > Node**	*Insert Node*	*40*
4	Set PDO	Left mouse click at the Node, then **Settings > Node Configuration**	*Node Configuration*	*42*
5	Set Offset address **(*1)**			
6	Set Bus Parameter	Left mouse click at the Master, then **Settings > Bus Parameter**	*Bus Parameter*	*77*
7	Set Device Assignment for the Master, if no automatic assignment has occurred	Left mouse click at the Master, then **Settings > Device Assignment**	*Device Assignment*	*67*
8	Set Device Assignment for the Node, if no automatic assignment has occurred	Left mouse click at the Node, then **Settings > Device Assignment**		
9	Save project	**File > Save**	*Save and Save As*	*126*
10	Download on the Master	Left mouse click at the Master, then **Online > Download**	*Downloading the Configuration*	*92*
11	Download on the Node	Left mouse click at the Node, then **Online > Download**		
12	Live List	Left mouse click at the Master, then **Online > Live List**	*Live List*	*98*
13	Start Debugger	Left mouse click at the Master, then **Online > Start Debug Mode**	*Debugmode (CANopen)*	*99*
14	Device Diagnostic	Left mouse click at the Node, then **Online > Device Diagnostic**	*CANopen Node specific Diagnostic*	*100*
15	Stop Debugger	**Online > Stop Debug Mode**	*Debugmode (CANopen)*	*99*
16	Global Diagnostic	Left mouse click at the Master, then **Online > Global State Field**	*Global State Field*	*103*
17	Transfer user data: Send data, Receive data	Left mouse click at the Master, then **Online > I/O Monitor** Left mouse click at the Node, then **Online > I/O Monitor**	*I/O-Monitor* or **(*2)** alternatively: *I/O Watch* *I/O-Monitor* **(*2)**	*108* *109* *108*

Table 10: Configuration Hilscher CANopen Master to a Hilscher CANopen Node (PDO)

Notes see next page.

Note (*1): The Offset addresses assigned in the Node configuration are always related to the Hilscher DP 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.

3.3 Configuration for SDO Communication

3.3.1 Configuration Hilscher CANopen Master to any CANopen Node (SDO)

The following table describes the steps to configure a Hilscher CANopen Master to any CANopen Node for SDO communication, as it is typical for many cases.

#	Action	Menu in the System Configurator	Detail information in section	Page
1	Create a new project	**File > New > CANopen**	*Setting up the CANopen Configuration*	*35*
2	Copy EDS file of CANopen Node, if Node is not available yet	**File > Copy EDS**	*EDS files*	*35*
3	Select Hilscher CANopen Master	**Insert > Master**	*Insert Master*	*36*
4	Select CANopen Node and set Node address	**Insert > Node**	*Insert Node*	*40*
5	Set Bus Parameter	Left mouse click at the Master, then **Settings > Bus Parameter**	*Bus Parameter*	*77*
6	Set Device Assignment, if no automatic assignment has occurred	Left mouse click at the Master, then **Settings > Device Assignment**	*Device Assignment*	*67*
7	Save project	**File > Save**	*Save and Save As*	*126*
8	Download	Left mouse click at the Master, then **Online > Download**	*Downloading the Configuration*	*92*
9	Live List	Left mouse click at the Master, then **Online > Live List**	*Live List*	*98*
10	Transfer user data: Read objects Write objects	Left mouse click at the Node, then **Online > Read Objects** **Online > Write Objects**	*Read Objects (SDO Upload)* *Write Object (SDO Download)*	*111* *111*

Table 11: Configuration Hilscher CANopen Master to any CANopen Node (SDO)

3.3.2 Configuration Hilscher CANopen Node to any CANopen Master (SDO)

The following table describes the steps to configure a Hilscher CANopen Node to any CANopen Master for SDO communication, as it is typical for many cases.

#	Action	Menu in the System Configurator	Detail information in section	Page
1	Create a new project	**File > New > CANopen**	*Setting up the CANopen Configuration*	*35*
2	Select Hilscher CANopen Master **(*1)**	**Insert > Master**	*Insert Master*	*36*
3	Select Hilscher CANopen Node and set Node address	**Insert > Node**	*Insert Node*	*40*
4	Set Bus Parameter	Left mouse click at the Master, then **Settings > Bus Parameter**	*Bus Parameter*	*77*
5	Set Device Assignment, if no automatic assignment has occurred	Left mouse click at the Node, then **Settings > Device Assignment**	*Device Assignment*	*67*
6	Save project	**File > Save**	*Save and Save As*	*126*
7	Download	Left mouse click at the Node, then **Online > Download**	*Downloading the Configuration*	*92*
8	SDO Diagnostic	Left mouse click at the Node, then **Online > Extended Device Diagnostic**	*COS_TASK SDO Transfer*	*173*
9	Transfer user data: Read objects Write objects	Left mouse click at the Node, then **Online > Message Monitor**	*Message Monitor*	*112*

Table 12: Configuration Hilscher CANopen Node to any CANopen Master (SDO)

Note (*1): Insert a Hilscher CANopen Master into the configuration. It serves as dummy and it does not have to agree with the connected Master.

3.3.3 Configuration Hilscher CANopen Master to a Hilscher CANopen Node

The following table describes the steps to configure a Hilscher CANopen Master to a Hilscher CANopen Node for SDO communication, as it is typical for many cases.

#	Action	Menu in the System Configurator	Detail information in section	Page
1	Create a new project	**File > New > CANopen**	*Setting up the CANopen Configuration*	*35*
2	Select Hilscher CANopen Master	**Insert > Master**	*Insert Master*	*36*
3	Select CANopen Node and set Node address	**Insert > Node**	*Insert Node*	*40*
4	Set Bus Parameter	Left mouse click at the Master, then **Settings > Bus Parameter**	*Bus Parameter*	*77*
5	Set Device Assignment for the Master, if no automatic assignment has occurred	Left mouse click at the Master, then **Settings > Device Assignment**	*Device Assignment*	*67*
6	Set Device Assignment for the Node, if no automatic assignment has occurred	Left mouse click at the Node, then **Settings > Device Assignment**		
7	Save project	**File > Save**	*Save and Save As*	*126*
8	Download on the Master	Left mouse click at the Master, then **Online > Download**	*Downloading the Configuration*	*92*
9	Download on the Node	Left mouse click at the Node, then **Online > Download**		
10	Live List	Left mouse click at the Master, then **Online > Live List**	*Live List*	*98*
11	Transfer user data: Read objects, Write objects	Left mouse click at the Node, then **Online > Read Objects** **Online > Write Objects** Left mouse click at the Node, then **Online > Message Monitor**	*Read Objects (SDO Upload)* *Write Object (SDO Download)* *Message Monitor*	*111* *111* *112*

Table 13: Configuration Hilscher CANopen Master to a Hilscher CANopen Node

3.4 Configuration for Send/Receive transparently (CAN)

3.4.1 Configuration Hilscher CANopen Master to any CAN Device for Send/Receive transparent (CAN)

The following table describes the steps to configure a Hilscher CANopen Master for send/receive CAN telegrams (Layer 2) transparently, as it is typical for many cases.

#	Action	Menu in the System Configurator	Detail information in section	Page
1	Create a new project	**File > New > CANopen**	*Setting up the CANopen Configuration*	*35*
2	Select Hilscher CANopen Master	**Insert > Master**	*Insert Master*	*36*
3	Set Bus Parameter	Left mouse click at the Master, then **Settings > Bus Parameter**	*Bus Parameter*	*77*
4	Set Device Assignment for the Master, if no automatic assignment has occurred	Left mouse click at the Master, then **Settings > Device Assignment**	*Device Assignment*	*67*
5	Save project	**File > Save**	*Save and Save As*	*126*
6	Download on the Master	Left mouse click at the Master, then **Online > Download**	*Downloading the Configuration*	*92*
7	Transfer user data: Send CAN Telegrams Receive CAN Telegrams **(*1)**	Left mouse click at the Master, then **Online > Message Monitor**	*Message Monitor for Sending CAN Telegrams (transparent)* *Message Monitor for Receiving CAN Telegrams (transparent)*	*120* *122*

Table 14: Configuration Hilscher CANopen Master to any CAN Device for Send/Receive transparent (CAN)

Note (*1): The information, which CAN Telegram should receive Identifier, is activated per message.

3.4.2 Configuration Hilscher CANopen Node to any CAN Device for Send/Receive transparently (CAN)

The following table describes the steps to configure a Hilscher CANopen Node for send/receive CAN telegrams (Layer 2) transparently, as it is typical for many cases.

#	Action	Menu in the System Configurator	Detail information in section	Page
1	Create a new project	**File > New > CANopen**	*Setting up the CANopen Configuration*	*35*
2	Select Hilscher CANopen Master **(*1)**	**Insert > Master**	*Insert Master*	*36*
3	Select Hilscher CANopen Node	**Insert > Node**	*Insert Node*	*40*
4	Set Bus Parameter	Left mouse click at the Master, then **Settings > Bus Parameter**	*Bus Parameter*	*77*
5	Set Device Assignment for the Node, if no automatic assignment has occurred	Left mouse click at the Node, then **Settings > Device Assignment**	*Device Assignment*	*67*
6	Save project	**File > Save**	*Save and Save As*	*126*
7	Download on the Node	Left mouse click at the Node, then **Online > Download**	*Downloading the Configuration*	*92*
8	Transfer user data: Send CAN Telegrams Receive CAN Telegrams **(*2)**	Left mouse click at the Node, then **Online > Message Monitor**	*Message Monitor for Sending CAN Telegrams (transparent)* *Message Monitor for Receiving CAN Telegrams (transparent)*	*120* *122*

Table 15: Configuration Hilscher CANopen Node to any CAN Device for Send/Receive transparently (CAN)

Note (*1): Insert a Hilscher CANopen Master into the configuration. It serves as dummy and it does not have to agree with the connected Master.

Note (*2): The information, which CAN Telegram should receive Identifier, is activated per message.

4 Configuration of CANopen with SyCon

4.1 Setting up the CANopen Configuration

To create a new configuration, choose the **File > New** menu. This will offer a selection list of fieldbus systems. Select **CANopen**. If only the CANopen fieldbus system is installed, 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

Each CANopen device manufacturer defines the CANopen characteristics of its device in a so called Electronic Data Sheet, also called EDS file. This description files form the basis of the configuration.

Devices	EDS files
Hilscher devices	The EDS files for Hilscher devices are already included in the delivery of the System Configurator SyCon.
Devices from other manufacturers	For other devices these have to be delivered by the device manufacturer.

Table 16: EDS files - Source of Supply

During the program start the System Configurator reads in automatically all EDS files, which are put down in the EDS directory. In this act the device names are taken up to an internal list. The device-specific data are read out during the configuration directly from the EDS file.

If a CANopen Node (Slave) is needed, which does not appear yet in the selection list, then the appropriate EDS file can be copied in the EDS directory with the menu **File > Copy EDS**. Another possibility is to copy the EDS file with the Windows Explore into the SyCon EDS directory and then read in the EDS files in the EDS directory again with the menu **Settings > Path**.

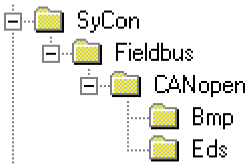


Figure 7: EDS files and bitmaps directory

The EDS path is changeable. The standard setting can be changed with the menu **Settings > Path**.

4.3 Master

4.3.1 Insert Master

In order 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:

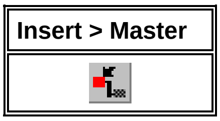
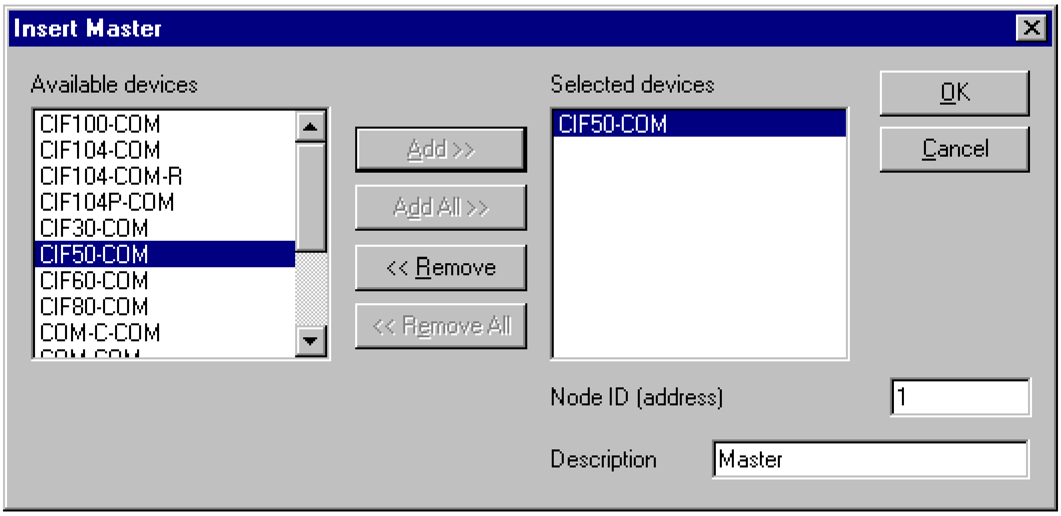


Figure 8: Insert > Master Symbol

A window appears where you can select one master device.



Available devices

CIF100-COM
CIF104-COM
CIF104-COM-R
CIF104P-COM
CIF30-COM
CIF50-COM
CIF60-COM
CIF80-COM
COM-C-COM

Selected devices

CIF50-COM

Node ID (address) 1

Description Master

Figure 9: Insert > Master

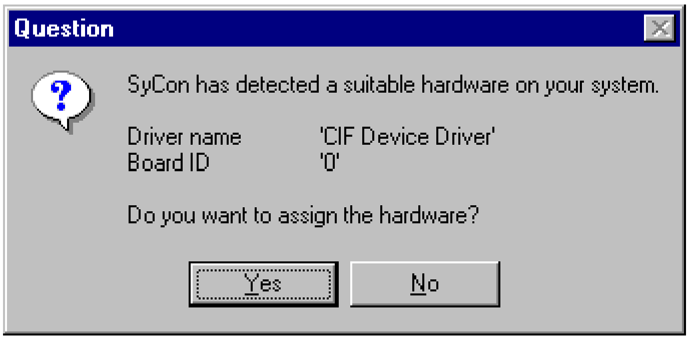
In this window you select the Master you want by clicking on it in the list **Available devices** and then click the **Add** button or make a double click to put the Master in the list **Selected devices**. With **OK** you confirm the selection and the Master will be insert.

This example shows a CIF 50-COM with the **Description** Master, which is changeable in this field.

The **Node ID (address)** is necessary for the addressing of the device at the bus and has to be unique. Therefore it is not allowed to use this number two times in the same network.

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.



SyCon has detected a suitable hardware on your system.

Driver name 'CIF Device Driver'
Board ID '0'

Do you want to assign the hardware?

Figure 10: 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** (in section *Device Assignment* at page *67*).

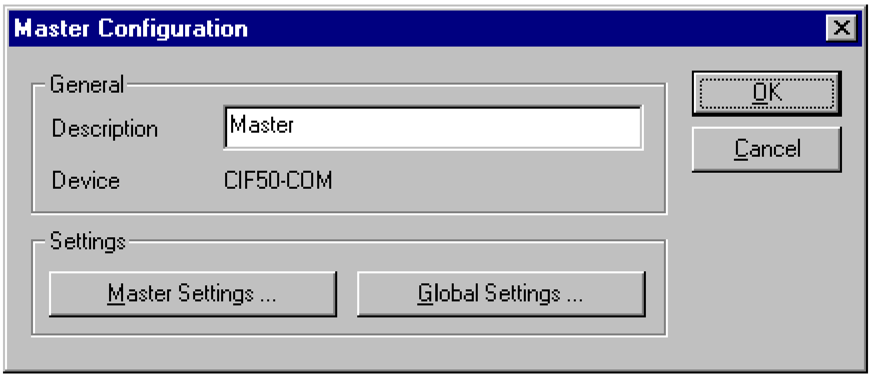
4.3.2 Master Configuration

The Master specific configuration is carried out in the following window and sub window.

Set the focus on the Master (left mouse click) and then select the **Settings > Master Configuration** menu

or

A double click on the symbol of the Master which should be configured will open the following window.



Description: Master

Device: CIF50-COM

Figure 11: Settings > Master Configuration

The following can be set in this Master Configuration window:

- a (symbolic) **Description** of the Master
- the window *Master Settings* (described on page *80*) can be opened
- the window *Global Settings* (described on page *84*) can be opened

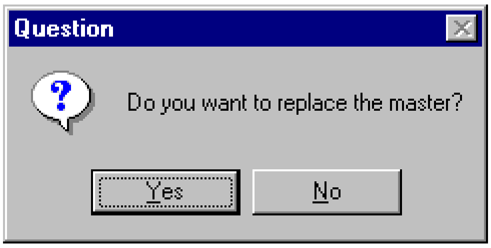
4.3.3 Replace Master

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) and then select the menu **Edit > Replace**.

or

make a right mouse click at the Master and select in the now opened window the menu **Replace**.

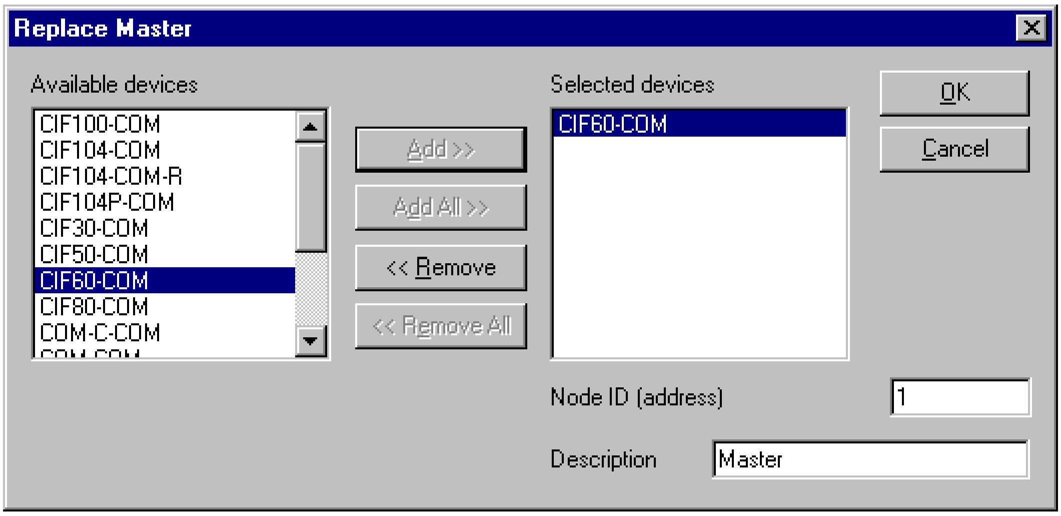
In the opened window appears the question if the Master should be replaced.



Do you want to replace the master?

Figure 12: Security question Replace Master

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



Selected devices: CIF60-COM

Node ID (address) 1

Description Master

Figure 13: Edit > Replace Master

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

4.4 Node (Slave)

4.4.1 Insert Node

In order to insert a CANopen Node into the configuration, select the **Insert > Node** menu to open the selection window, or click on the symbol:

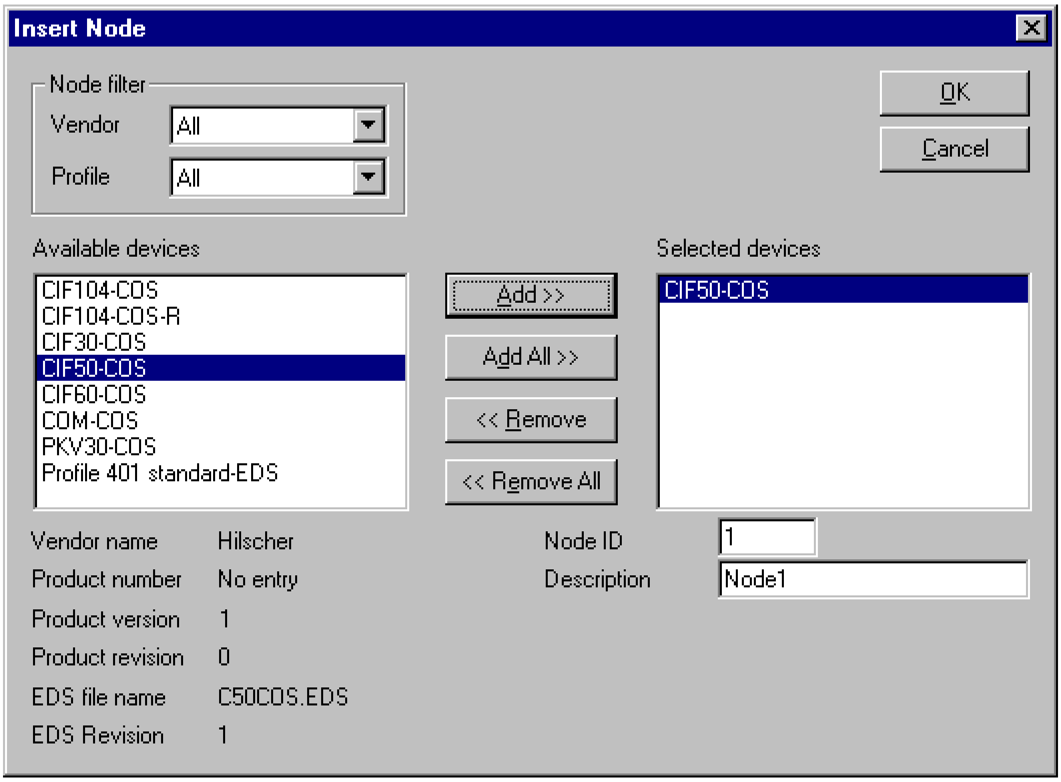
Insert > Node

Figure 14: Insert > Node

The mouse cursor changes automatically to the insert Node cursor. Click on the position where you want to insert the new Node. A dialogue box appears where you can select one or more Nodes for insertion.

Mouse pointer for Insert> Node
N

Figure 15: Mouse pointer for Insert > Node



Node filter
Vendor: All
Profile: All

Available devices
- CIF104-COS
- CIF104-COS-R
- CIF30-COS
- CIF50-COS
- CIF60-COS
- COM-COS
- PKV30-COS
- Profile 401 standard-EDS

Selected devices
- CIF50-COS

Vendor name: Hilscher
Product number: No entry
Product version: 1
Product revision: 0
EDS file name: C50COS.EDS
EDS Revision: 1

Node ID: 1
Description: Node1

Figure 16: Insert > Node

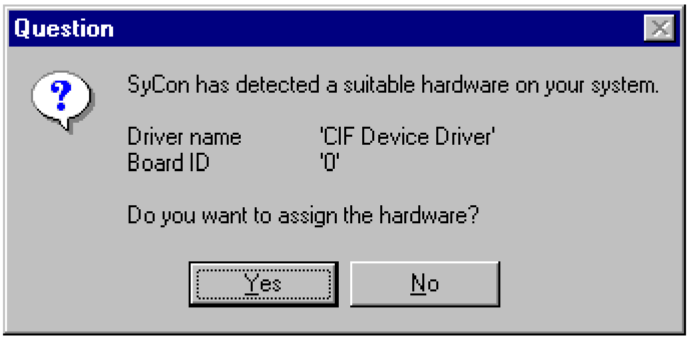
The left list all available Node devices are shown which are present in the EDS directory. A filter can be used to limit the selection list via the **Vendor** and the **Profile**. If one Node is selected there you can see some additional information about that Node below the list box.

With a double click or with the button **Add**, the Node appears in the list **Selected devices**. When a new Node is chosen SyCon always looks for the next free Node ID value and propose it. If you select each Node by each you can change its Node ID and give it a short description in the field **Description**. The Description field will accept up to 32 characters of text.

It is possible to configured an available Node multiple times with different **Node ID**s. In CANopen the Node address is called Node ID. The Node ID distinguishes the different Nodes from each other in the network. It's a unique number that can't be forgiven twice. Therefore your made entry in the field **Node ID** must be equivalent to the real Node ID itself, else the master will get no contact later to the Node when it wants to establish the communication.

4.4.1.1 Hardware Assignment

If you have configured the CIF Device Driver Setup for your hardware and you insert the correct Node 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.



Question

SyCon has detected a suitable hardware on your system.

Driver name 'CIF Device Driver'
Board ID '0'

Do you want to assign the hardware?

Figure 17: Hardware Assignment Node

If you answer with **Yes**, the Hardware is assigned. If you answer with **No** you have to assign this hardware with **Settings > Device Assignment** (in section *Device Assignment* at page *67*).

4.4.2 Node Configuration

At first you have to make a left mouse click on the symbol of the Node and select the menu **Settings > Node Configuration**.

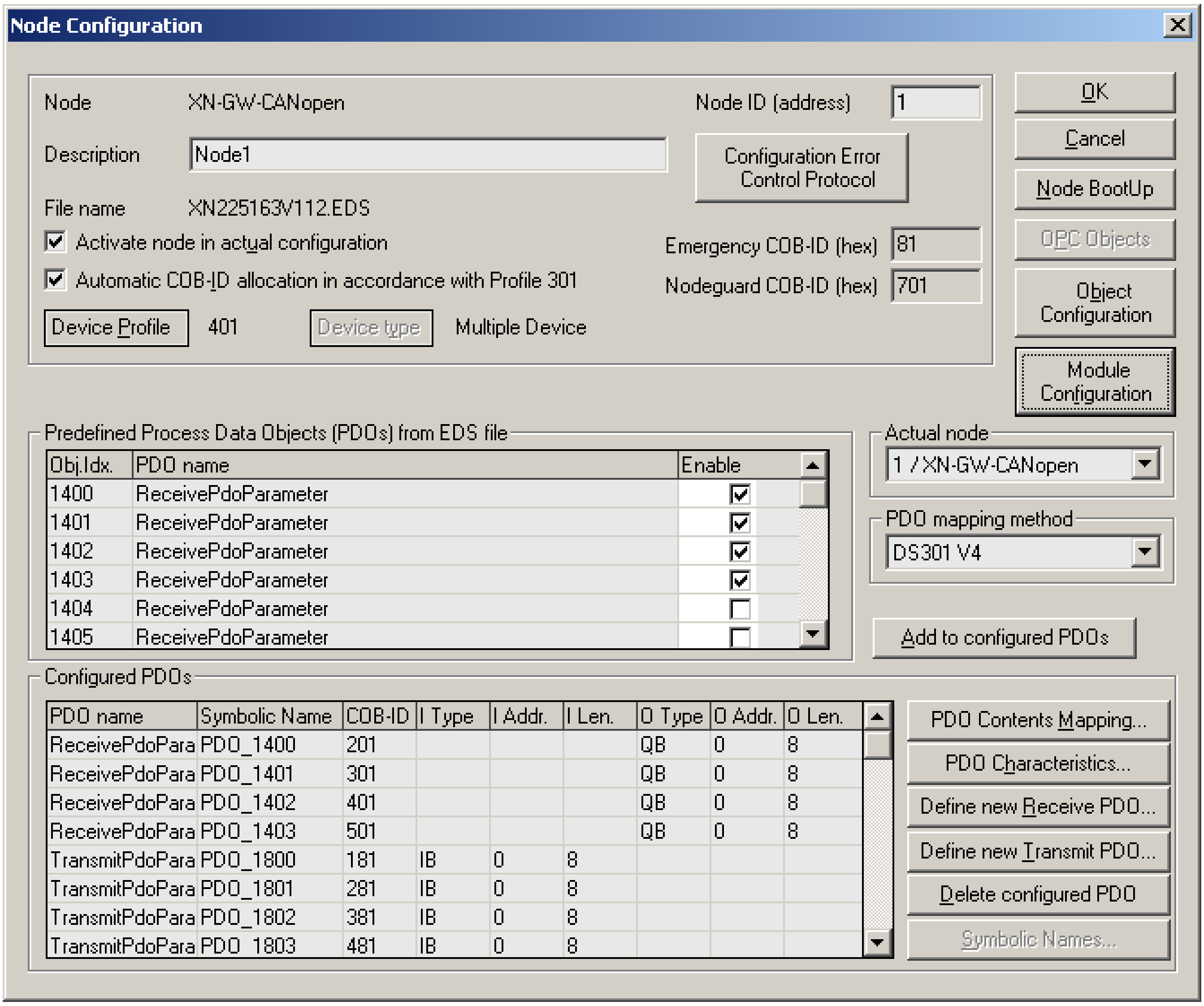
or

Make a double click on the CANopen Node to open the Node Configuration window.

The Node specific configuration is carried out in this window. Here the PDO (Process data objects) and their addresses in the process data image are assigned in the Hilscher Master. Please note, that the addresses have to agree with the addresses in the PC application program.

Note 1 (Hilscher Master): The information about the Offset addresses relate to the addressing of the data in the Master! The addresses don't relate to the addressing of the data in the Node. The Node organizes its data addressing itself.

Note 2 (Hilscher Node): In case of a Hilscher Node (Slave) the In- and Output data are taken over at the bus directly in the Dual-port memory. The Offset addresses relate to the Master.



Node XN-GW-CANopen

Node ID (address) 1

Description Node1

File name XN225163V112.EDS

☑ Activate node in actual configuration

☑ Automatic COB-ID allocation in accordance with Profile 301

Emergency COB-ID (hex) 81

Nodeguard COB-ID (hex) 701

Device Profile 401

Device type Multiple Device

Predefined Process Data Objects (PDOs) from EDS file

Obj.Idx.	PDO name	Enable
1400	ReceivePdoParameter	☑
1401	ReceivePdoParameter	☑
1402	ReceivePdoParameter	☑
1403	ReceivePdoParameter	☑
1404	ReceivePdoParameter	☐
1405	ReceivePdoParameter	☐

Actual node 1 / XN-GW-CANopen

PDO mapping method DS301 V4

Configured PDOs

PDO name	Symbolic Name	COB-ID	I Type	I Addr.	I Len.	O Type	O Addr.	O Len.
ReceivePdoPara	PDO_1400	201				QB	0	8
ReceivePdoPara	PDO_1401	301				QB	0	8
ReceivePdoPara	PDO_1402	401				QB	0	8
ReceivePdoPara	PDO_1403	501				QB	0	8
TransmitPdoPara	PDO_1800	181	IB	0	8			
TransmitPdoPara	PDO_1801	281	IB	0	8			
TransmitPdoPara	PDO_1802	381	IB	0	8			
TransmitPdoPara	PDO 1803	481	IB	0	8			

Figure 18: Settings > Node Configuration

The following table shows the fields and elements of the Node Configuration window.

- **Node**

  The name of the device coming from the EDS file is shown in the field **Node**.

- **Description**

  The field **Description** contains a symbolic name for the Node.

- **File name**

  File name of the EDS file.

- **Activate Node in actual configuration**

  If **Activate Node in actual configuration** is selected, process memory in the Master is reserved for this Node and the Master makes a data exchange at the bus to this Node. If this setting is deactivated, the Master reserves memory in the process data image for this Node, but no data exchange to this Node is made at the bus.

- **Automatic COB-ID allocation in accordance with Profile 301**

  In the basic setting **Automatic COB-ID allocation in accordance with Profile 301** is activated. Then the COB-ID is presetted for a PDO depending on the Node address and depending on the used PDO. If this field is deactivated, a manual assignment can be done.

  In order to reduce configuration effort for simple networks a mandatory default identifier allocation scheme is defined, which is described in section *COB-ID* at page *176*. These identifiers are available in the Pre-operational state of a Node which works in accordance to the Communication Profile 301 directly after initialization. These pre-defined connection sets are used by SyCon if automatic allocation is enabled. Then the COB-IDs in the already configured PDO COB-ID column are not editable. So if the automatic allocation is disabled the COB-IDs can be edited in the range from 0 –2047.

Note: If the setting Automatic COB-ID allocation in accordance with Profile 301 is deactivated, SyCon does not check if a COB-ID was assigned two times. This is the job of the user. Further more you have to check, if the Node supports this function.

- **Device Profile and Device Type**

  Because of the information of the Device Profile and the Device Type the Master can read out the Object 1000H from the Node and compare it with this information when it start communication.

  If the Device Profile and the Device Type do not agree the Master reports a parameterization error.

  Further information about the Device Profile and the Device Type you find in section *Device Profile and Device Type* at page *47*.

- **NodeID (address)**

  The **NodeID (address)** is necessary for the addressing of the device at the bus and has to be unique. Therefore it is not allowed to use this number two times in the same network. And it has to agree with the set Node address of the device. Otherwise it is not possible for the Master to build up a communication to this device.

  The NodeID (address) also fixes the used COB-ID.

- **Configuration Error Control Protocol**

  The **Configuration Error Control Protocol** makes for setting the device control via the Node Guarding, Live Guarding and respectively Heartbeat.

  The **Guard Time** and the **Live Time Factor** for the Node Guarding are set in the **Configuration Error Control Protocol** window. Further information is described in section *Configuration Error Control Protocol* on page *60*.

  The **Node Heartbeat Producer Time** and the **Master Guarding Time of Node** are set in the **Configuration Error Control Protocol** window. Further information is described in section *Configuration Error Control Protocol* on page *60*.

- **Emergency COB-ID**

  Is an information of the COB-ID of the Emergency telegram.

- **Nodeguard COB-ID**

  Is an information of the COB-ID of the Nodeguard telegram.

- **OK**

  To close the Node Configuration window and to take over the settings.

- **Cancel**

  To close the Node Configuration window and to reject the settings.

- **Node BootUp**

  The Node BootUp defines the start up behaviour of the Master with regard to each individual Node and is described in section *Node BootUp* at page *58*.

- **OPC Objects**

  The information in the OPC Objects field relate to the symbols of the OPC server and the SDO communication. Further information you find in the manual for the OPC server.

- **Object Configuration**

  Via the button Object Configuration the object directory can be read out from the EDS file and if necessary added to the Node configuration. Further information you find in section *Object Configuration* at page *64*.

- **Module Configuration**

  Via the button module configuration the modules of modular nodes can be displayed (read from the EDS file) and mappable objects can be added for the configuration of the node. Further information you find in section *Module Configuration* at page *48*.

- **Actual Node**

  Changes to the Node configuration of another Node without leaving the window.

- **PDO mapping method**

  Lays down the procedure of the PDO mapping. You can select between the methods DS301 V4 and DS301 V3. The difference between this methods are described in section *PDO Mapping Method* at page *187*.

- **Predefined Process Data Objects (PDOs) from the EDS file**

  Shows the list of the PDOs which are given in the EDS file and which can be used for the configuration. Further information you find in section *Process Data Configuration - Selection of PDO* at page *49*.

- **Enable**

  The checkbox **Enable** indicates which PDOs of the predefined PDOs are configured. By activating a PDO this is inserted into the table configured PDOs and by deactivating this selection the PDO is removed from this table.

- **Configured PDOs**

  Shows the PDOs which are used for the data exchange between Master and Node. In addition to the Offsets in the process data image also the length of the PDOs is indicated. Further information you find in section *Process Data Configuration - Selection of PDO* on page *49*.

- **Add to configured PDOs**

  By clicking on a PDO in the list **Configured PDOs** and afterwards a click on the button **Add to configured PDOs** the selected PDO is taken over in the list of **Configured PDOs**.

- **PDO Contents Mapping**

  First you have to select a PDO in the list **Configured PDOs**. By making a double click or a click at the **PDO Contents Mapping** button the in the PDO transferred user data can be shown and the combination can be changed if necessary. Further information you find in section *PDO Contents Mapping* on page *57*.

- **PDO Characteristics**

  First you have to select a PDO in the list **Configured PDOs**. With a click at the **PDO Characteristics** button the transmission settings of the PDO can be shown and adjusted if necessary.

- **Define new Receive PDO**

  By clicking on this button a new Receive PDO is added to the **Configured PDOs**. This is described in section *Creating own Receive PDOs* at page *53*.

- **Define new Transmit PDO**

  By clicking on this button a new Transmit PDO is added to the **Configured PDOs**. This is described in section *Creating own Transmit PDOs* at page *56*.

- **Delete configured PDO**

  To delete a configured PDO you first have to select the PDO in **Configured PDOs** and then click on the **Delete configured PDO** button.

- **Symbolic Names**

  The information at **Symbolic Names** relate to the symbols for the OPC server. Further information you find in the manual for the OPC server.

4.4.2.1 Overview Node Configuration

For the Node Configuration to transfer PDO data the following typical steps have to be made.

Configuration step	**Description**
Device Profile and Device Type	Set or take over the value which is read out of the EDS file
Module Configuration	Select the modules of the node
Process Data Configuration	Select the PDO
Process Data Configuration	Set the PDO transmission characteristics
PDO Mapping	Take over the basic setting or adjust the PDO combination
Node BootUp	Set startup behaviour
Node supervision	Set Nodeguarding and/or Lifeguarding

Table 17: Overview Node Configuration

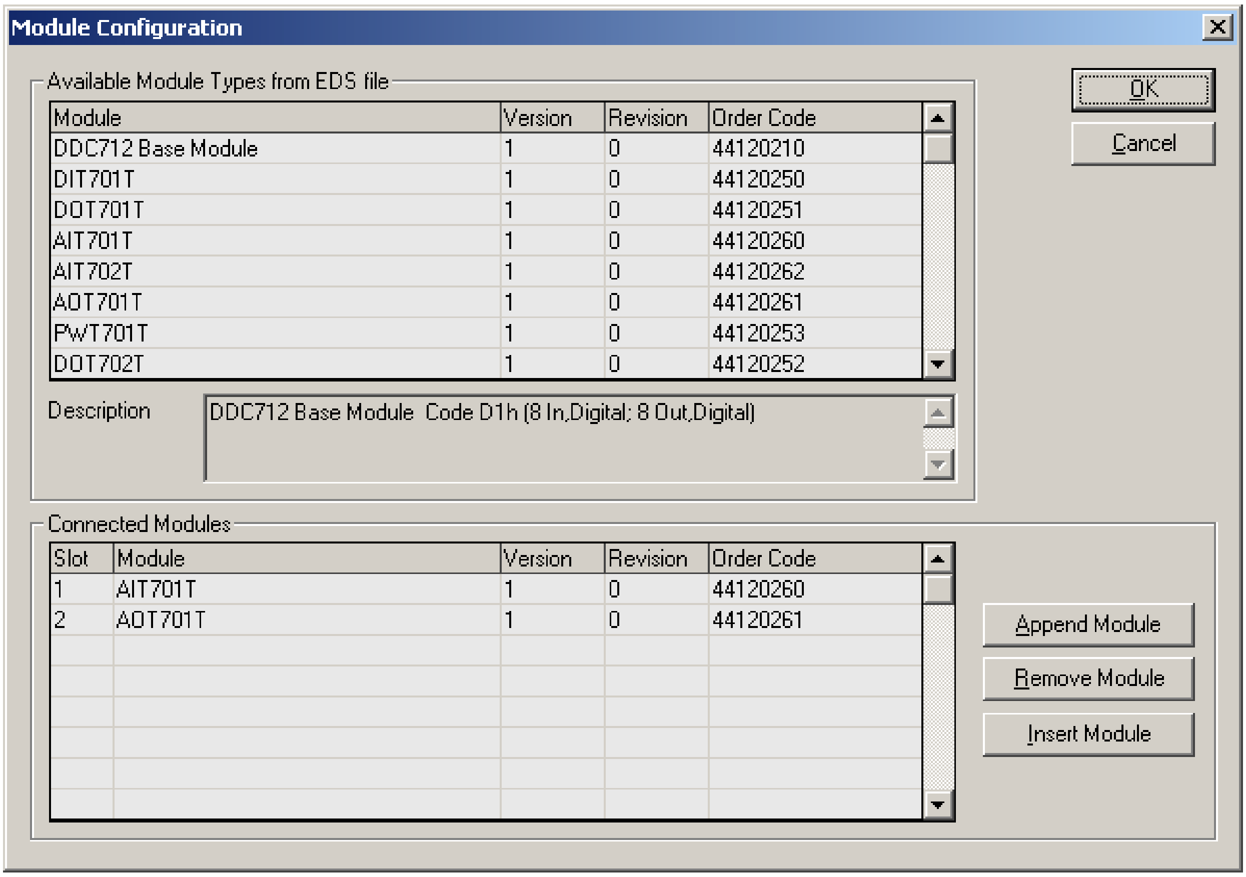
4.4.2.2 Device Profile and Device Type

Each CANopen Node has a mandatory Object 1000H, which has to be existing in the object directory. This object is named Device Type. The Device Type also includes the information about the Device Profile.

The Master reads out the Object 1000H from the Node when starting up the CANopen bus and compares the entries, which are made in the two available fields **Device Profile** and **Device Type**. If the Device Profile and the Device Type do not agree, the Master reports a parameterization error and does not establish a process data transfer to the Node. To get the real values of the Node, use the online function **Online > Read Object** or click on the Node in Debug mode.

4.4.2.3 Module Configuration

Some CANopen Nodes support the configuration by modules. That means a user can define which modules are used for the node.



Available Module Types from EDS file

Module	Version	Revision	Order Code
DDC712 Base Module	1	0	44120210
DIT701T	1	0	44120250
DOT701T	1	0	44120251
AIT701T	1	0	44120260
AIT702T	1	0	44120262
AOT701T	1	0	44120261
PWT701T	1	0	44120253
DOT702T	1	0	44120252

Description: DDC712 Base Module Code D1h (8 In,Digital; 8 Out,Digital)

Connected Modules

Slot	Module	Version	Revision	Order Code
1	AIT701T	1	0	44120260
2	AOT701T	1	0	44120261

Table 18: Module Configuration

The available modules are listed below **Available Module Types from EDS file**. A description of a module is shown at **Description** when a module is marked.

Mark the module to be used in the configuration in the upper list and append it to the end of the **Connected Modules** list with a click on the button **Append Module**. A module from the upper list can be inserted at the top or in the middle of the **Connected Modules** list. Therefore mark the module in the **Connected Modules** list where the the module from the upper list should be inserted before, then mark the module to be inserted in the upper list and then click the button **Insert Module** to insert the module to the **Connected Modules** list.

The modules in the **Connected Modules** list add mappable objects for the PDO mapping for the configuration of the node. These mappable objects can be used to configure the node and is described in section *PDO Contents Mapping - Arrange a PDO* at page *57*.

The column **Slot** shows a counter for for the number of used modules.

4.4.2.4 Process Data Configuration - Selection of PDO

The process data are transmitted via process data objects, short PDOs, and assigned to the process data image. CANopen distinguishes between receive- and send PDOs.

Receive PDOs	Send PDOs
Data from the Master to the Node Output data are processed by the Node	Data from the Node (Slave) to the Master Input data are generated by the Node (Slave)

Table 19: PDO: Send PDO and Receive PDO

The data of the Node in the process data image of the Master are serviced for the application with the configuration of the PDOs.

The configuration window contains two tables. The upper table **Predefined Process Data Objects (PDOs) from EDS file** shows all configurable PDOs, which are predefined in the EDS file of the device. By making a double click on a table entry or via the **Add to configured PDOs** button the entry is taken over in the table **Configured PDOs**.

The columns of the table **Configured PDOs** have the following meaning:

- **PDO name**

  Here the RxPDO parameter and TxPDO parameter are shown.

- **Symbolic Name**

  Here the symbolic name, which is used in case of OPC communication, is given. PDO_1400 and PDO_1800 and continuous names are used as pre-set value. This can be overwritten by the user.

- **COB-ID**

  In this column the CAN telegram identifier is shown. In case of automatic award of COB-ID the routine described in section *COB-ID (Predefined Connection Set)* on page *176* is used. In case of manual award the telegram identifier of the CAN telegram which is transmitted with the PDO can be edit in the range from 0 to 2047.

- **I Type and O Type**

  The specification IB stands for Input Byte and the specification QB stands for Output Byte.

- **I Addr. and O Addr.**

  The **I Addr.** (Input Address) and the **O Addr.** (Output Address) define the address of the PDO data in the process data image, which is lead in the Dual-port memory of the Master. The range can be between 0 and 3583. According to information the number of data bytes is shown under **I Number** and **O Number**.

  The addresses can be assigned automatically by SyCon or manually by the user. This is set in the menu **Settings > Global Settings** in the field **Process Data Auto Addressing**, which is described in section *Global Settings* at page *84*. A screening for double addresses takes place before the Download of Configuration and when you open the window **Address Table**.

- **I Len. and O Len.**

  Gives the length of the PDO in bytes and can be max 8. If the value 0 is shown, the PDO still does not includes user data. Via the PDO Mapping the user data for this PDO have to be set.

4.4.2.5 PDO Communication Parameter (PDO Characteristic)

Before a chosen PDO is moved into the lower window, the **PDO characteristics** window is opened automatically.

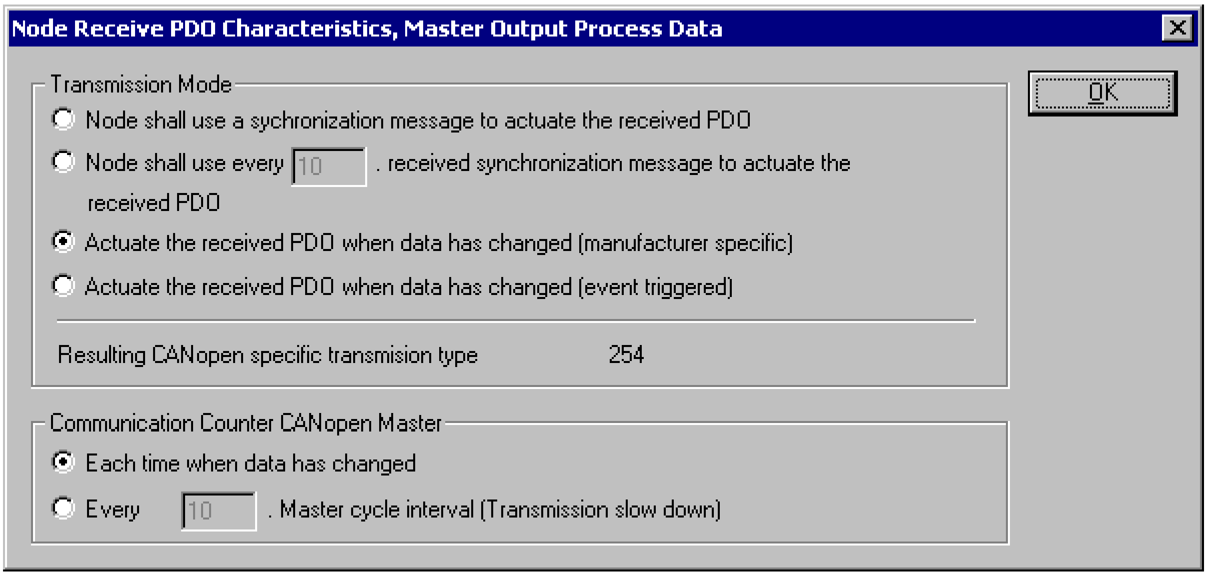
A PDO in CANopen can be configured in Event Driven mode or Cyclic Transmission. Both kinds of transmission types can be synchronized to a special synchronization message which is sent by the master in defined time intervals. Because of the different behaviour of a transmit and receive PDO, two different windows will be open during the PDO insertion. The several transmissions are distinguished in the so-called **Transmission type** value.

Synchronous means that the transmission of the PDO shall be related to the SYNC message that is sent cyclically by the Master. Preferably the Nodes use the SYNC message as a trigger to output or actuate based on the previous synchronous Receive-PDO respectively to update the data transmitted at the following synchronous Transmit-PDO. Details of this mechanism depend on the device type and are defined in the device profile.

Asynchronous means that the transmission of the PDO is not related to the SYNC message and can happen at any time.

4.4.2.6 Receive PDO characteristics

Receive PDO are output data of the Master and where received from the Node. One calls this PDOs therefore Receive PDOs from view of the Node.



Node Receive PDO Characteristics, Master Output Process Data

Transmission Mode

○ Node shall use a sychronization message to actuate the received PDO

○ Node shall use every 10 . received synchronization message to actuate the received PDO

● Actuate the received PDO when data has changed (manufacturer specific)

○ Actuate the received PDO when data has changed (event triggered)

Resulting CANopen specific transmision type 254

Communication Counter CANopen Master

● Each time when data has changed

○ Every 10 . Master cycle interval (Transmission slow down)

Figure 19: Receive PDO Parameter

Transmission Type	cycl.	acycl.	synchronous	asynchronous	RTR	Description
0		X	X			The telegram is transferred related to the SYNC, but not periodically.
1..240	X		X			A value between 1 and 240 means that the PDO is transferred synchronously and periodically and that the value of the type of transmission shows the number of SYNC telegrams between the two transferring PDOs.
241..251			res.			reserved
254				X		Type of transmission 254 means that the application event is manufacturer dependent.
255				X		The Transmission type 255 means that the application event is defined in the respectively supported equipment profile. The exact transmission mode whether cyclically, event-controlled etc. can reread there.

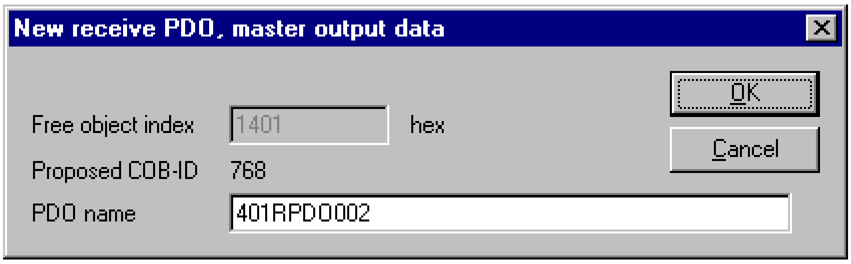
Table 20: PDO Communication Parameter > Transmission Types (Receive PDO)

The Master has two configuration possibilities to send the PDO:

- On the one hand there is the selection **Each time when data has changed**, which configures the Master in such a way, that the Master sends the PDO only if it has changed. This kind of the event control keeps the bus load low.
- On the other hand there is the possibility to transmit the PDO cyclic. However this time is indicated here not in milliseconds, but in Node cycle intervals. A Node cycle interval is the time the Master needs to test all configured PDOs in their states and to process them once. The smallest cycle interval is indicated with about 300µsec.

4.4.2.7 Creating own Receive PDOs

If further PDOs shall be used, which are not predefined in the table **Predefined Process Data Objects (PDOs) from EDS file**, this can be done with the functions **Define new Receive PDO**.



New receive PDO, master output data

Free object index 1401 hex

Proposed COB-ID 768

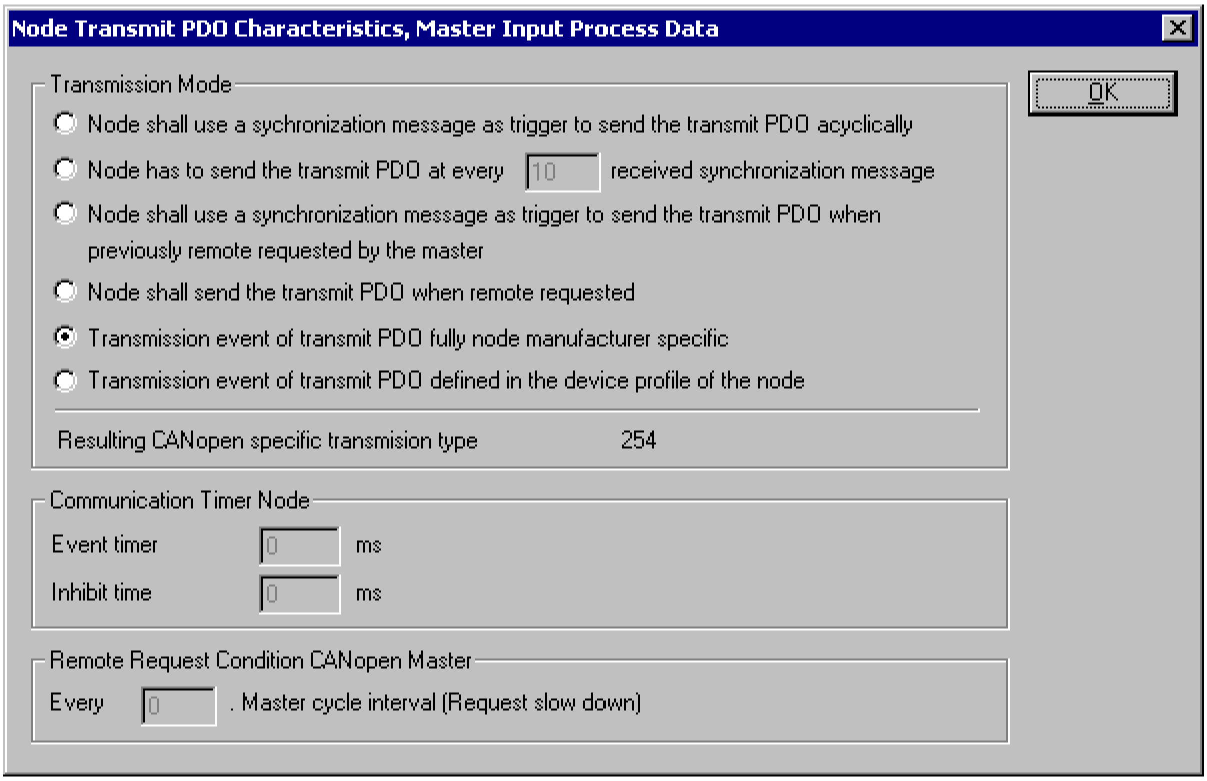
PDO name 401RPDO002

Figure 20: Definite a new receive PDO

SyCon suggests a free message number, which can be edited later in the PDO configuration window, if the PDO was taken over.

4.4.2.8 Transmit PDO characteristics

Transmit PDOs are input data of the Masters and they were sent by the Node. These PDOs are called Transmit PDOs from view of the Node.



Node Transmit PDO Characteristics, Master Input Process Data

Transmission Mode

○ Node shall use a sychronization message as trigger to send the transmit PDO acyclically

○ Node has to send the transmit PDO at every 10 received synchronization message

○ Node shall use a synchronization message as trigger to send the transmit PDO when previously remote requested by the master

○ Node shall send the transmit PDO when remote requested

● Transmission event of transmit PDO fully node manufacturer specific

○ Transmission event of transmit PDO defined in the device profile of the node

Resulting CANopen specific transmision type 254

Communication Timer Node

Event timer 0 ms

Inhibit time 0 ms

Remote Request Condition CANopen Master

Every 0 . Master cycle interval (Request slow down)

Figure 21: Transmit PDO Parameter

Transmission Type	cycl.	acycl.	synchronous	asynchronous	RTR	Description
0		X	X			The telegram is transferred related to the SYNC, but not periodically.
1..240	X		X			A value between 1 and 240 means that the PDO is transferred synchronously and periodically and that the value of the type of transmission indicates the number of SYNC of telegrams between the two transferring PDOs.
241..251			res.			reserved
252			X		X	The transmission types 252 and 253 mean that the PDO is an event without immediate notification and it is only transferred with remote transmission requirement. With the type of transmission 252 the data are immediately updated after receiving the SYNC Telegram (however not sent).
253				X	X	The transmission types 252 and 253 mean that the PDO is an event without immediate notification and it is only transferred with remote transmission requirement. With the type of transmission 253 the data are immediately updated after receiving the SYNC Telegram
254				X		The Transmission type 254 means that the application event is manufacturer dependent.
255				X		The Transmission type 255 means that the application event is defined in the respectively supported equipment profile. The exact transmission mode whether cyclically, event-controlled etc. can reread there.

Table 21: PDO Communication Parameter > Transmission Types (Transmit PDO)

- **Communication Timer Node**

  Event timer: Lays down the time for the event timer for the send PDO. The expiration of the time in the Node is used as event, in order to send the PDO. Manufacturer and/or device-specifically also an application event can activate sending the PDO and reset the event timer.

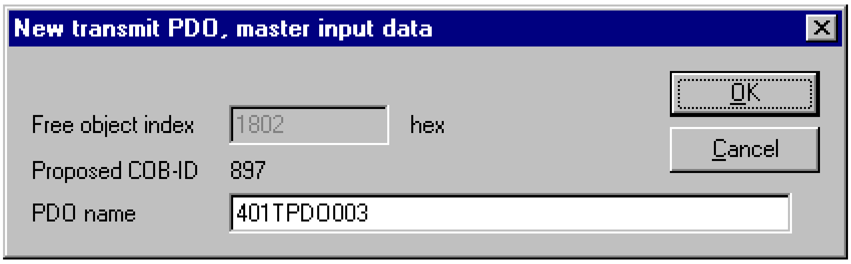
  Inhibit time: The inhibit time describes the time interval, how long at least it must be waited between sending two equal telegrams. Thus a too frequent sending of the same telegram is inhibited.

- **Remote Request Condition CANopen Master**

  Gives the number of internal Master cycles, after that the Master sends a remote request to the Node in order to ask for the PDO.

4.4.2.9 Creating own Transmit PDOs

If further PDOs shall be used, which are not predefined in the table **Predefined Process Data Objects (PDOs) from EDS file**, this can be done with the functions **Define new Transmit PDO**.


New transmit PDO, master input data

Free object index 1802 hex

Proposed COB-ID 897

PDO name 401TPDO003

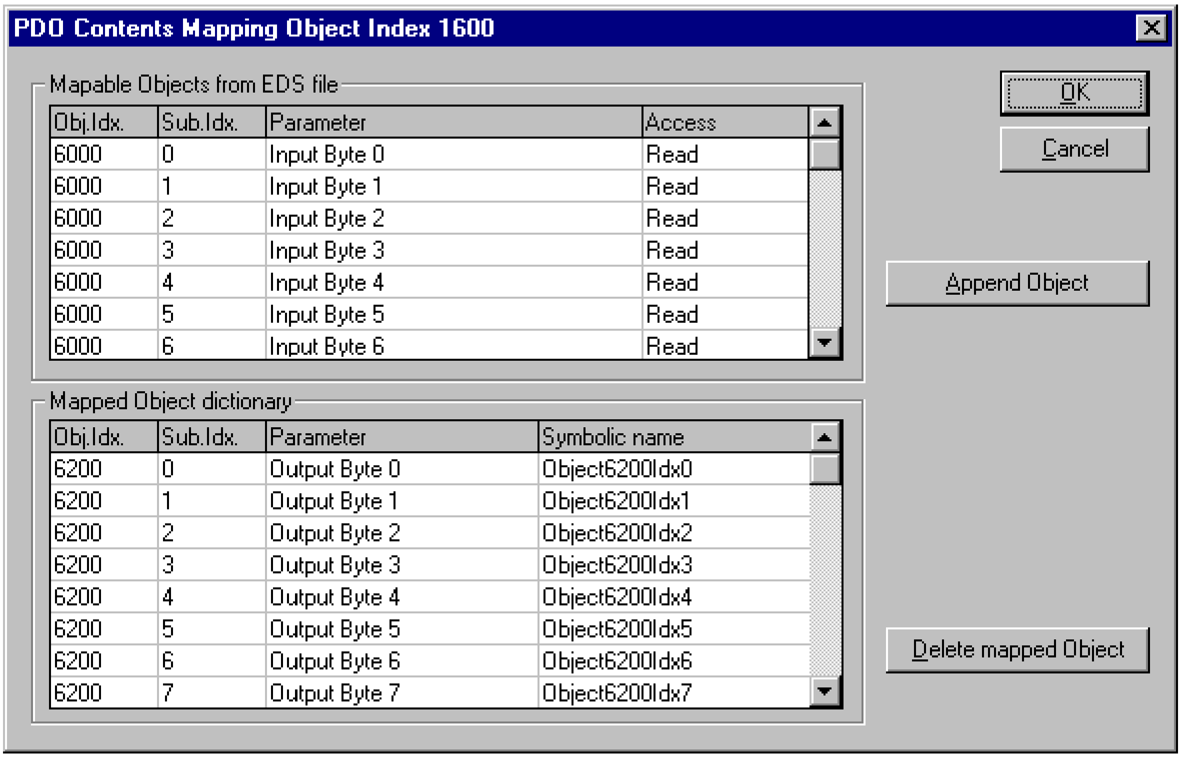
Figure 22: Definite a new Transmit PDO

SyCon suggests a free message number, which can be edited later in the PDO configuration window, if the PDO was taken over.

4.4.2.10 PDO Contents Mapping - Arrange a PDO

Some CANopen Nodes support the PDO data mapping and dynamic distribution. That means a user defined containment mapping of objects into a PDO. The mapping itself is always done by the Node internally after is has received new RX-PDO or has to send new TX-PDO, so that the master can handle the input and output PDOs coming from and going to the Node completely transparent. This guarantees high speed data transfer and execution in the view of the master. His job is it only to configure the Node's mapping dictionary during its configuration phase once.

A PDO can contain always up to 8 byte process data. The combination of these individual process data elements can be changed when the button **Append Object** is used. When a PDO was transferred from the upper table to the **Configured PDOs** table, SyCon maps automatically all found map able process data from the Node EDS file into this PDO.



Mapable Objects from EDS file

Obj.Idx.	Sub.Idx.	Parameter	Access
6000	0	Input Byte 0	Read
6000	1	Input Byte 1	Read
6000	2	Input Byte 2	Read
6000	3	Input Byte 3	Read
6000	4	Input Byte 4	Read
6000	5	Input Byte 5	Read
6000	6	Input Byte 6	Read

Mapped Object dictionary

Obj.Idx.	Sub.Idx.	Parameter	Symbolic name
6200	0	Output Byte 0	Object6200Idx0
6200	1	Output Byte 1	Object6200Idx1
6200	2	Output Byte 2	Object6200Idx2
6200	3	Output Byte 3	Object6200Idx3
6200	4	Output Byte 4	Object6200Idx4
6200	5	Output Byte 5	Object6200Idx5
6200	6	Output Byte 6	Object6200Idx6
6200	7	Output Byte 7	Object6200Idx7

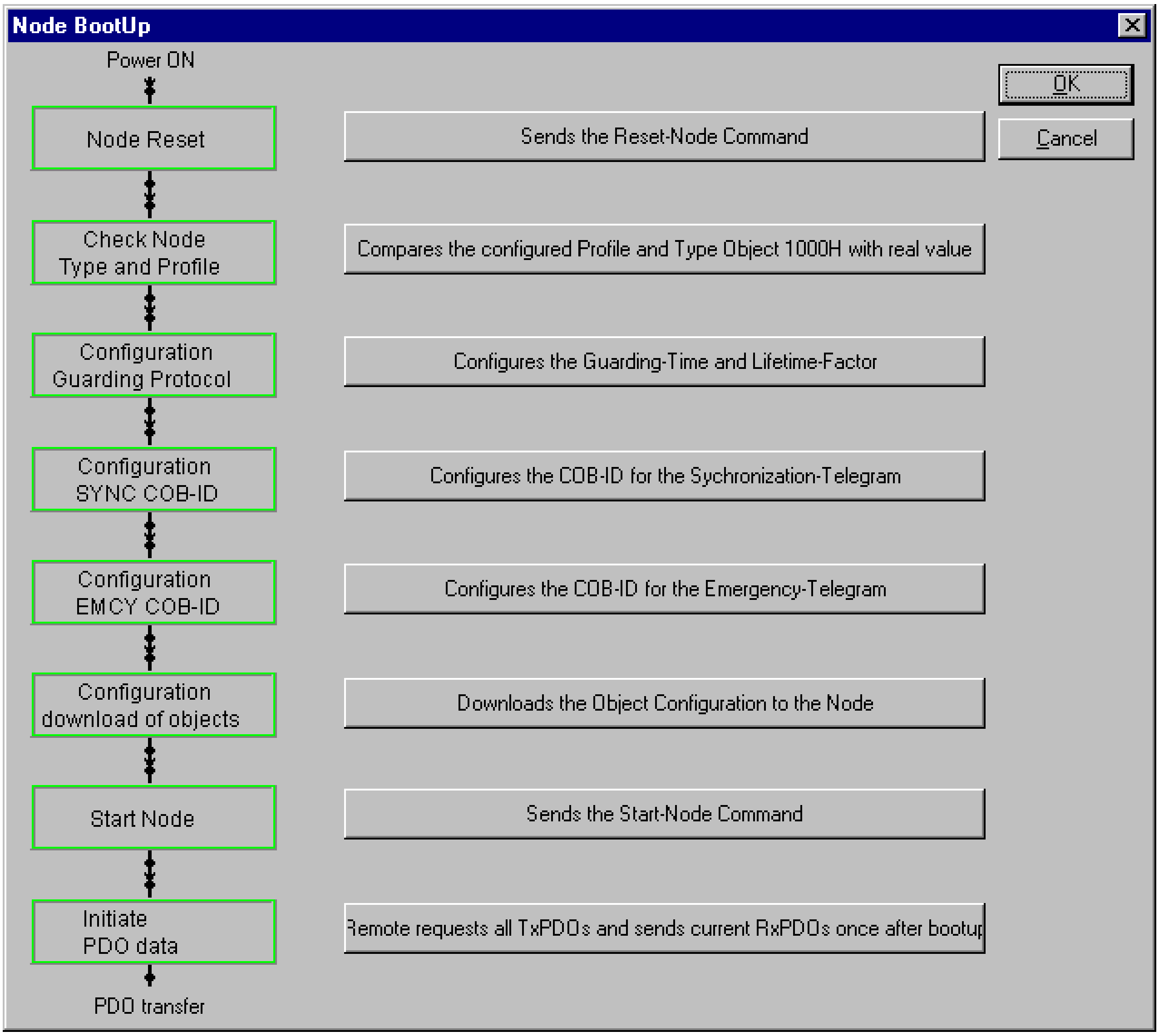
Figure 23: PDO Contents Mapping

The picture above is an example for a TX-PDO mapping. The upper table shows all available objects with their access right which are declared as supported in the node's EDS file. A double click onto one of these transfer it into the lower table. This table contains the real mapped objects that shall be a content of the PDO later in the process data exchange phase.

Note: Not all CANopen Nodes supports the PDO mapping feature!

4.4.2.11 Node BootUp

The Node BootUp defines the network startup behaviour of the Master for the particular Node to get it operative. There are different states a Master is running through per Node, till the BootUp sequence is finished for the Node. Each state now is configurable and can be enabled (activated) or disabled (deactivated) here. In the basic setting all states are activated.


Node BootUp

Power ON

Node Reset — Sends the Reset-Node Command

Check Node Type and Profile — Compares the configured Profile and Type Object 1000H with real value

Configuration Guarding Protocol — Configures the Guarding-Time and Lifetime-Factor

Configuration SYNC COB-ID — Configures the COB-ID for the Sychronization-Telegram

Configuration EMCY COB-ID — Configures the COB-ID for the Emergency-Telegram

Configuration download of objects — Downloads the Object Configuration to the Node

Start Node — Sends the Start-Node Command

Initiate PDO data — Remote requests all TxPDOs and sends current RxPDOs once after bootup

PDO transfer

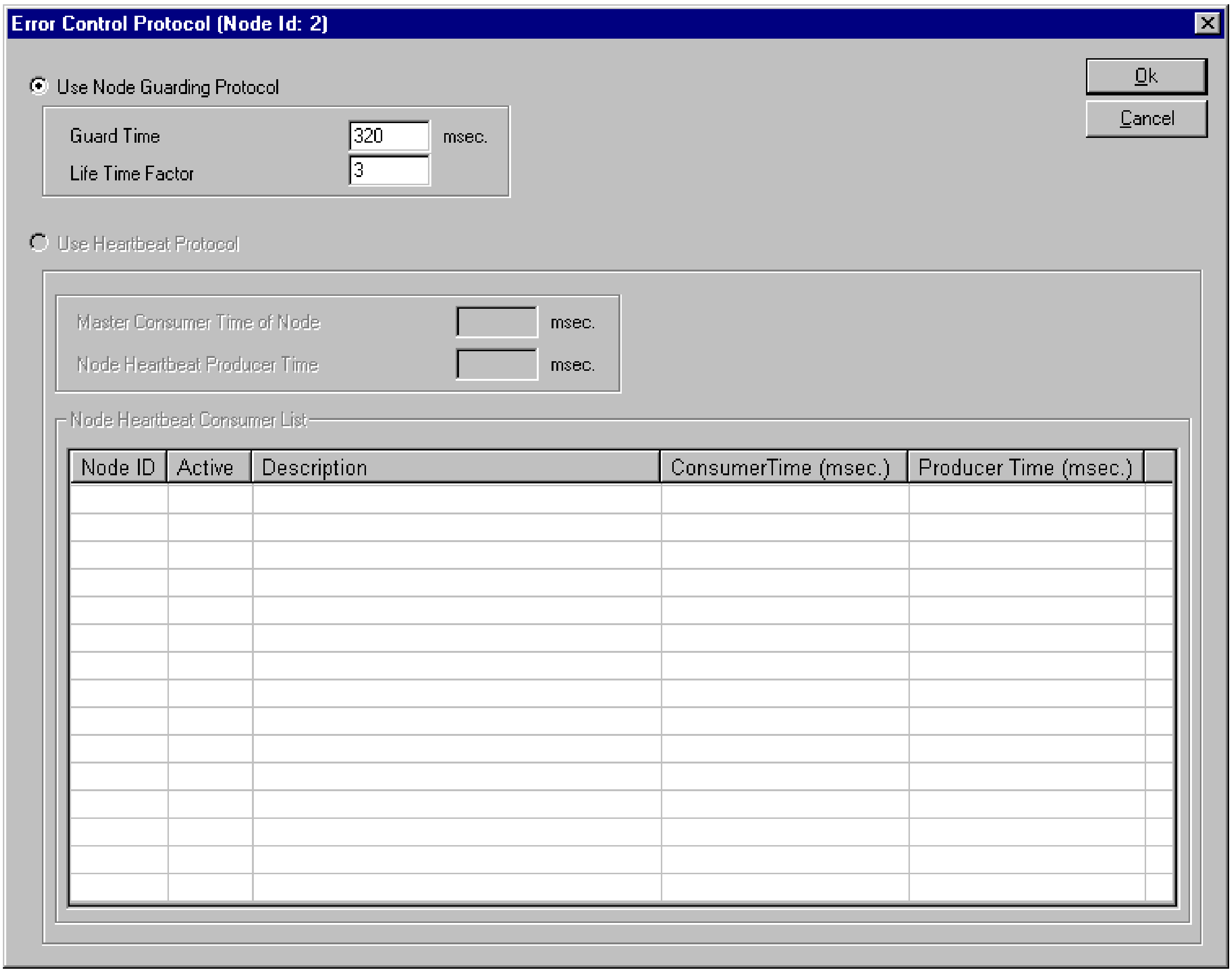
Figure 24: Online > Node Configuration > Node BootUp

Node BootUp parameter	**Description**
Node Reset	If enabled, the master sends as first the CANopen specific Node Reset Communication command.
Check Node Type and Profile	If enabled, the master will compare the contents of the mandatory Node Object 1000H is the device type with the values that are configured within SyCon. If the values are different, the master will report a parameterization error.
Configuration Guarding Protocol	A CANopen has two specific register responsible for the Node guarding protocol. If the item is enabled, the master will write the Guard Time and Life-Time factor of the Node configuration into the corresponding objects of the Node during startup.
Configuration SYNC COB-ID	If the item is enabled, the master will write the SYNC COB-ID of the configuration into the corresponding objects of the Node during startup.
Configuration EMCY COB-ID	If the item is enabled, the master will write the EMCY COB-ID of the configuration into the corresponding objects of the Node during startup.
Configuration download of objects	To get a PDO communication to a Node working, the master has to send all relevant configuration objects to the Node. For example the mapping table, the COB-ID a PDO shall be sent through are covered here. If enabled, all these parameter and also the user specific objects which are added manually in the Node object configuration window are written down to the Node by the master.
Start Node	To reach the operational state in CANopen a Node has to get the CANopen specific Start Node command. If enabled, the master will send the Start Node command to the Node at the end of the boot-up procedure.
Initiate PDO data	This item selects if the installed PDOs shall be automatically written and read by the master directly after the startup once. This ensures that the latest output data which can be found within the Masters output process data area is sent to the Node and that the latest Node input data is read from the Node and be placed into the input process data area.

Table 22: Node BootUp

4.4.2.12 Configuration Error Control Protocol

The Configuration Error Control Protocol defines the device monitoring. In doing so the Node Guarding Protocol or the Heartbeat Protocol can be used. In the following you find information about the functional principle, about the settings as far as notes to the Node Guarding and the Heartbeat Protocol.



Error Control Protocol (Node Id: 2)

● Use Node Guarding Protocol

Guard Time: 320 msec.

Life Time Factor: 3

○ Use Heartbeat Protocol

Master Consumer Time of Node: msec.

Node Heartbeat Producer Time: msec.

Node Heartbeat Consumer List

Node ID	Active	Description	ConsumerTime (msec.)	Producer Time (msec.)

Figure 25: Error Control Protocol - Node Guarding Protocol

- **Node Guarding Protocol**

  Functional Principle: The Master sends cyclically polling telegrams (Remote Request) to the Node (Node Guarding), to check, if the Node still exists on the bus. The Node sends its actual state as answer back to the Master. The Nodes can use the poll telegrams of the Master to supervise the Master on its part (Life Guarding).

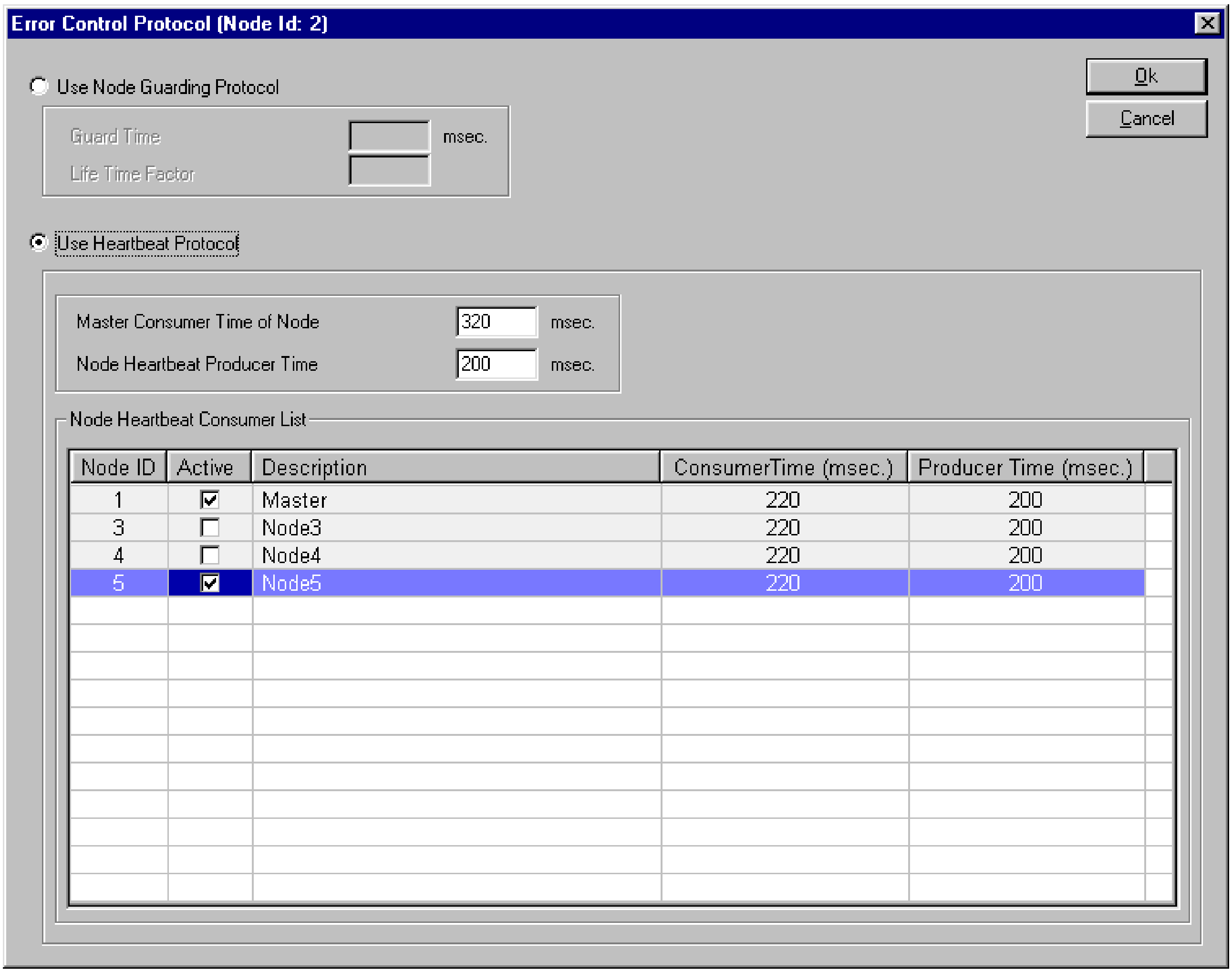
  Requirement: The Node has to support the Node Guarding Protocol.

Settings: The **Guard time** is the setting for the supervision of the Node in the view of the Master (Master controls Node). If the communication is running, the Master will poll the Node in the time interval **Guard time**, to check, if the Node is still present in the network or not. If this value is configured with 0, the supervision is disabled in the Master as well as in the Node.

The **Life time factor** is the setting for the supervision of the Master in the view of the Node (Node controls Master). If the communication is running, the Node will control the Master in the **Guard time** multiplicated with the **Life time factor** as time interval, to check, if the Master is still present in the network or not. If this value is configured with 0, the supervision in the Node is deactivated.

Warning: To reach a stable communication of the Node on the CANopen, the **Life Time Factor** has to be set to minimal 2.

Note: A Life Guarding can only be used, if the Master carries out a Node Guarding, that means Life Guarding presumes Node Guarding.



○ Use Node Guarding Protocol

Guard Time ______ msec.

Life Time Factor ______

● Use Heartbeat Protocol

Master Consumer Time of Node 320 msec.

Node Heartbeat Producer Time 200 msec.

Node Heartbeat Consumer List

Node ID	Active	Description	ConsumerTime (msec.)	Producer Time (msec.)
1	☑	Master	220	200
3	□	Node3	220	200
4	□	Node4	220	200
5	☑	Node5	220	200

Figure 26: Error Control Protocol - Heartbeat Protocol

- **Heartbeat Protocol**

  Functional Principle: A Heartbeat Producer transmits the Heartbeat telegram cyclically with the time interval defined in the field **Node Heartbeat Producer Time**. One or more Heartbeat Consumer may receive the indication. The relationship between producer and consumer is configurable via the Object Dictionary entries. The Heartbeat Consumer guards the reception of the Heartbeat telegram within the Heartbeat consumer time. If the Heartbeat telegram is not received within this time a Heartbeat Event will be generated.

  Requirements: In order that the Heartbeat Protocol can be used, the following requirements are necessary:

  1) The Heartbeat Function has to be activated in the bus parameters of the Hilscher Master.

  2) In the Hilscher Master device the firmware version V1.070 has to be loaded.

  3) The Node has to support the Heartbeat Protocol. The SyCon System Configurator reads out this information from the EDS file of the corresponding device.

  Settings: The **Master Consumer Time** is the setting for supervising the Node from view of the Master (Master controls Node). If the communication is running, the Node sends telegrams in the time interval **Producer Heartbeat Time**, which are received and controlled by the Master. With this the value of the **Master Consumer Time** must have at least the value of the **Producer Heartbeat Time** or must be higher. If the value of the **Master Consumer Time** is 0, the Master does not control this Node.

  Devices, which have to be supervised by this Node, can be selected in the **Consumer Heartbeat list**.

  If the Node should control the Master, the **Consumer Heartbeat Node List** has to be activated by clicking on it in the line to the Master (default setting). The **Consumer Time** can be changed in this line by overwriting in this line (double click). The **Producer Time** is displayed informatively. Thereby the value of the **Consumer Time** must have the value of the **Producer Time** or must be higher. If the value of the **Consumer Time** is 0, the Node does not control the Master.

  If this Node should control another Node, the **Consumer Heartbeat Node List** has to be activated in the line to the Node by enabling the check box. The **Consumer Time** can be changed by overwriting in this line (double click). The **Producer Time** is displayed informatively. Thereby the value of the **Consumer Time** must have the value of the **Producer Time** or must be higher.

  The picture for example shows, that Node 2 controls Node 5 with 220ms, as well as Node 1 (the Master).

Note: The number of devices which can be controlled by this Node, depends on the scope of performance of this device.

- **Further Notes to the Node Guarding Protocol and Heartbeat Protocol**

Note: Heartbeat can displace Node Guarding Protocol (and the Life Guarding).

Note: SyCon reads out the EDS file with regard to the support of the Nodeguarding as well as the Heartbeat Protocol and gives out the following in the Error Control Protocol window: If only Nodeguarding is supported, Nodeguarding is preset. If only Heartbeat is supported, Heartbeat is preset. If Nodeguarding and Heartbeat is supported, Nodeguarding is preset. If there is no declaration about Nodeguarding or about Heartbeat, Nodeguarding is preset. In this case it must be checked, if the Node supports the Nodeguarding effectively of not.

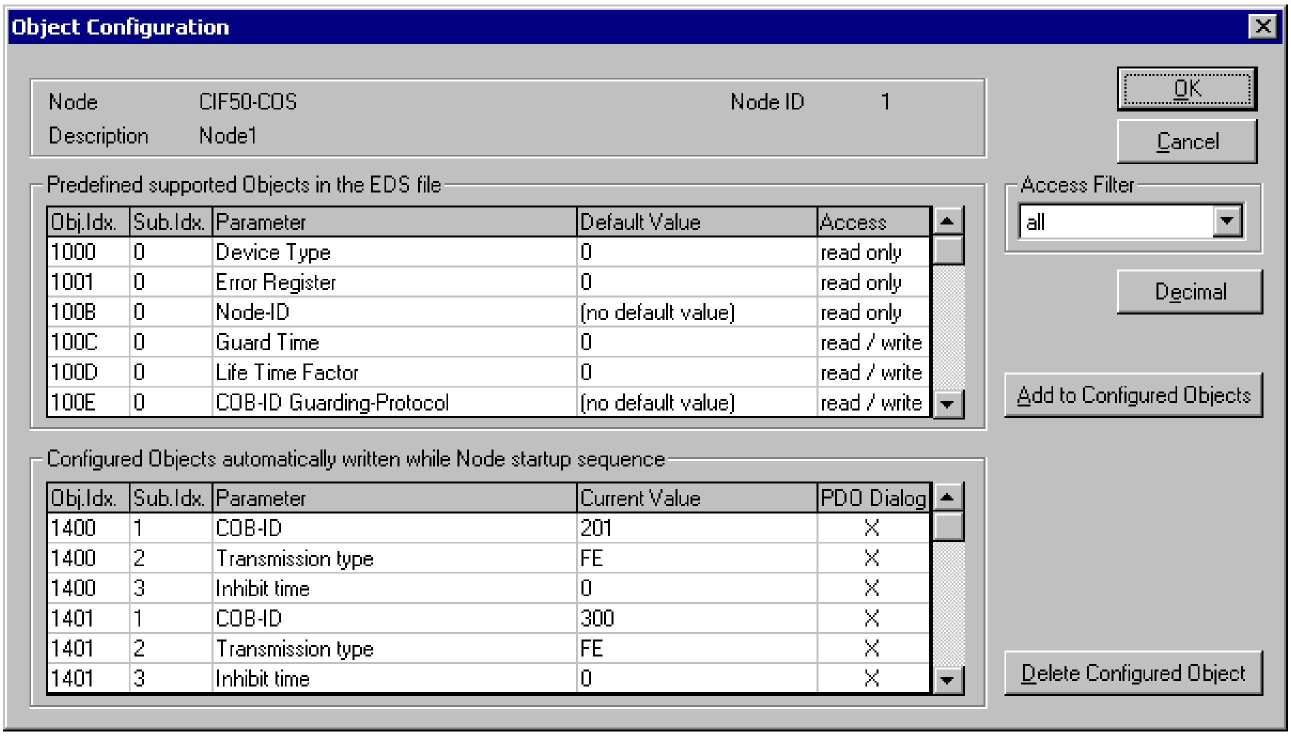
Note: If the Node does not support Node Guarding Protocol or the Heartbeat Protocol, the Master cannot detect, if the Node has failed! If the Node does not support the Node Guarding or the Heartbeat, **the Guard Time** and the **Life Time Factor** have to be set to 0.

4.4.2.13 Object Configuration

The most important part of a CANopen device is its object directory. The object directory is essentially a grouping of objects accessible via the network in an ordered predefined fashion. Each object within the directory is addressed using a 16-bit index. The Device profiles of CANopen define the name, the meaning, the value range and data type of the Service Data Objects (SDO). With so-called Service Data Messages the contents of an object and its sub index can be changed. This is necessary to set up the behaviour of a CANopen Node in the right manner. This is necessary to change the behaviour of the CANopen Node.

To get access to the SDO configuration press the **Object Configuration** button. The following window below will appear and SyCon shows in the upper table all supported objects read out from the EDS file of the Node. If you have already inserted some PDOs you will find existing entries in the lower table too.

SyCon places some objects in this table automatically when a PDO in the **Node Configuration** window is inserted, to set up the several variables of the PDO objects right, so that the wished configuration corresponds to the Node behaviour later in the process data communication. These values can not be edited.



Node CIF50-COS Node ID 1

Description Node1

Access Filter: all

Predefined supported Objects in the EDS file

Obj.Idx.	Sub.Idx.	Parameter	Default Value	Access
1000	0	Device Type	0	read only
1001	0	Error Register	0	read only
100B	0	Node-ID	(no default value)	read only
100C	0	Guard Time	0	read / write
100D	0	Life Time Factor	0	read / write
100E	0	COB-ID Guarding-Protocol	(no default value)	read / write

Configured Objects automatically written while Node startup sequence

Obj.Idx.	Sub.Idx.	Parameter	Current Value	PDO Dialog
1400	1	COB-ID	201	X
1400	2	Transmission type	FE	X
1400	3	Inhibit time	0	X
1401	1	COB-ID	300	X
1401	2	Transmission type	FE	X
1401	3	Inhibit time	0	X

Figure 27: Online > Node Configuration > Object Configuration

4.4.2.14 OPC Objects

With the information in this window symbols for the OPC communication are entered over SDO. For this the objects are selected from the indicated list. Further information you find in the manual for the OPC Server.

4.4.2.15 OPC User Defined Objects

With the information in this window symbols for the OPC communication are entered over SDO. For this the objects can be entered by the user. Further information you find in the manual for the OPC Server.

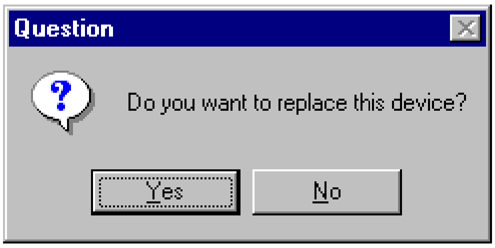
4.4.3 Replace Node

If a Node already exists in the configuration and should be replaced against the other Node, you first have to set the focus on the Node (left mouse click) and then choose the menu **Edit > Replace**.

or

make a right mouse click at the Node and select **Replace** in the now opened menu.

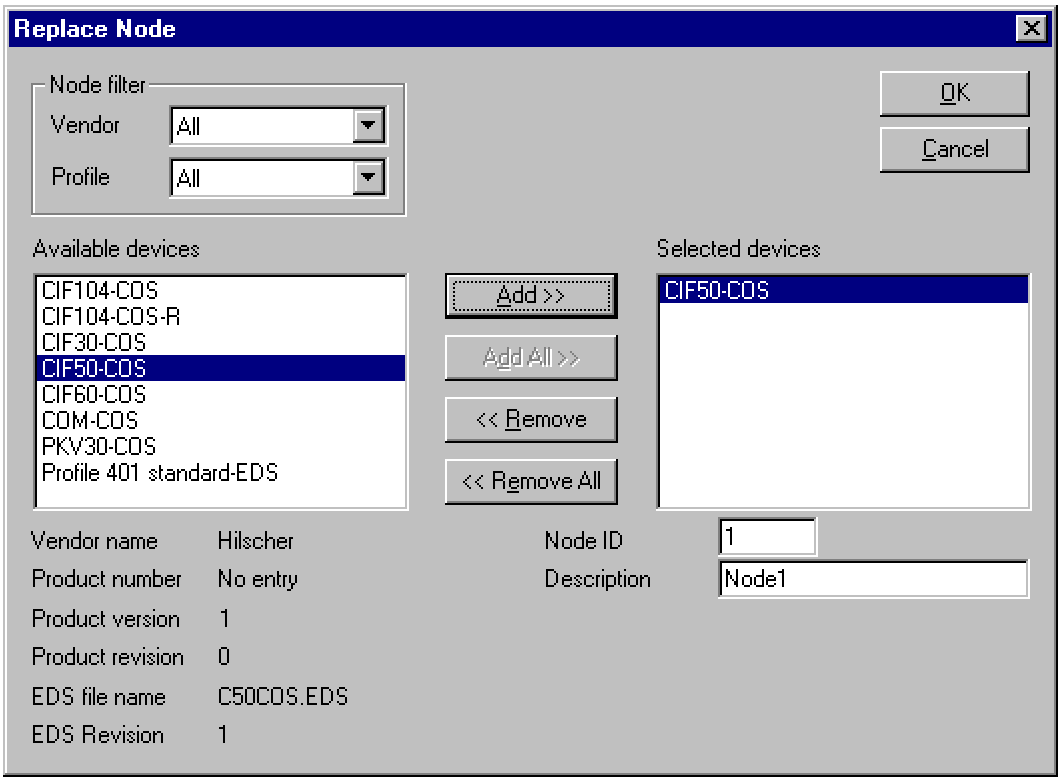
In the opened window the question appears if the Node should be replaced.



Do you want to replace this device?

Figure 28: Security question Replace Device

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



Node filter
Vendor: All
Profile: All

Available devices:
CIF104-COS
CIF104-COS-R
CIF30-COS
CIF50-COS
CIF60-COS
COM-COS
PKV30-COS
Profile 401 standard-EDS

Selected devices:
CIF50-COS

Vendor name	Hilscher
Product number	No entry
Product version	1
Product revision	0
EDS file name	C50COS.EDS
EDS Revision	1

Node ID: 1
Description: Node1

Figure 29: Edit > Replace Node

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

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.

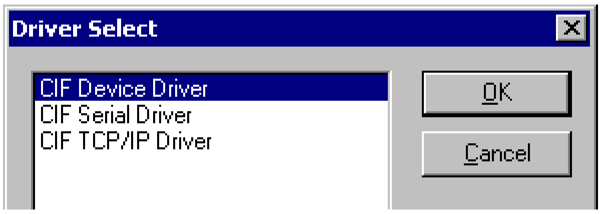


Figure 30: Driver Selection

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	*CIF Device Driver*	*69*
CIF Serial Driver	*CIF Serial Driver*	*71*
CIF TCP/IP Driver	*CIF TCP/IP Driver*	*73*

Table 23: Driver Selection

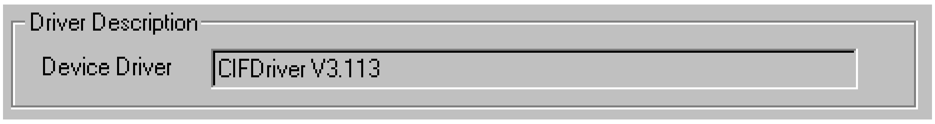
To select a driver, mark the favored driver by clicking on it in the dialog window **Driver Select** and confirm your selection with **OK**.

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



Driver Description

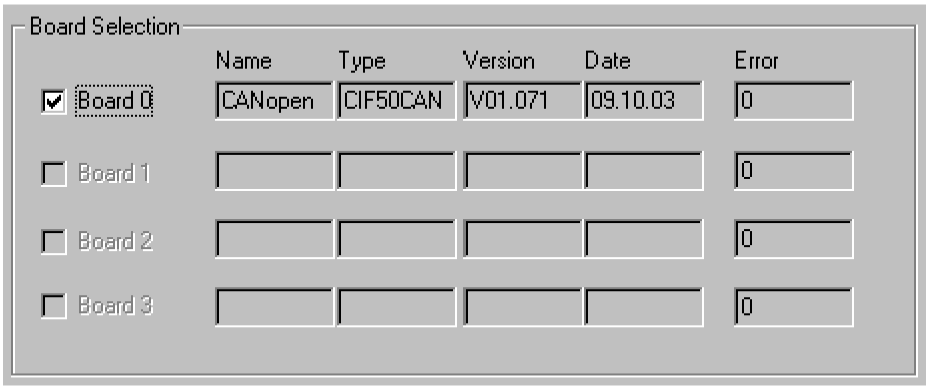
Device Driver: CIFDriver V3.113

Figure 31: 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



Board Selection

	Name	Type	Version	Date	Error
☑ Board 0	CANopen	CIF50CAN	V01.071	09.10.03	0
□ Board 1					0
□ Board 2					0
□ Board 3					0

Figure 32: CIF Device Driver - Board Selection

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

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

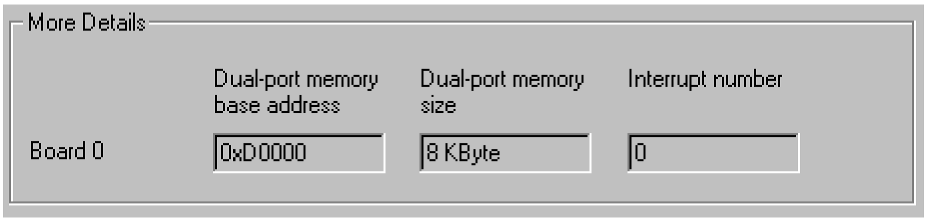
Table 24: Device Assignment - Checkboxes of the CIF Device Driver

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.



More Details

	Dual-port memory base address	Dual-port memory size	Interrupt number
Board 0	0xD0000	8 KByte	0

Figure 33: CIF Device Driver - More Details

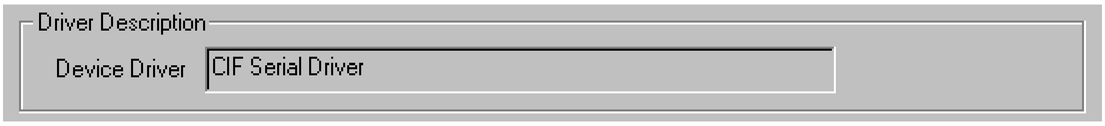
In this dialog the used **Dual-port memory base address**, the **Dual-port memory size** and the **Interrupt number** of the selected board are displayed. Interrupt number 0 means polling mode.

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



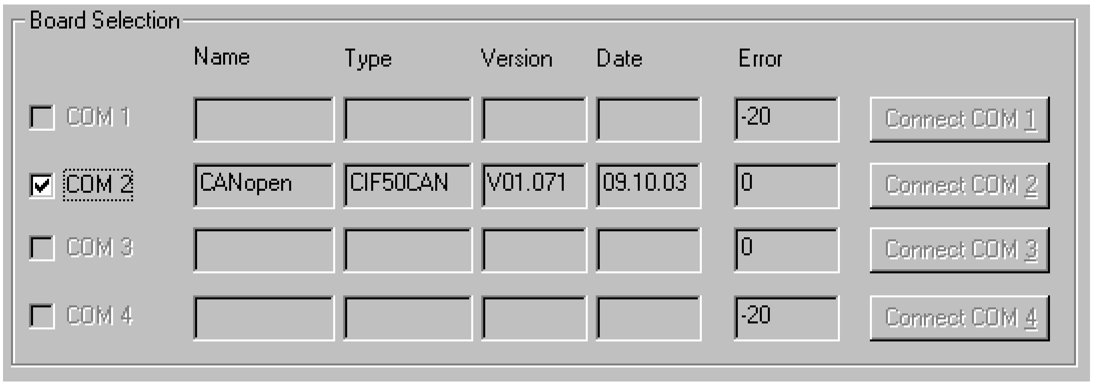
Device Driver: CIF Serial Driver

Figure 34: 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



	Name	Type	Version	Date	Error
☐ COM 1					-20
☑ COM 2	CANopen	CIF50CAN	V01.071	09.10.03	0
☐ COM 3					0
☐ COM 4					-20

Figure 35: CIF Serial Driver - Board Selection

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

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

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

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

Table 25: 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:

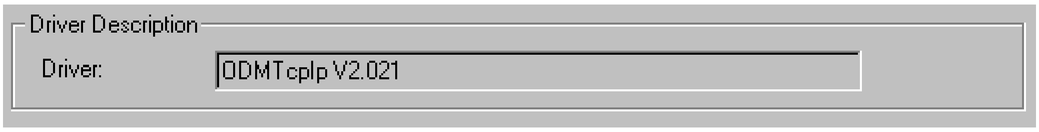
1. The Hilscher device is installed in a PC and the TCP/IP connection is built up to the PC, that means the IP address of the PC is used as IP address.

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



Driver: ODMTcpIp V2.021

Figure 36: 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.

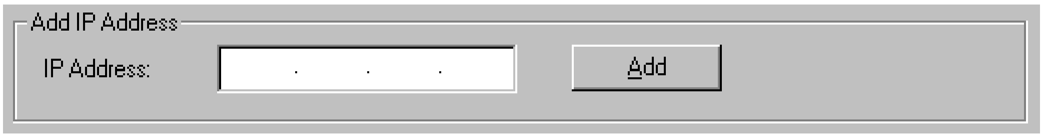


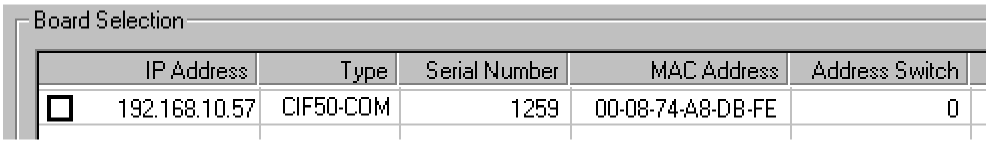
Figure 37: CIF TCP/IP Driver - Type in IP Address manually

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.



Board Selection

	IP Address	Type	Serial Number	MAC Address	Address Switch
□	192.168.10.57	CIF50-COM	1259	00-08-74-A8-DB-FE	0

Figure 38: 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 *76*.

The column **Address Switch** has no meaning here.

Connect Device

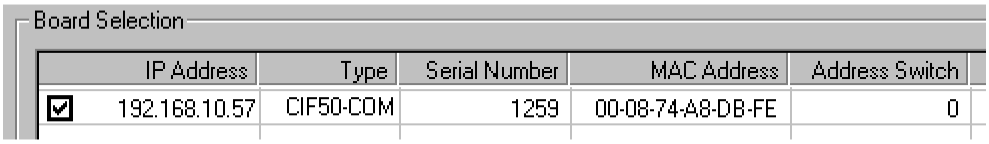
To connect a device to the PC, the checkbox of the favored device has to be selected in front of the **IP Address** field.

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

Table 26: 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:



Board Selection

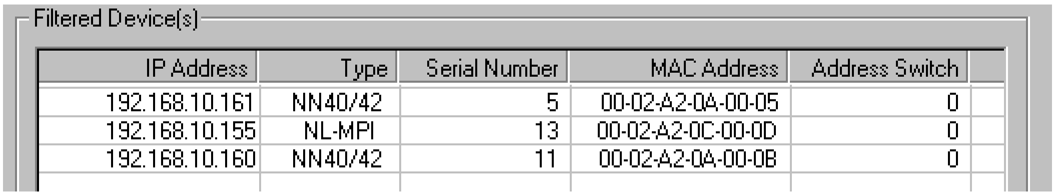
	IP Address	Type	Serial Number	MAC Address	Address Switch
☑	192.168.10.57	CIF50-COM	1259	00-08-74-A8-DB-FE	0

Figure 39: 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 40: CIF TCP/IP Driver - Filtered Devices

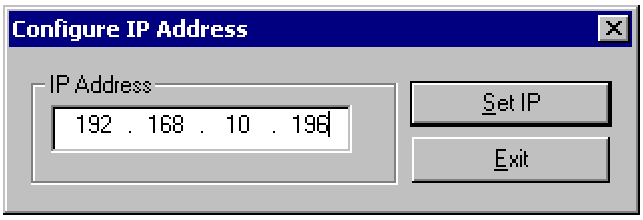
Devices listed in the table **Filtered Device(s)** were found during the network scan in the local Ethernet network, but they can not be assigned, because they belong to another device family.

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:



Configure IP Address

IP Address: 192 . 168 . 10 . 196

Figure 41: Set IP Address

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

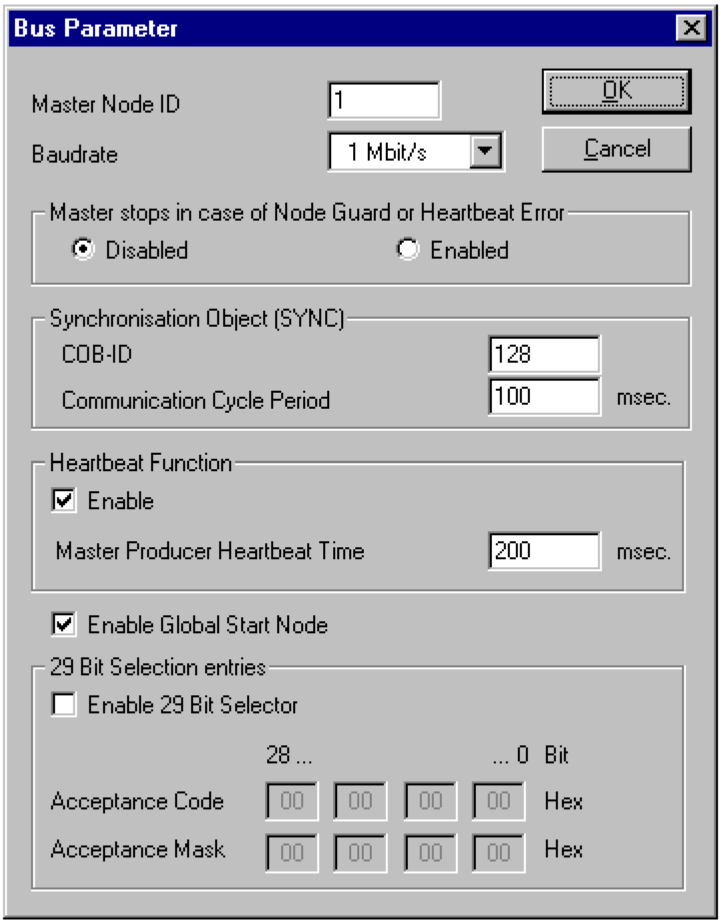
Note: The IP address set by clicking the **Set IP button** is only temporarily adjusted. A permanent storage of the IP address takes place with a download of the configuration from the framework.

5.2 Bus Parameter

In this window the basic settings for the CANopen network are done. Mainly, this concerns the setting of the **Baudrate**.

Attention: Check that all CANopen Nodes support also the selected Baud rate.

Basic rule: The Baud rate must be set same for all devices. The Node address on the other hand must differ from Node to Node.



Bus Parameter

Master Node ID: 1

Baudrate: 1 Mbit/s

Master stops in case of Node Guard or Heartbeat Error
● Disabled ○ Enabled

Synchronisation Object (SYNC)
COB-ID: 128
Communication Cycle Period: 100 msec.

Heartbeat Function
☑ Enable
Master Producer Heartbeat Time: 200 msec.

☑ Enable Global Start Node

29 Bit Selection entries
□ Enable 29 Bit Selector

	28 ...			... 0	Bit
Acceptance Code	00	00	00	00	Hex
Acceptance Mask	00	00	00	00	Hex

Figure 42: Settings > Bus Parameter

- **Master Node ID**

  The **Master Node ID** is necessary for the addressing of the device at the bus and has to be unique. Therefore it is not allowed to use this number two times in the same network.

- **Baudrate**

  Set the **Baudrate**. The following Baudrates are available:

Baudrate
10 kBit/s
20 kBit/s
50 kBit/s
100 kBit/s
125 kBit/s
250 kBit/s
500 kBit/s
800 kBit/s
1 Mbit/s

Table 27: Baudrate

- **Master stops in case of Node Guard or Heartbeat Error**

  The **Master stops in case of Node Guard or Heartbeat Error** feature defines the behaviour of the Master if the communication is interrupted to at least one Node. If the flag is set to **Activated**, the Master will also stop the communication to all further Nodes which were still responding and active. If the flag set to **Deactivated**, then a lost communication to one Node has no influence on the communication of the still present Nodes. For all the error affected Nodes the master remains in the state to try the reestablishment of the communication again.

- **Heartbeat Function**

  If the Heartbeat Protocol should be used for the device supervision, this has to be activated. In the **Master Producer Heartbeat Time** field the cycle time for the Master is entered in ms.

- **Synchronization Object (SYNC)**

  Furthermore the **Communication Cycle Period** and the message number **COB-ID** has to be set. The default value for the COB-ID is 128.

- **Enable Global Start Node**

  After the Master started all Nodes configured individually first, it sends a Global Start Node with activated menu option afterwards, in order to synchronize all Nodes again.

Attention: The following stetting are just for Master devices which support the 29 Bit CAN specific identifier according to CAN 2.0B. At the moment this is only available in case of the TSX CPP 100 module hardware.

- **Enable 29 Bit Selector**

  If this menu option is activated the 29 Bit identifier is switched free for the Master.

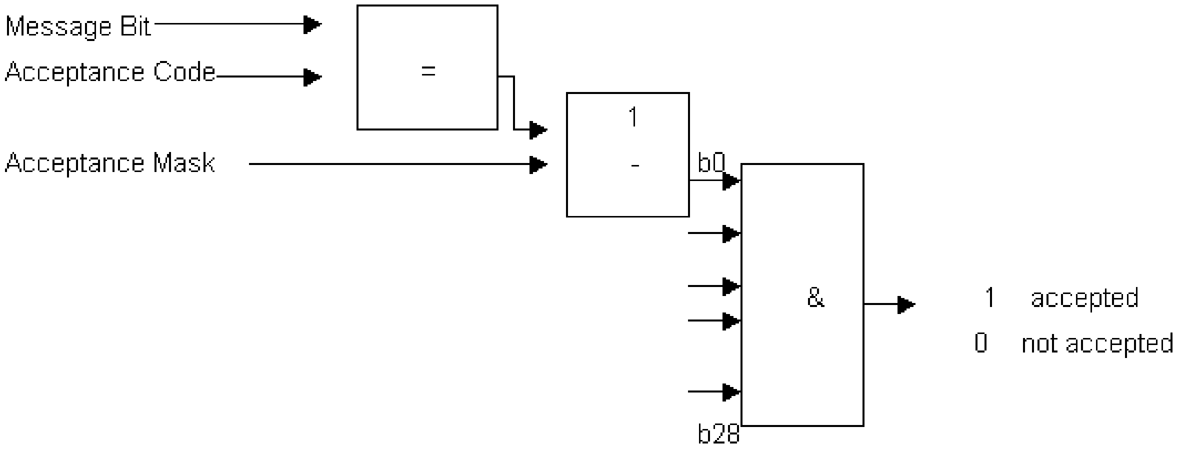


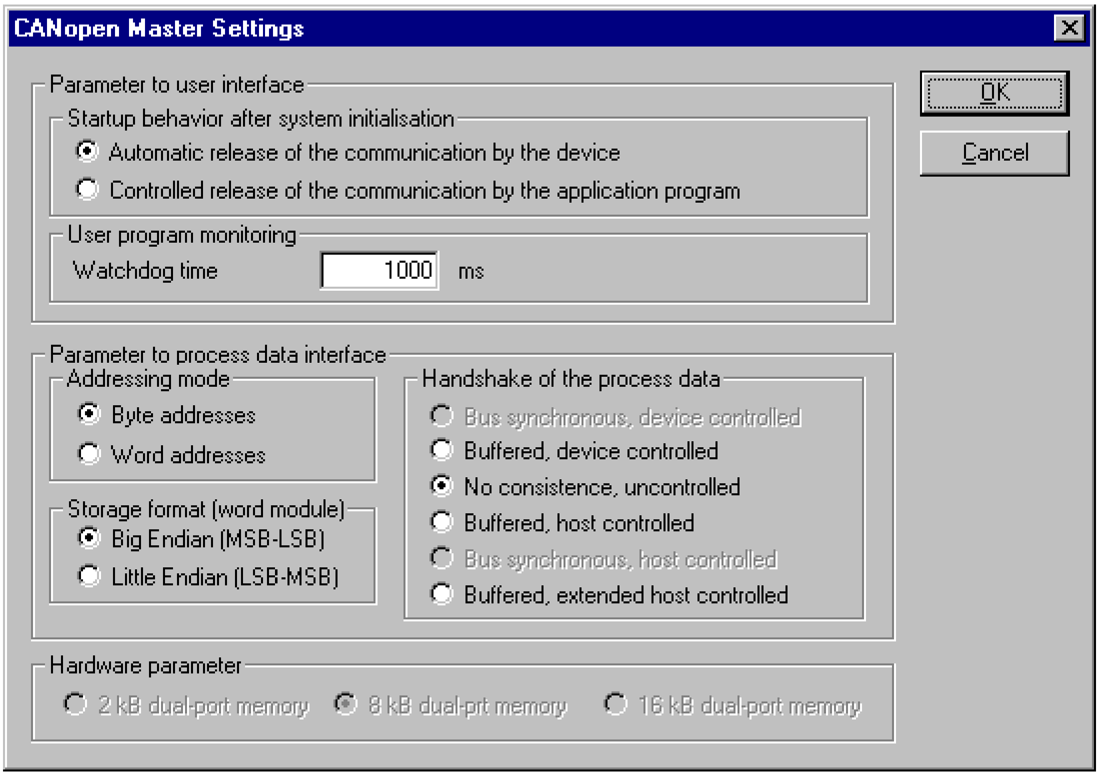
Figure 43: Diagram Acceptance Code / Acceptance Mask

5.3 CANopen Master

5.3.1 Master Settings

To enter the CANopen Master settings, select the menu **Settings > Master Settings** or click with the right mouse button on the corresponding Master symbol and select **Master Settings** from the list which opens up. The Master Settings are also available from the **Master Configuration** window.

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



CANopen Master Settings

Parameter to user interface

Startup behavior after system initialisation

● Automatic release of the communication by the device

○ Controlled release of the communication by the application program

User program monitoring

Watchdog time 1000 ms

Parameter to process data interface

Addressing mode

● Byte addresses

○ Word addresses

Storage format (word module)

● Big Endian (MSB-LSB)

○ Little Endian (LSB-MSB)

Handshake of the process data

○ Bus synchronous, device controlled

○ Buffered, device controlled

● No consistence, uncontrolled

○ Buffered, host controlled

○ Bus synchronous, host controlled

○ Buffered, extended host controlled

Hardware parameter

○ 2 kB dual-port memory ● 8 kB dual-prt memory ○ 16 kB dual-port memory

Figure 44: Settings > CANopen Master Settings

- **Startup behaviour after system initialization**

  When **Automatic release of the communication by the device** is selected, the Master device starts with the data exchange at the Bus after the initializing has been finished. If **Controlled release of communication by the application program** has been set, the application program must activate the data exchange at the Bus.

- **User program monitoring**

  The **Watchdog time** determines how long the device waits for a triggering of the software watchdog by the application program until it sets the outputs of the Slave devices to 0. This behaviour must be activated by the user program and does not start automatically.

Note: The Watchdog is not a special CANopen function, but an often needed feature in interaction with a SoftPLC.

- **Addressing mode**

  The addressing mode of the process data image determines how the addresses (Offsets) of the process data are interpreted. Either of the addressing modes **Byte addresses** or **Word addresses** are possible. See also details on the next page.

- **Storage format (word module)**

  The storage format determines how the data are interpreted and laid down in the process image. For the Word data type it is possible to select higher/lower value and for Byte data type lower/higher value Byte.

- **Handshake of the process data**

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

  The set handshake of the process data must be supported by the application program. The handshake buffered, host controlled is mostly supported. The handshake no consistence, uncontrolled works without handshake and both processes run free.

  A detailed description is provided in the manual for the Toolkit or the manual for the device driver.

  If you run a soft PLC or a visualization software on your PC, please check in their documentation, which handshake mode this program expects.

- **Hardware parameter**

  With this parameter you 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 8K, whereby 7K are process data.

Note: For CIF 30-COM, CIF 104-COM and COM-COM 8 KByte are prescribed.

5.3.2 Addressing Mode

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

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

Table 28: 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 image:

IEC addresses in Byte mode	IEC addresses in word mode	Offset addresses 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	QB 1	2	1110 0010	Output of QB2 / QB1 to a single Byte module: D7 D6 D5 D4 D3 D2 D1 D0 1 1 1 0 0 0 1 0
QB 3		3	0000 0000	
QB 4 QB 5	QB 2	4 5	1111 1000 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): D7 D6 D5 D4 D3 D2 D1 D0 D7 D6 D5 D4 D3 D2 D1 D0 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1
QW 6	QW 3	6 7	1111 1111 0100 0100	Output of QW6 / QW3 in the data format lower/higher value Byte: D15 D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0 0 1 0 0 0 1 0 0 1 1 1 1 1 1 1 1 Output of QW6 / QW3 in the data format higher/lower value Byte: D15 D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0 1 1 1 1 1 1 1 1 0 1 0 0 0 1 0 0

Table 29: Example for place to keep data in the process image

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

Byte addressing		
Byte 0	IB 0	IW 0
Byte 1	IB 1	
Byte 2	IB 2	IW 2
Byte 3	IB 3	
Byte 4	IB 4	IW 4
Byte 5	IB 5	

Word addressing		
Word 0	IB 0	IW 0
	-	
Word 1	IB 1	IW 1
	-	
Word 2	IB 2	IW 2
	-	

Table 30: Image of the method of addressing for input

Byte addressing		
Byte 0	QB 0	QW 0
Byte 1	QB 1	
Byte 2	QB 2	QW 2
Byte 3	QB 3	
Byte 4	QB 4	QW 4
Byte 5	QB 5	

Word addressing		
Word 0	QB 0	QW 0
	-	
Word 1	QB 1	QW 1
	-	
Word 2	QB 2	QW 2
	-	

Table 31: Image of the method of addressing for output

5.3.3 Master Configuration

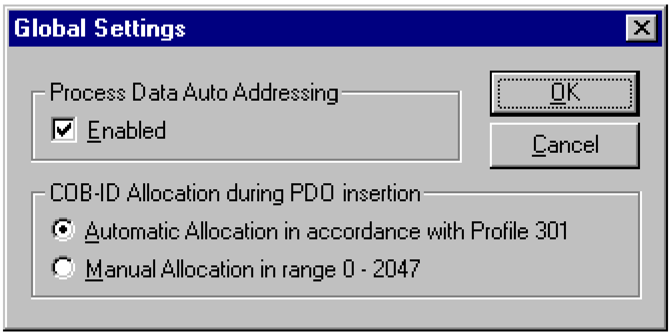
The Master configuration is described further above in section *Master Configuration* on page *38*.

5.3.4 Global Settings

First you have to set the focus on the Master (left mouse click) and select the menu **Settings > Global Settings**

or

click with the right mouse key on the symbol of the Master device to select the menu **Master Configuration** and then click the button **Global Settings**.



Process Data Auto Addressing

☑ Enabled

COB-ID Allocation during PDO insertion

● Automatic Allocation in accordance with Profile 301

○ Manual Allocation in range 0 - 2047

Figure 45: Settings > Global Settings

- **Process Data Auto Addressing**

  In this window it is adjusted whether the process data addressing is executed automatically by SyCon (active selected) or manually by the user (active not selected).

Auto Addressing active	**Auto Addressing deactivated**
Auto addressing (by SyCon)	Manually addressing (by the user)
The addresses will be allocated beginning with 0 and incremented in accordance with the entry sequence of the Slaves before downloading and can be viewed and checked in the menu **View > Address Table**.	The address 0 is shown in the **I Addr or O Addr** and must be overwritten by the user.

Table 32: Process Data Auto Addressing activated / deactivated

- **COB-ID Allocation during PDO insertion**

  The CANopen specification provides that the message number (COB-ID) of a PDO is given relatively to the Node address according to a fixed routine. It is called Pre-Defined Connection Set. This is described in section *COB-ID (Predefined Connection Set)* on page *176*.

- **Automatic Allocation in accordance with Profile 301**

  If this menu option is selected, an alteration of the message number of the PDOs is not possible and its assignment takes place automatically by the CANopen profile 301.

- **Manual Allocation in range 0-2047**

  If this menu option is selected, a definition of the message number of the PDOs is possible and can take place in the context of the possible 2048 (11 bit) different CAN-Identifier manually.

5.4 CANopen Node

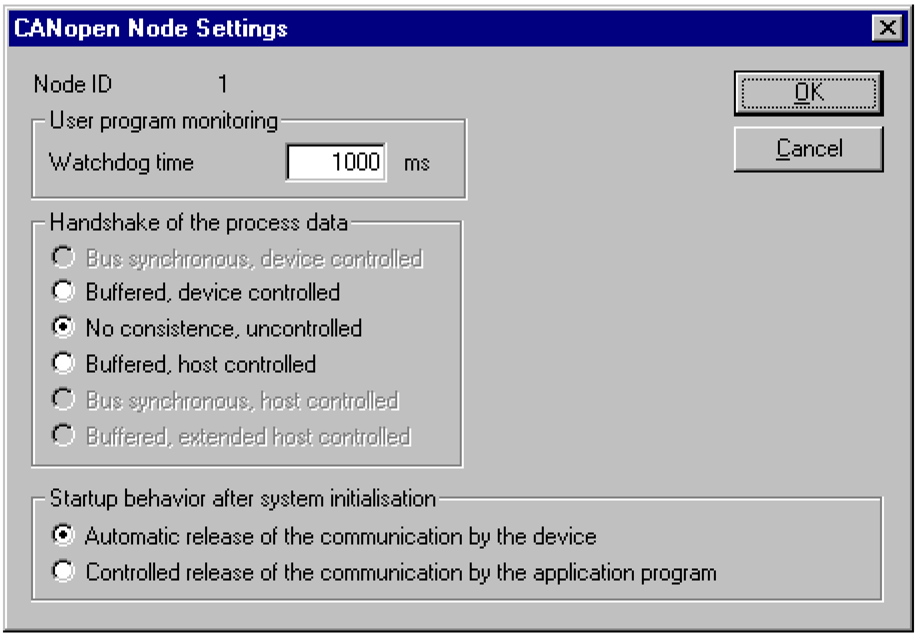
5.4.1 Node Settings

The CANopen Node Settings contain parameters that define the behaviour of the device at its user interface, which does not belong to the CANopen Node configuration. This menu point is applicable only to Hilscher devices. These settings are transferred with the download of the CANopen configuration to the device.

In order to open the CANopen Node Settings menu, first choose the Node and then open the window in the **Settings > Node Settings** menu.

or

make a right mouse click at the symbol of the Hilscher Node device and then select **Node Settings**.



CANopen Node Settings

Node ID 1

User program monitoring

Watchdog time 1000 ms

Handshake of the process data

○ Bus synchronous, device controlled
○ Buffered, device controlled
● No consistence, uncontrolled
○ Buffered, host controlled
○ Bus synchronous, host controlled
○ Buffered, extended host controlled

Startup behavior after system initialisation

● Automatic release of the communication by the device
○ Controlled release of the communication by the application program

Figure 46: Settings > CANopen Node Settings

- **User program monitoring**

  The watchdog time appoints how long the device will wait for a user trigger of the watchdog, until it resets all outputs to zero. This must be activated from the application.

- **Handshake of the process data**

  With this 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.

  The set handshake of the process data must be supported by the application program. The handshake buffered, host controlled is mostly supported. The handshake no consistence, uncontrolled works without handshake and both processes run free.

  A detailed description is provided in the manual for the Toolkit or the manual for the device driver.

  If you run a soft PLC or a visualization software on your PC, please check in their documentation, which handshake mode this program expects.

- **Startup behaviour after system initialization**

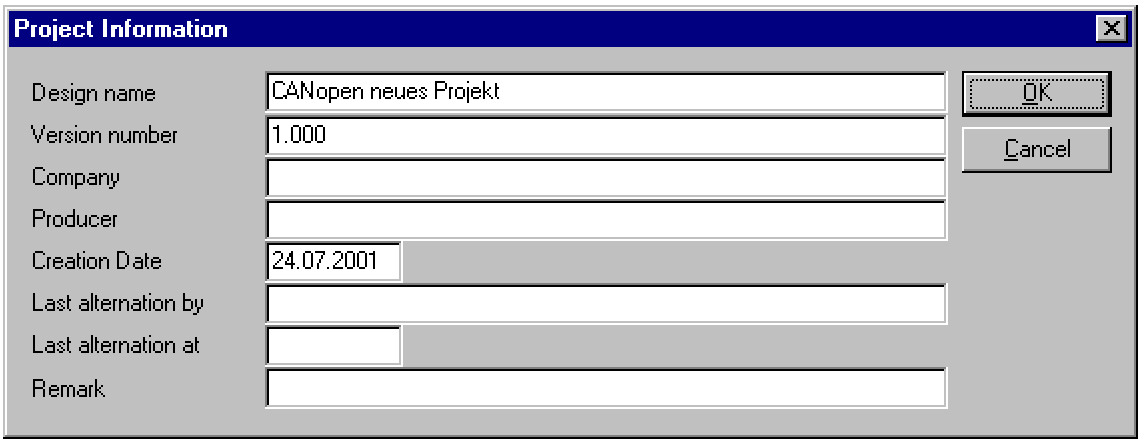
  If **Automatic release of the communication by the device** is selected, the Slave is ready to communicate with the master when started. If **Controlled release of the communication by the application program** is selected, the user has to release communication by a defined release procedure.

5.4.2 CANopen Node Configuration

The CANopen Node Configuration is described further above in section *Node Configuration* on page *42*.

5.5 Project Information

If the user creates his 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 Information

Design name: CANopen neues Projekt

Version number: 1.000

Company:

Producer:

Creation Date: 24.07.2001

Last alternation by:

Last alternation at:

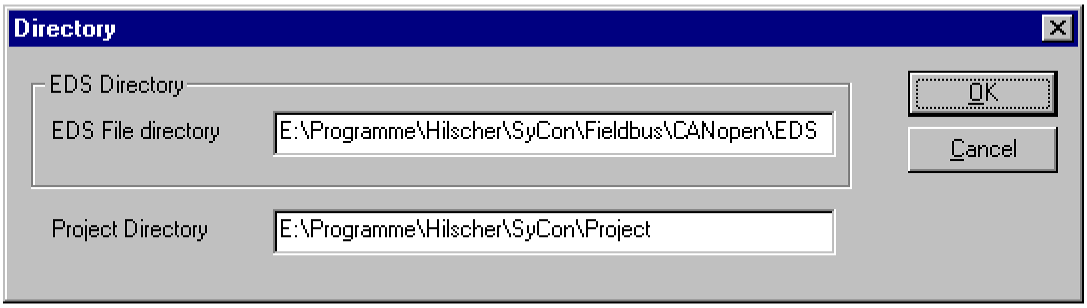
Remark:

Figure 47: Settings > Project Information

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

5.6 Path

When the **Settings > Path** menu is selected, then the search path for EDS files is displayed.



Directory

EDS Directory

EDS File directory: E:\Programme\Hilscher\SyCon\Fieldbus\CANopen\EDS

Project Directory: E:\Programme\Hilscher\SyCon\Project

Figure 48: Settings > Path

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

5.7 Language

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

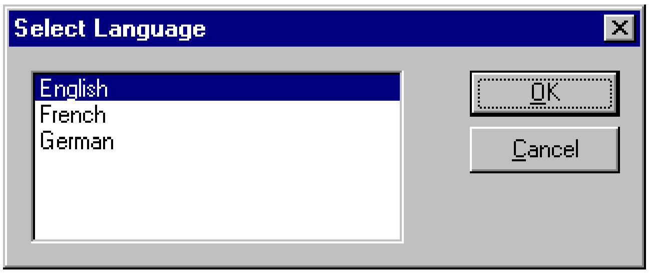


Figure 49: Settings > Language

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.

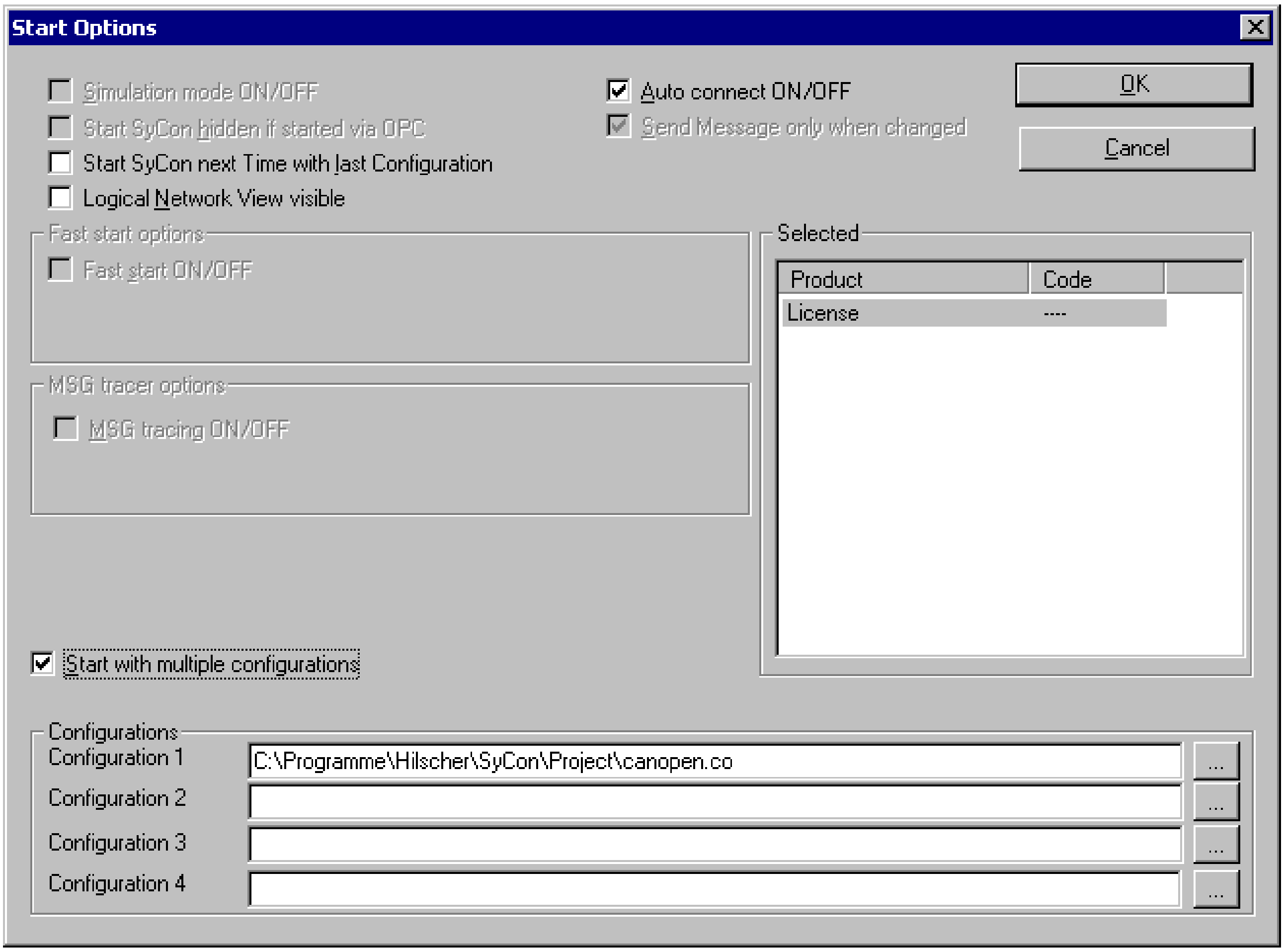
After restarting the System Configurator, the language will have changed to the one selected.

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

5.8 Start Options

After activating the **Settings > Start...** menu point in the network mode, 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.

Note: The point of menu start options appears only in the selection settings, if the network view is opened.



Start Options

□ Simulation mode ON/OFF

□ Start SyCon hidden if started via OPC

□ Start SyCon next Time with last Configuration

□ Logical Network View visible

☑ Auto connect ON/OFF

☑ Send Message only when changed

Fast start options

□ Fast start ON/OFF

MSG tracer options

□ MSG tracing ON/OFF

Selected

Product	Code
License	----

☑ Start with multiple configurations

Configurations

Configuration 1: C:\Programme\Hilscher\SyCon\Project\canopen.co

Configuration 2:

Configuration 3:

Configuration 4:

Figure 50: Settings > Start Options

- **Simulation mode ON/OFF**

  Only valid for the OPC Server.

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

  Only valid for the OPC Server.

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

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

- **Logic Network View visible**

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

- **Fast start ON/OFF**

  Only valid for the OPC Server.

- **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 allocation 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 that directly influence Hilscher CANopen devices, e.g. CIF 50-COM, CIF 50-COS are presented.

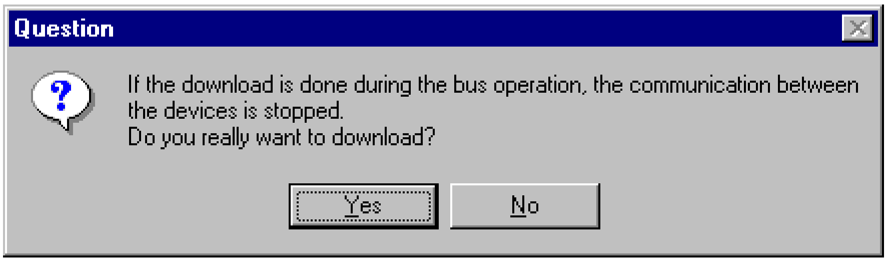
Note: Please note that this also permits an interruption of the running communication or that input and output can be switched ON or OFF.

6.2 Online to the CIF

6.2.1 Downloading the Configuration

First, the 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 CANopen will be interrupted. This warning must be confirmed.

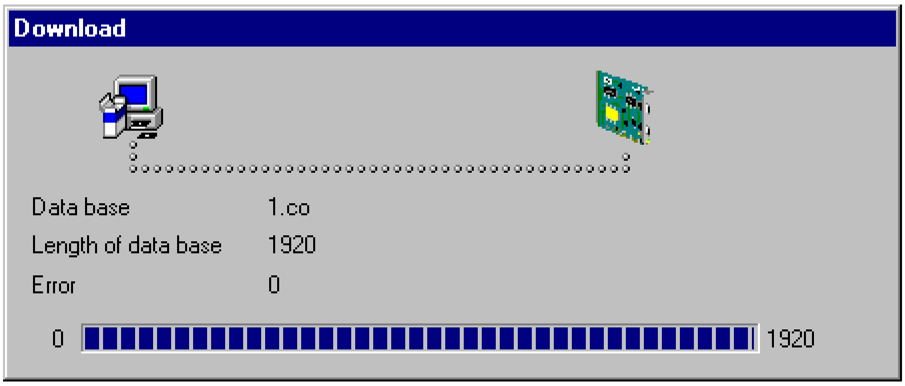


Question

If the download is done during the bus operation, the communication between the devices is stopped.
Do you really want to download?

Figure 51: Security question before Download

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



Download

Data base 1.co
Length of data base 1920
Error 0

Figure 52: Online > Download

Before the Download is executed, the configuration is checked 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.

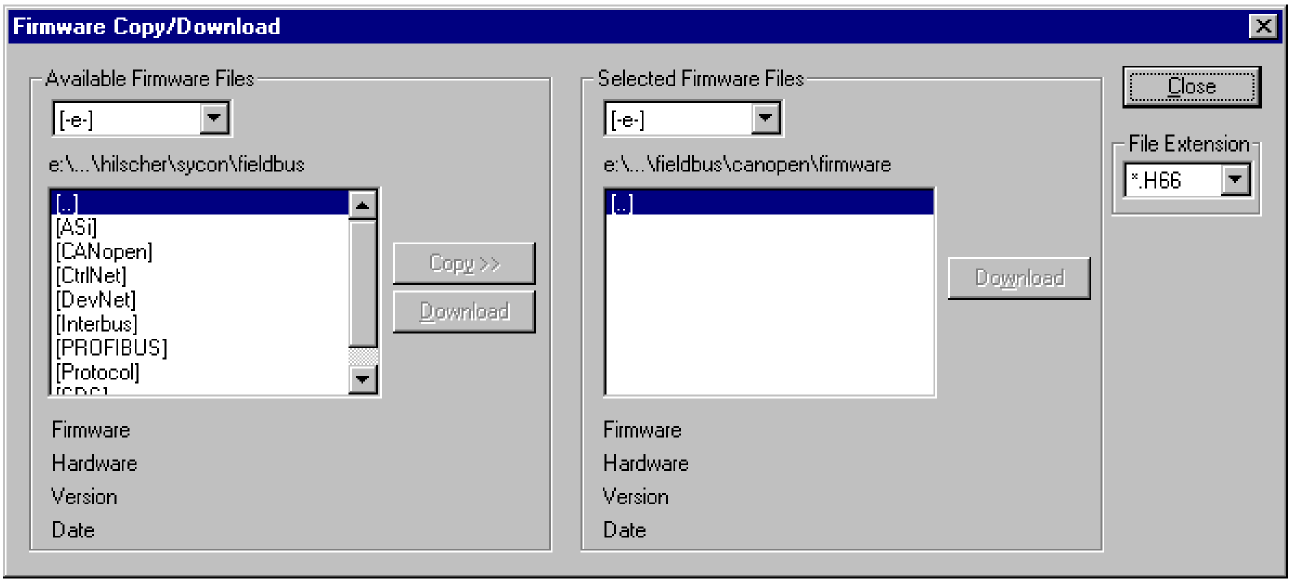
If the issue of addresses in the process data image should be carried out automatically, then the **Auto Addressing** button in the **Master Configuration** window must be activated.

The configuration is transferred into the selected device and is 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 **CANopen 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.



Firmware Copy/Download

Available Firmware Files: [-e-]

e:\...\hilscher\sycon\fieldbus

[..]
[ASi]
[CANopen]
[CtrlNet]
[DevNet]
[Interbus]
[PROFIBUS]
[Protocol]
[SDS]

Selected Firmware Files: [-e-]

e:\...\fieldbus\canopen\firmware

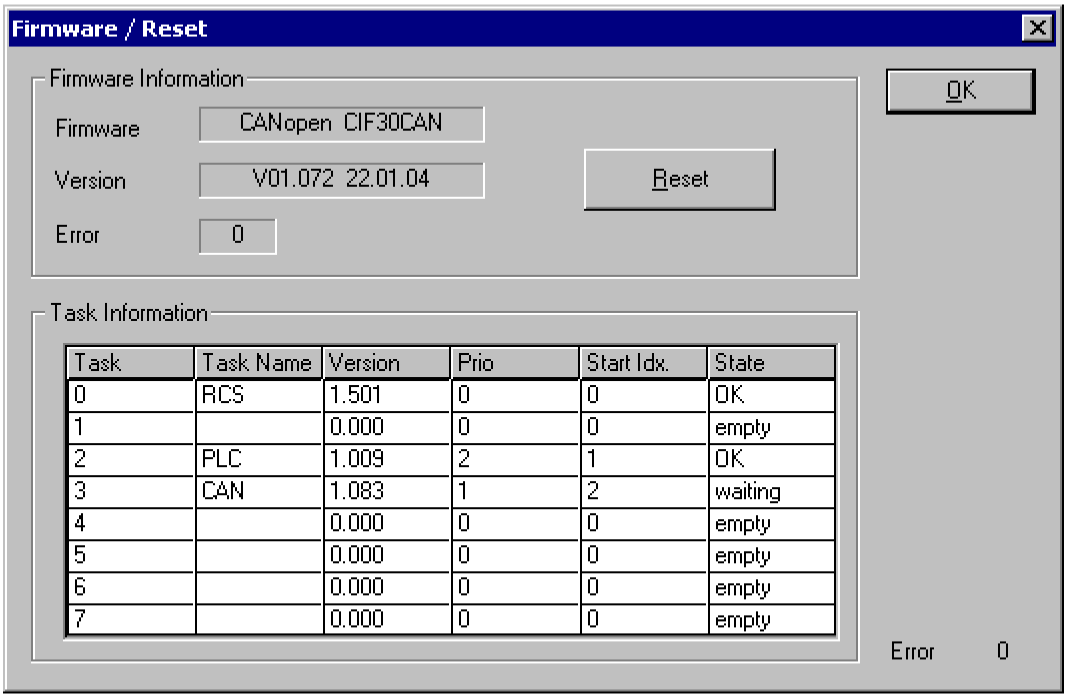
[..]

File Extension: *.H66

Figure 53: 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.



Firmware Information

Firmware: CANopen CIF30CAN

Version: V01.072 22.01.04

Error: 0

Task Information

Task	Task Name	Version	Prio	Start Idx.	State
0	RCS	1.501	0	0	OK
1		0.000	0	0	empty
2	PLC	1.009	2	1	OK
3	CAN	1.083	1	2	waiting
4		0.000	0	0	empty
5		0.000	0	0	empty
6		0.000	0	0	empty
7		0.000	0	0	empty

Error 0

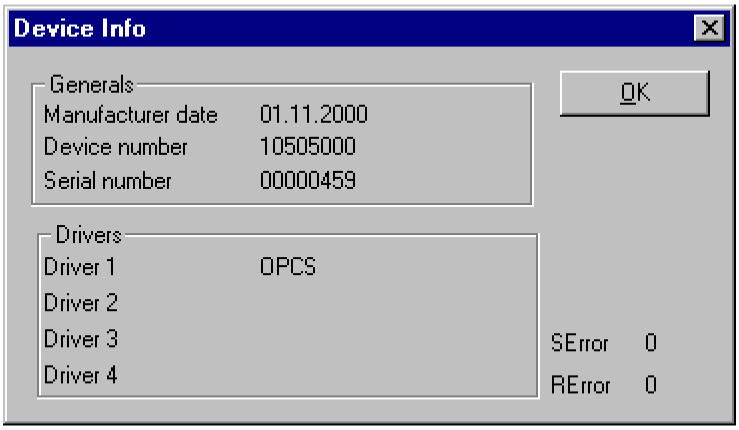
Figure 54: Online > Firmware / Reset

The device is reset with the **Reset** button.

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.



Generals

Manufacturer date 01.11.2000

Device number 10505000

Serial number 00000459

Drivers

Driver 1 OPCS

Driver 2

Driver 3

Driver 4

SError 0

RError 0

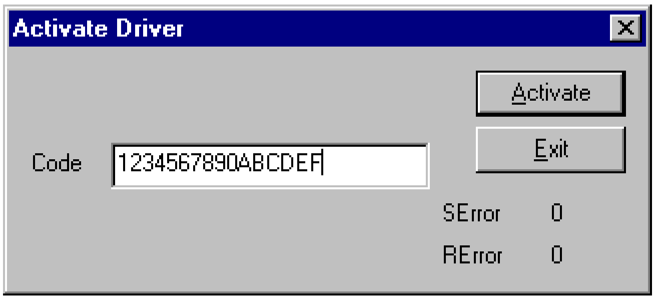
Figure 55: Online > Device Info

6.2.5 Activate Driver

The driver has to be licensed, if the software PLC or SyCon OEM is used.

If the driver was ordered by buying the SyCon, you don't need to license it because this was done before.

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.



Code 1234567890ABCDEF

SError 0

RError 0

Figure 56: Online > Activate Driver

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

6.3 Start/Stop Communication

The communication between CANopen Master and CANopen Node can be manually started or stopped.

First the desired device must be chosen with a left mouse click on the symbol of the device. Then select the **Online > Communication start** or **Online > Communication stop** menu.

6.4 Diagnostic Functions

The following table shows diagnostic functions and their using in case of

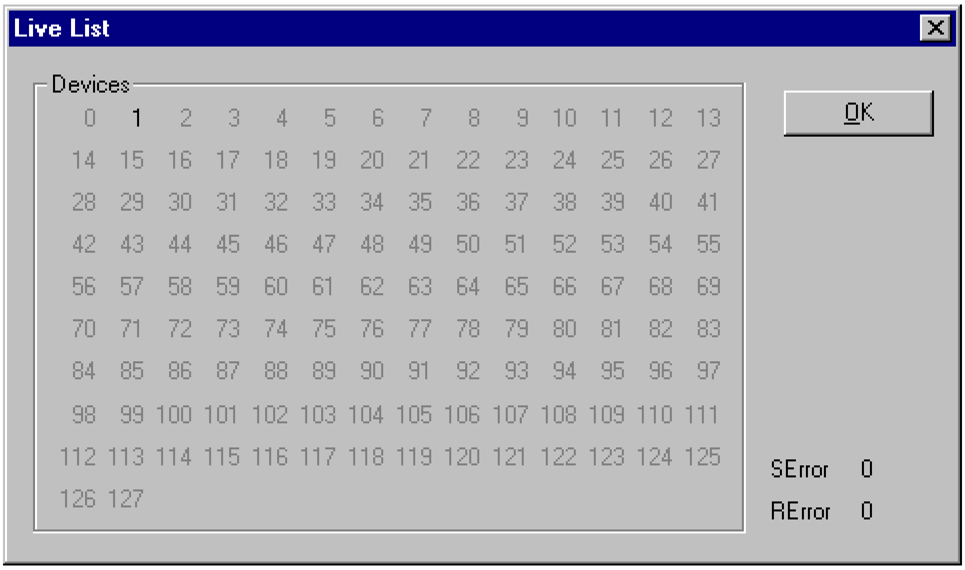
- Hilscher CANopen Master devices
- Hilscher CANopen Nodes (Slaves).

Diagnostic Function	**Using**	**Usable for Hilscher CANopen Master devices**	**Usable for Hilscher CANopen Nodes**
Live List	Determine, which devices are connected to the Hilscher CANopen Master device.	Yes	No, only for Hilscher CANopen Master devices
Debugmode (CANopen)	Determine, to which CANopen Nodes the Hilscher CANopen Master has communication	Yes	No, only for Hilscher CANopen Master devices
Global State Field	Status information of the Hilscher CANopen Master	Yes	No, only for Hilscher CANopen Master devices
Extended Device Diagnostic	Statistic information and status information from the Hilscher CANopen device	Yes	Yes

Table 33: Overview Diagnostic Functions

6.4.1 Live List

First the desired device must be chosen with a left mouse click on the symbol of the device. Then select the **Online > Live List** menu and obtain an overview over all active devices at the CANopen network.



Live List

Devices

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	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79	80	81	82	83
84	85	86	87	88	89	90	91	92	93	94	95	96	97
98	99	100	101	102	103	104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119	120	121	122	123	124	125
126	127												

SError 0

RError 0

Figure 57: Online > Live List

Generally all devices are displayed grey. At the bus detected Nodes are represented black on the basis their appropriate Node address.

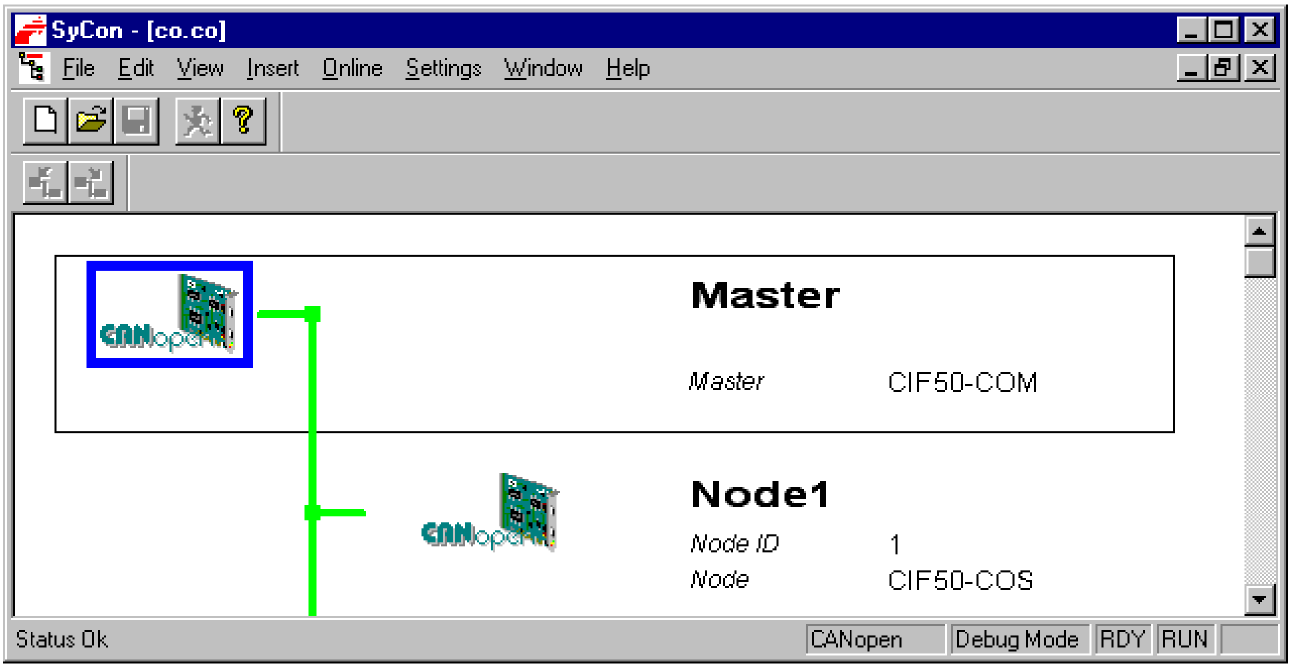
6.4.2 Debugmode (CANopen)

Click the menu item **Online > Start Debug Mode**. Then the System Configurator cyclically interrogates the status of the network communication from the CIF, COM or PKV and the individual conditions of the Nodes.

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

6.4.2.1 The Debugwindow

When the debug session is started 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.



Master

Master CIF50-COM

Node1

Node ID 1

Node CIF50-COS

Figure 58: The Debugwindow

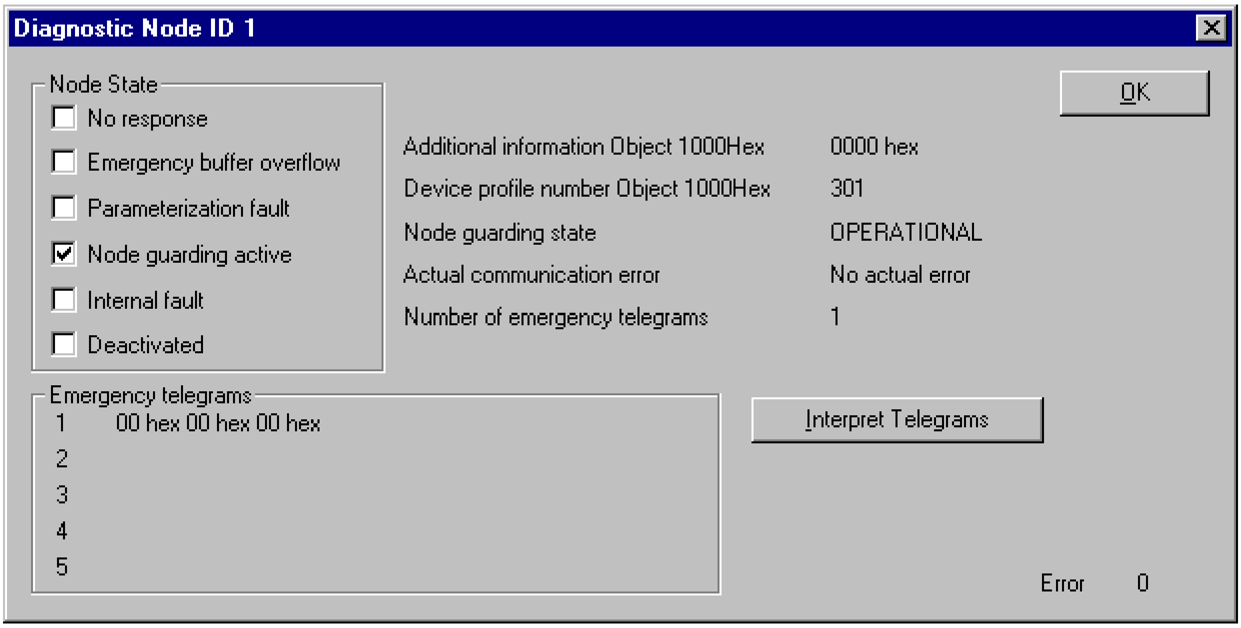
If a diagnostic information is available for a specific device, next to the device Icon the text **Diag** appears in red. To get further device specific diagnostic information then double-click on the device itself or set the focus to the device and select **Online > Device Diagnostic**.

Note: Only if the CANopen Node supports the Nodeguarding, the CANopen Master (NMT-Master) can recognize that the Node has failed.

6.4.2.2 CANopen Node specific 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 displayed in red colour otherwise it is green. SyCon also displays the letters **Diag**, if the device signals a diagnostic information. This information is displayed closer if you click with the mouse onto the corresponding device in debug mode.

To activate the debug mode you have to mark the Master with a left mouse click and select the menu **Online > Start Debug Mode**. Now set the focus at the Node (left mouse click) and select the menu **Online > Device Diagnostic** to show the CANopen Device Diagnostic. To end the Debug Mode you have to mark the Master again and select the menu **Online > Stop Debug Mode**.


Diagnostic Node ID 1

Node State
- ☐ No response
- ☐ Emergency buffer overflow
- ☐ Parameterization fault
- ☑ Node guarding active
- ☐ Internal fault
- ☐ Deactivated

Additional information Object 1000Hex	0000 hex
Device profile number Object 1000Hex	301
Node guarding state	OPERATIONAL
Actual communication error	No actual error
Number of emergency telegrams	1

Emergency telegrams
1 00 hex 00 hex 00 hex
2
3
4
5

Error 0

Figure 59: Online > Device Diagnostic (CANopen Standard Diagnostic)

Description see next page.

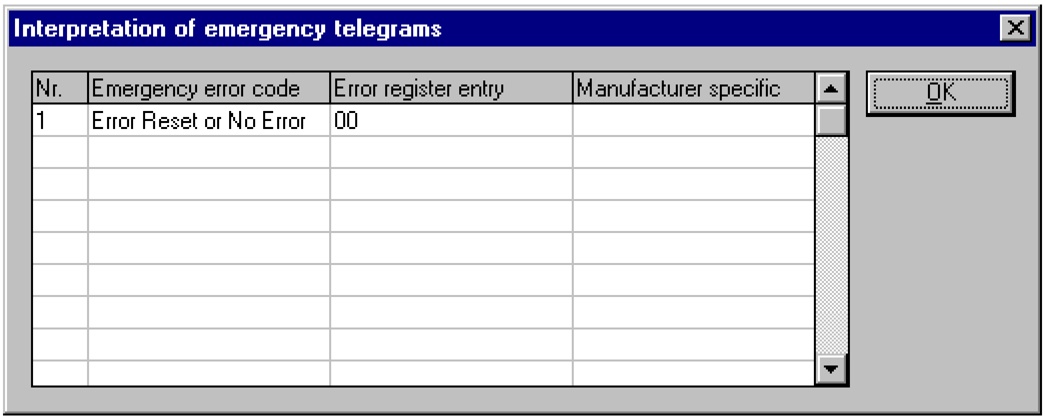
The individual bits in the **Device Diagnostic** and their meaning:

Bits in the Device Diagnostic	**Meaning**
No Response	The Node is configured but is not present in the network. Please check the physical connection between the Master and this Node. Compare the selected baudrate of the Node with the baudrate of the Master, if this baudrate is supported by the Node. Further more compare the Node address.
Emergency buffer overflow	CANopen defines a special reserved emergency channel for each Node 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 are saved in an internal buffer on the Master. The buffer will be cleared when SyCon reads out this buffer and shows the telegrams in the lower **Emergency telegrams** window. If this buffer now is overstepped it will cause lost telegrams. In this case the buffer overflow event is reported. By means of CANopen Communication Profile defined emergency error codes the emergency condition is specified. Collected and shown **Emergency telegrams** in the lower table can be interpreted textual by clicking onto **the Interpret Telegrams** button.
Parameterization fault	The Master compares the configured Device **Profile** and the corresponding **Device Type** value of the **Node Configuration** window with the real physically present ones in the Node by reading out the Node object 1000H. If the Master detects differences between the values it will report the Parameterization Fault. The real 1000H containment that is just read out online from the Node is shown similar as value in this window behind the **Device profile number Object 1000H** and **Additional information Object 1000H** entry.
Node guarding active	As soon as the master has finished up the configuration phase of the Node it will start the cyclic Node guarding mechanism and set the Node guarding active indication flag. Remember: the Node guarding will only be activated if neither the Guard time nor the Life time factor in the Node Configuration window is zero.
Internal fault	The internal fault indication serves to report master internal fatal error situations. If it is reported the office Hilscher should be called.
Deactivated	This bit is set by the master automatically, if the Node state was configured to **Deactivate Node in actual configuration** in the **Node Configuration** window.

Table 34: Meaning of the bits in the Device Diagnostic

6.4.2.3 Emergency Telegrams

Emergency telegrams are sent by the Node when an internal event occurs if a Node enters. The CANopen Master can buffer maximally 5 Emergency telegrams.



Nr.	Emergency error code	Error register entry	Manufacturer specific
1	Error Reset or No Error	00	

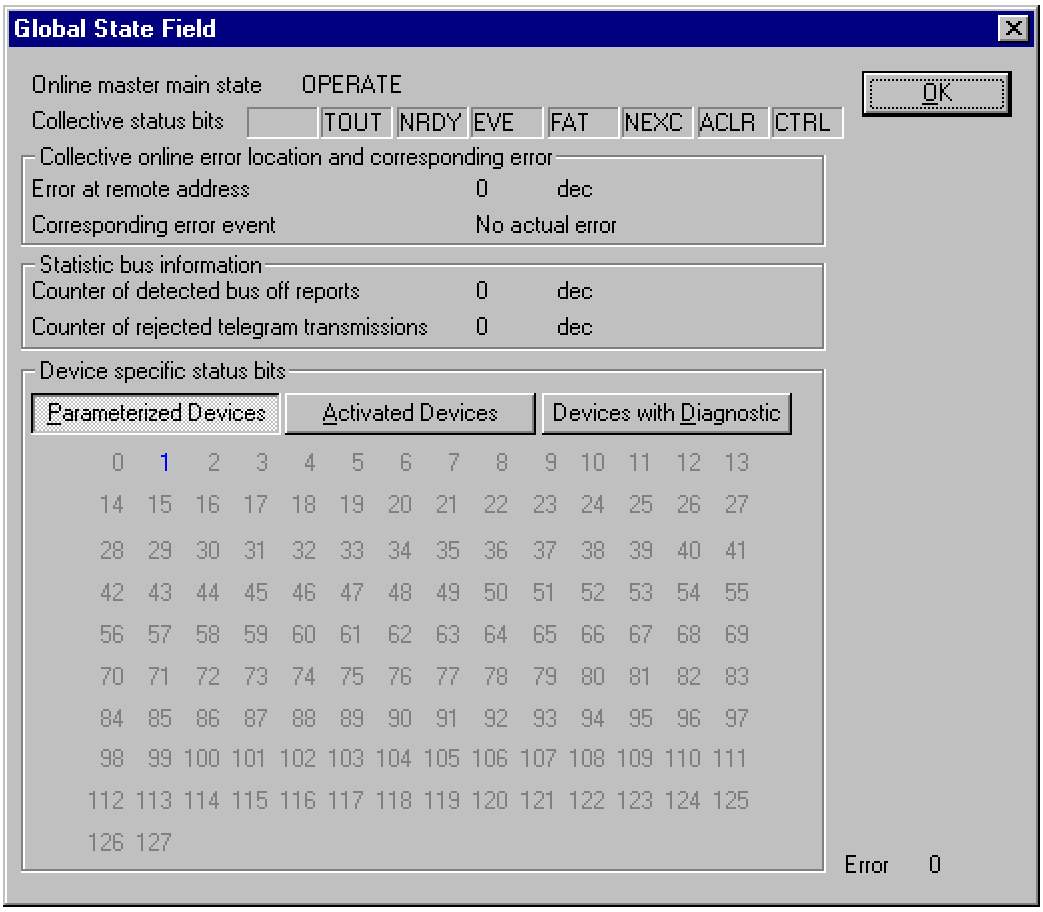
Figure 60: Online > Device Diagnostic > Interpretation of emergency telegrams

A table with the Error Codes is described in section *Emergency Telegram Error Codes* at page *192*.

Note: The table Emergency Error Codes is a general list. For the exact meaning it is referred to the manual of the Node manufacturer.

6.4.3 Global State Field

With the menu option **Online > Global State Field** opens a window in that cyclically statistic about the bus status and connected devices are shown.



Global State Field

Online master main state OPERATE

Collective status bits: TOUT NRDY EVE FAT NEXC ACLR CTRL

Collective online error location and corresponding error

Error at remote address 0 dec

Corresponding error event No actual error

Statistic bus information

Counter of detected bus off reports 0 dec

Counter of rejected telegram transmissions 0 dec

Device specific status bits

Error 0

Figure 61: Online > Global State Field

The first row displays the main status of the Master. It can take the status **OPERATE** or **STOP** or **OFFLINE**.

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

Bus error	Meaning
TOUT	Timeout Error
NRDY	HOST-NOT-READY-NOTIFICATION shows, if the application program is ready or not. If this bit is set the application program is not ready to communicate.
EVE	EVENT-ERROR the CAN chip has detected transmission errors. The number of detected events are 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 because of heavy bus error, no further bus communication is possible.
NEXC	NON-EXCHANGE-ERROR At least one Node has not reached the data exchange state and no process data are exchange with it.
ACLR	AUTO-CLEAR-ERROR device stopped the communication to all Nodes and reached the auto-clear end state.
CTRL	CONTROL-ERROR a parameterization error has occurred.

Table 35: Meaning of collective status bits in the Global State Field

Further contents are given:

Collective online error location and corresponding error gives the address of the incorrect station and the lining up error in plain text.

Statistic bus information gives the number of detected bus short-circuits and 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 colored numbers. You can see the diagnostic by double-clicking at a highlighted station address of a device.

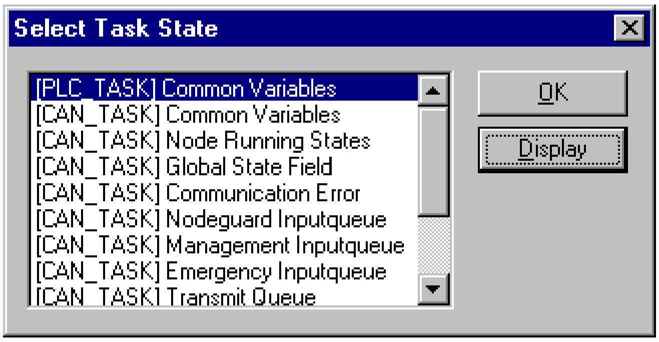
This displaying is cyclically updated.

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

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



[PLC_TASK] Common Variables
[CAN_TASK] Common Variables
[CAN_TASK] Node Running States
[CAN_TASK] Global State Field
[CAN_TASK] Communication Error
[CAN_TASK] Nodeguard Inputqueue
[CAN_TASK] Management Inputqueue
[CAN_TASK] Emergency Inputqueue
[CAN_TASK] Transmit Queue

Figure 62: Online > Extended Device Diagnostic

First the specification for the CANopen Master and far down for the CANopen Slave follows.

6.4.4.1 Extended Device Diagnostic CANopen Master

Task/Taskstate	**Page**
PLC_TASK Common Variables	*158*
CAN_TASK Common Variables	*159*
CAN_TASK Node Running State	*160*
CAN_TASK Global State Field	*161*
CAN_TASK Communication Error	*161*
CAN_TASK Nodeguard Inputqueue	*162*
CAN_TASK Management Inputqueue	*162*
CAN_TASK Emergency Inputqueue	*163*
CAN_TASK Transmit Queue	*163*
CAN_TASK CMS Domain Services	*164*
CAN_TASK Timeout Counter	*165*
CAN_TASK Node Init Counter	*166*

Table 36: CANopen Master Taskstate

6.4.4.2 Extended Device Diagnostic CANopen Node

Task/Taskstate	**Page**
PCL_TASK Common Variables	*167*
COS_TASK Common Variables	*168*
COS_TASK User Communication	*170*
COS_TASK Node Management	*171*
COS_TASK PDO Transfer	*172*
COS_TASK SDO Transfer	*173*
COS_TASK Object Dictionary	*174*
COS_TASK Receive Queue	*174*
COS_TASK Transmit Queue	*175*

Table 37: CANopen Node Taskstate

6.5 User Data Transfer

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

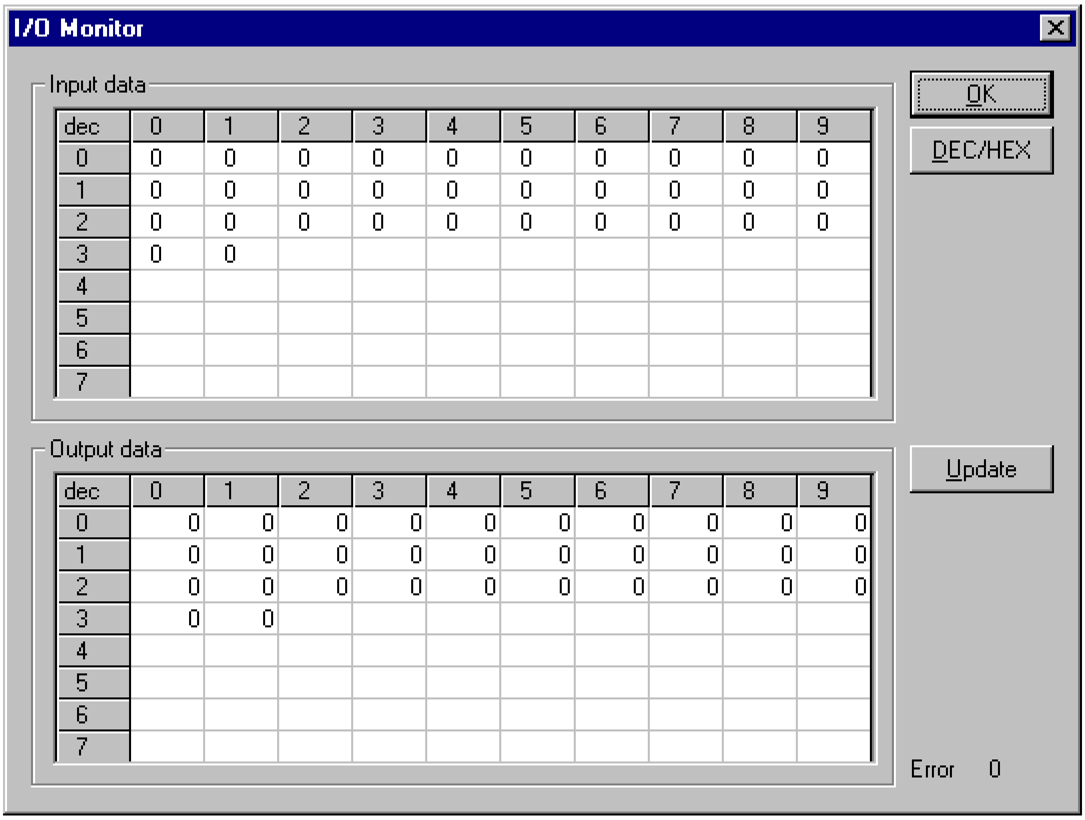
- Hilscher CANopen Master devices
- Hilscher CANopen Nodes

User data transfer function	**Usage**	**Usable with Hilscher CANopen Master devices**	**Usable with Hilscher CANopen Slave devices**
I/O-Monitor	Read input data and set output data. (cyclic I/O data exchange)	Yes	Yes
I/O Watch	Read input data and set output data. (cyclic I/O data exchange)	Yes	No
Read Objects (SDO Upload)	Read objects (SDO Upload)	Yes	No
Write Object (SDO Download)	Write objects (SDO Download)	Yes	No

Table 38: Overview User Data Transfer

6.5.1 I/O-Monitor

This is an easy way of viewing and changing the first 32 Bytes of the process data image. The I/O Monitor is called up with the menu **Online > I/O Monitor**.



Input data

dec	0	1	2	3	4	5	6	7	8	9
0	0	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										

Output data

dec	0	1	2	3	4	5	6	7	8	9
0	0	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										

Error 0

Figure 63: Online > I/O-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 depiction are shown, also when these Bytes have not been occupied by the configuration.

The display is always in a Byte manner.

A more comfortable display is offered by the I/O-Watch Monitor that is described in the next section.

6.5.2 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) resp. 1.140 (DP-Master)
InterBus Master	2.040
CANopen Master	1.040
DeviceNet Master	1.058
AS-Interface Master	1.010

Table 39: Firmware for I/O Watch function

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

Preconditions:

- The project/configuration already exists, containing a CANopen Master and the CANopen Node(s) as described in section *Getting Started - Configuration Steps* on page *23*.
- The Configuration has been downloaded into the CANopen 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
Logical network view	Tag list	I/O Watch

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

Project > Master > Node > Module > (possible) Submodul

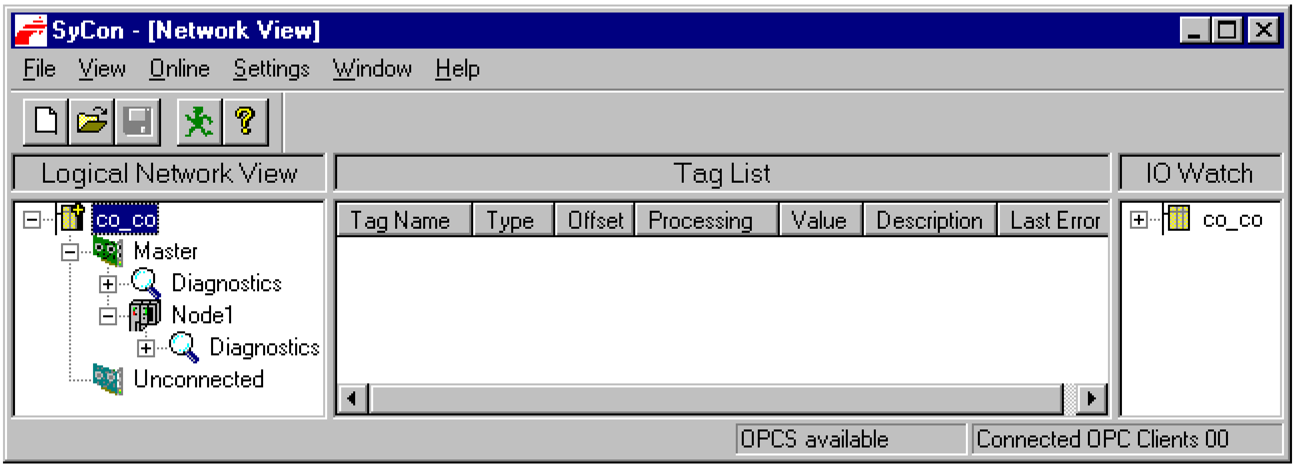
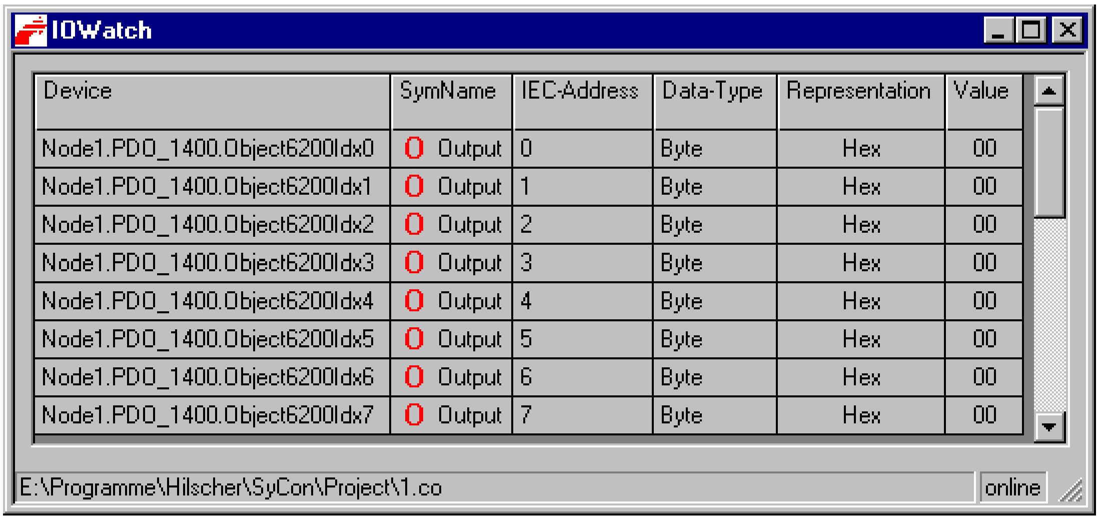


Figure 64: Logical Network View and I/O Watch

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.
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 I/O Watch appears.
7. A table shows the Device, Symbolic Name, IEC Address (Offset), Data type Representation and Value.
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
Node1.PDO_1400.Object6200Idx0	O Output	0	Byte	Hex	00
Node1.PDO_1400.Object6200Idx1	O Output	1	Byte	Hex	00
Node1.PDO_1400.Object6200Idx2	O Output	2	Byte	Hex	00
Node1.PDO_1400.Object6200Idx3	O Output	3	Byte	Hex	00
Node1.PDO_1400.Object6200Idx4	O Output	4	Byte	Hex	00
Node1.PDO_1400.Object6200Idx5	O Output	5	Byte	Hex	00
Node1.PDO_1400.Object6200Idx6	O Output	6	Byte	Hex	00
Node1.PDO_1400.Object6200Idx7	O Output	7	Byte	Hex	00

E:\Programme\Hilscher\SyCon\Project\1.co online

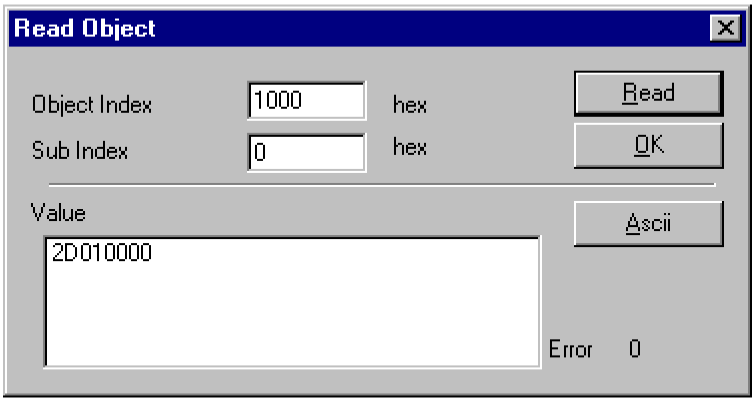
Figure 65: I/O Watch window

In the column representation can be selected the data type: Bit Pattern, Char, decimal Signed, decimal Unsigned, Hex

6.5.3 Read Objects (SDO Upload)

With this function you can execute the CANopen services read object and write object based on the current configuration.

As the first you must select the required device, by clicking with the left mouse button on the symbol Node. Then select the menu **Online Read Object** or **Online > Write Object**.



Read Object

Object Index 1000 hex

Sub Index 0 hex

Value

2D010000

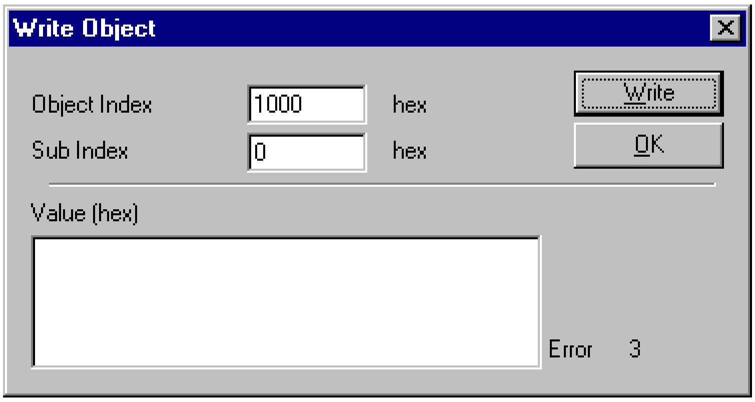
Error 0

Figure 66: Online > Read Object

6.5.4 Write Object (SDO Download)

With this function you can execute the CANopen services read object and write object based on the current configuration.

As the first you must select the required device, by clicking with the left mouse button on the symbol Node. Then select the menu **Online Read Object** or **Online > Write Object**.



Write Object

Object Index 1000 hex

Sub Index 0 hex

Value (hex)

Error 3

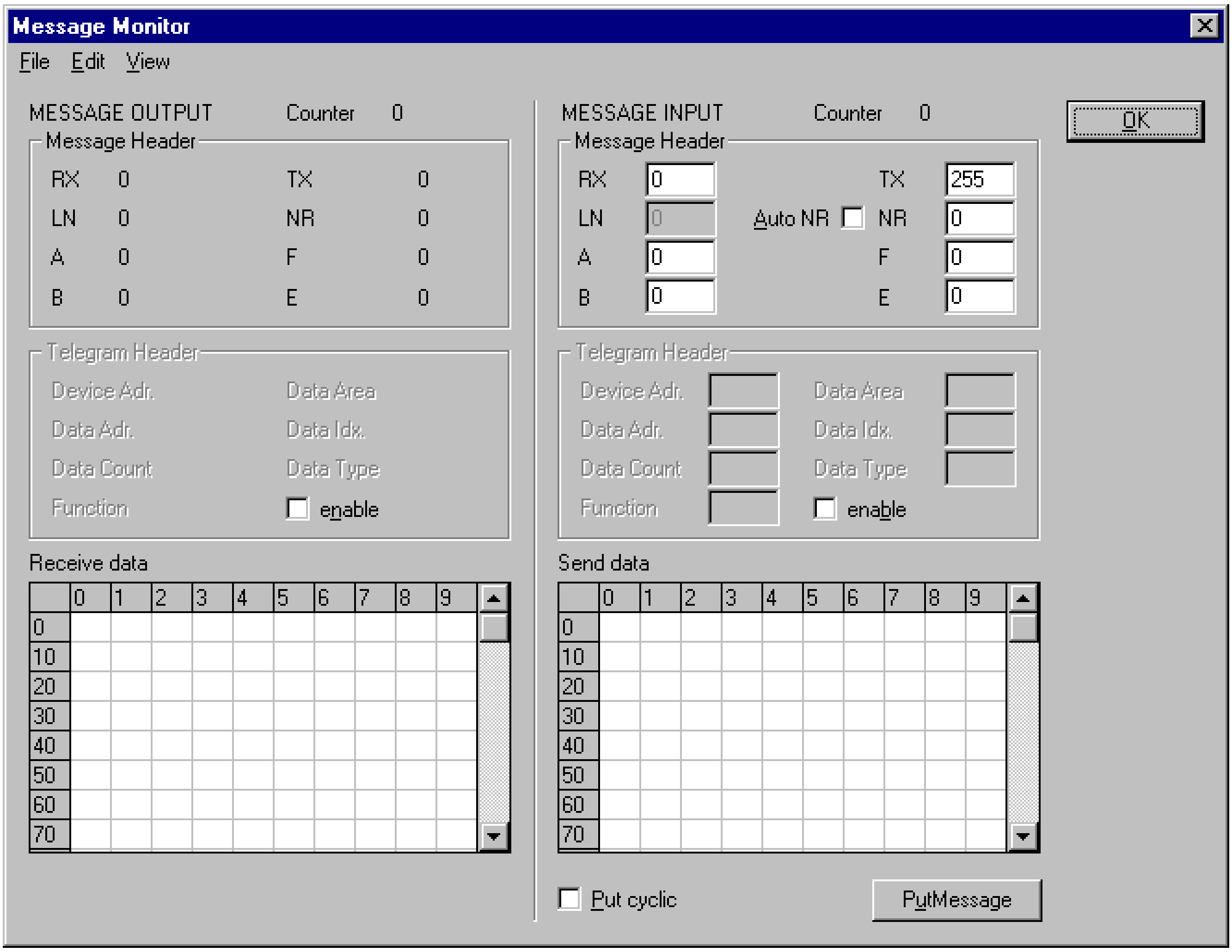
Figure 67: Online > Write Object

Objects in CANopen are addresses in the manner Object-Index and corresponding Sub-Index. Both values must be specified in the selected window. Press **Read** or **Write** button to start the action. SyCon informs about success and failure of the action.

6.6 Message Monitor

The Message Monitor permits access to the Mailbox of the CIF. The usage of the Message Monitor assumes advanced knowledge on the part of 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 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 □ 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 □ 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 □ enable

Send data

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

□ Put cyclic

Figure 68: 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.

6.6.1 Message Monitor for Using LSS/LMT

For setting the baud rate and the Node address LSS/LMT services can be used for some Nodes.

The LSS/LMT Master sends telegrams to the LSS/LMT Slave with CAN telegram identifier 2021 (07E5H). The LSS/LMT Slave replies to the LSS/LMT Master with the CAN telegram identifier 2020 (07E4H).

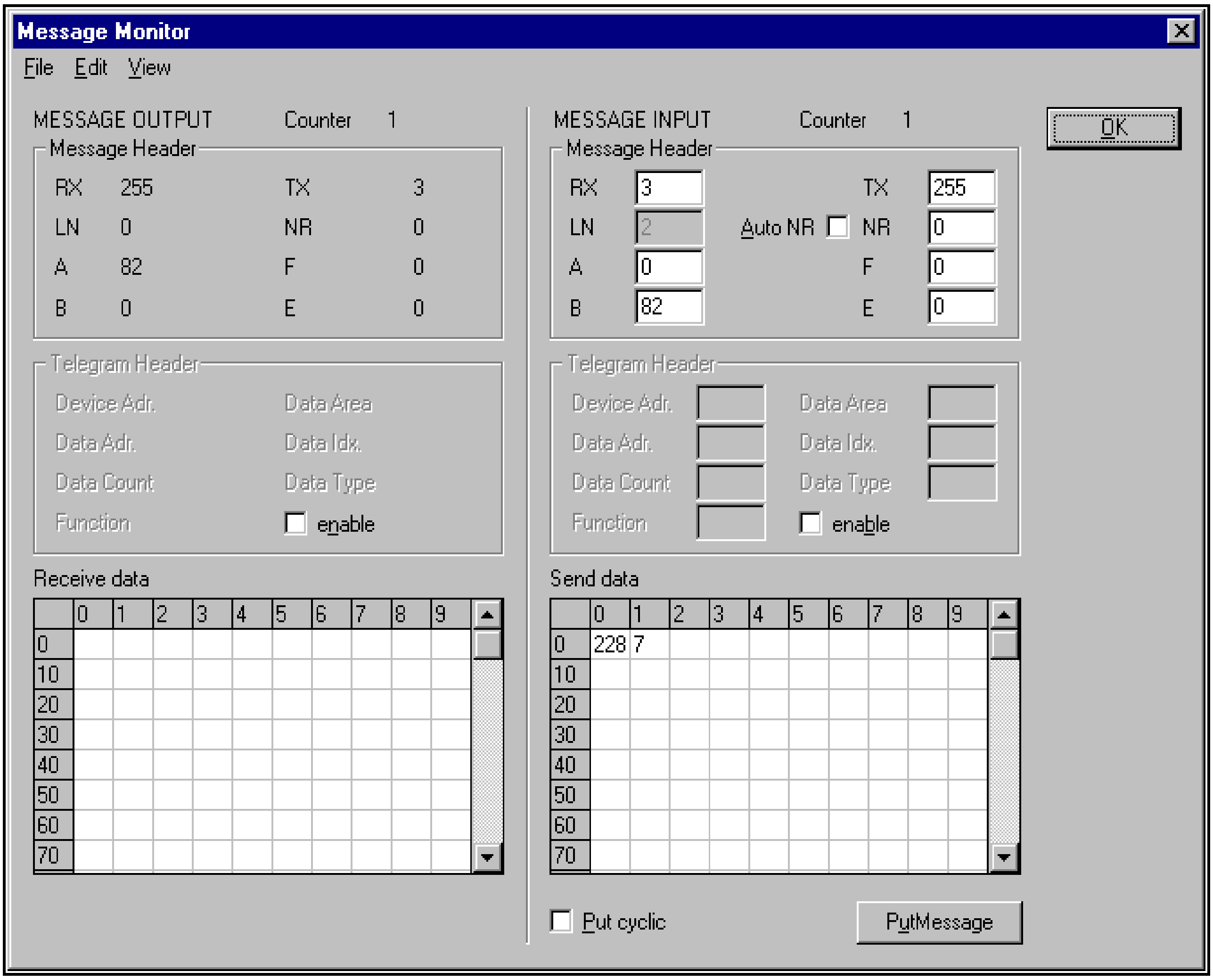
Note: It may be coupled only one Node to the Master at a time.

First the baud rate of the Hilscher CANopen Master have to be set equal to the baud rate of the Node.

Then the CAN telegram identifier 2020 has to be set with the receive filter.

Message for Setting the Receive Filter CAN (Layer 2)		
Message Header		
Rx = 3 (fixed)	Tx = 255	
Ln = (is calculated)	Nr = 0...255	
A = 0	F = 0	
B = 82	E = 0	
Send Data	**Meaning for CAN**	**Range of value**
	CAN Receive ID Part 1 (LSB)	228
	CAN Receive ID Part 2 (MSB)	7

Table 40: Message Monitor for LSS/LMT > Setting the Receive Filter



MESSAGE OUTPUT Counter 1

Message Header

RX	255	TX	3
LN	0	NR	0
A	82	F	0
B	0	E	0

Telegram Header

Device Adr. Data Area
Data Adr. Data Idx.
Data Count Data Type
Function □ enable

Receive data

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

MESSAGE INPUT Counter 1

Message Header

RX	3		TX	255
LN	2	Auto NR □	NR	0
A	0		F	0
B	82		E	0

Telegram Header

Device Adr. Data Area
Data Adr. Data Idx.
Data Count Data Type
Function □ enable

Send data

	0	1	2	3	4	5	6	7	8	9
0	228	7								
10										
20										
30										
40										
50										
60										
70										

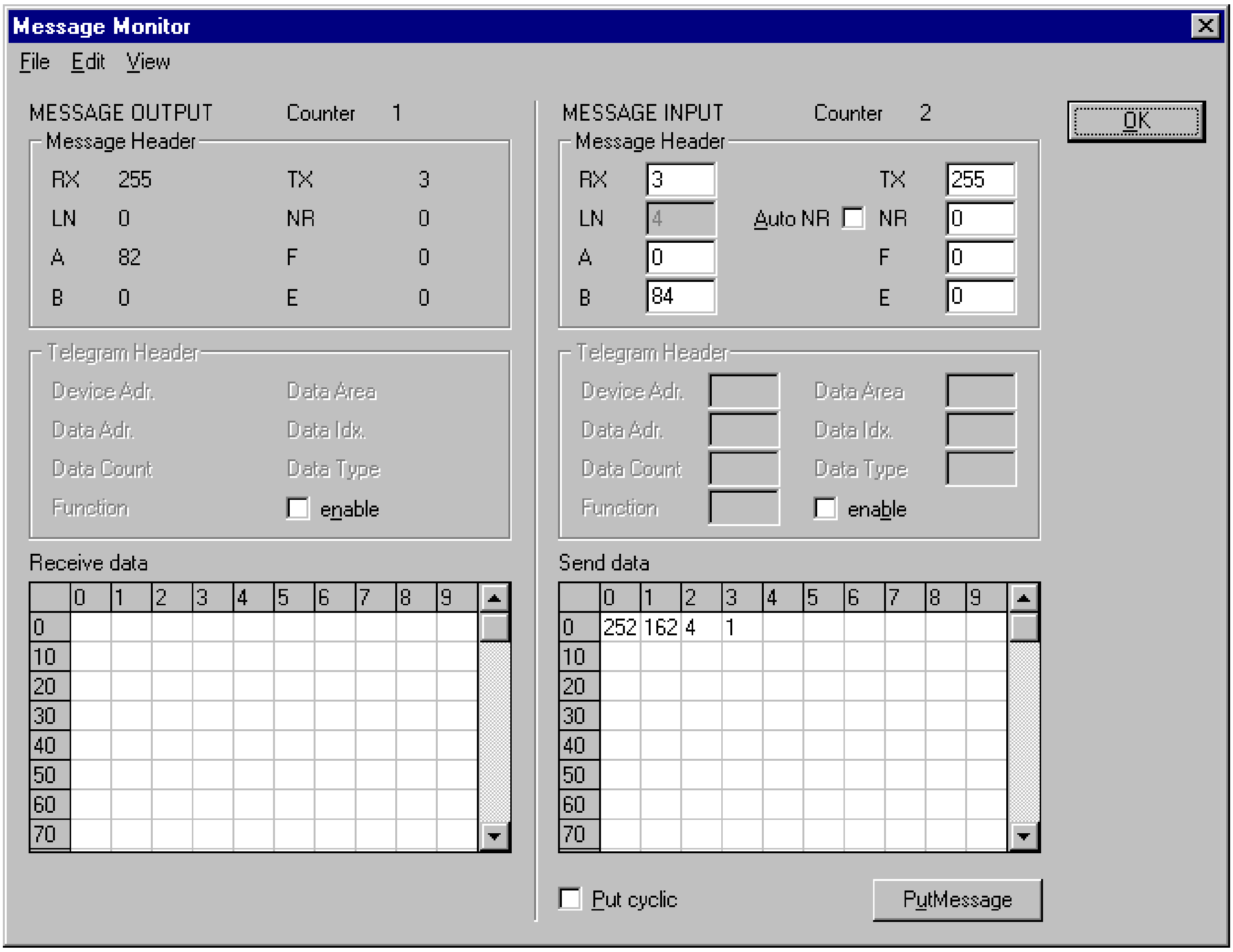
□ Put cyclic

Figure 69: Message Monitor for LSS/LMT > Set the Receive Filter

1. Switch in configuration mode

Message for Sending via CAN (Layer 2)		
Message Header		
Rx = 3 (fixed)	Tx = 255	
Ln = (is calculated)	Nr = 0...255	
A = 0	F = 0	
B = 84	E = 0	
Send Data	**Meaning for CAN**	**Range of value**
	CAN Receive ID Part 1 (LSB)	252
	CAN Receive ID Part 2 (MSB)	162
	Send data 1: Mode Global Service	4
	Send data 2: Config Mode	1

Table 41: Message Monitor LSS/LMT (1) > Switch Configuration Mode on



MESSAGE OUTPUT Counter 1

Message Header

RX	255	TX	3
LN	0	NR	0
A	82	F	0
B	0	E	0

Receive data

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

MESSAGE INPUT Counter 2

Message Header

RX	3	TX	255
LN	4	Auto NR ☐ NR	0
A	0	F	0
B	84	E	0

Send data

	0	1	2	3	4	5	6	7	8	9
0	252	162	4	1						
10										
20										
30										
40										
50										
60										
70										

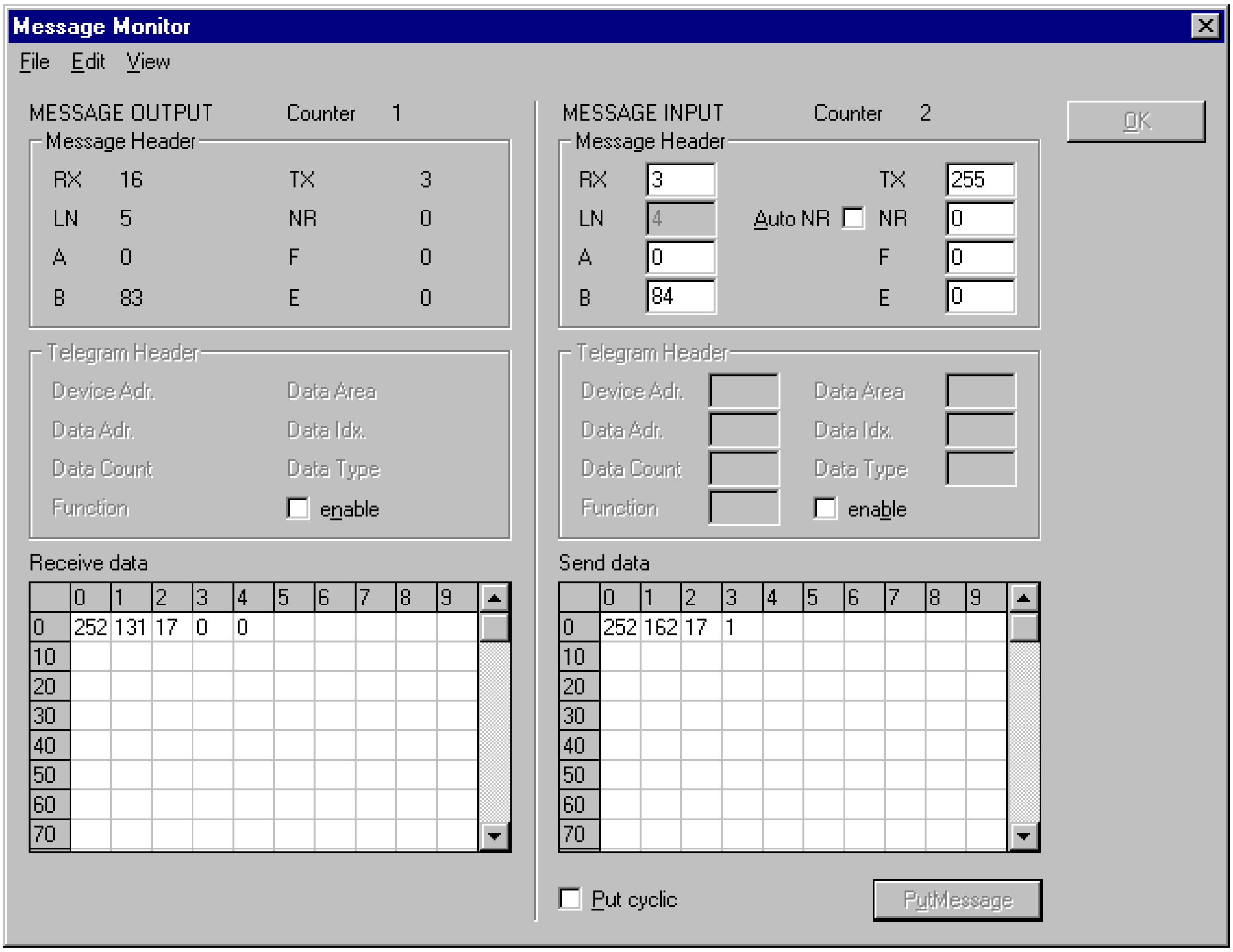
☐ Put cyclic

Figure 70: Message Monitor LSS/LMT (1) > Switch Configuration Mode on

2. Set Node Address

Message for Sending via CAN (Layer 2)		
Message Header		
Rx = 3 (fixed)	Tx = 255	
Ln = (is calculated)	Nr = 0...255	
A = 0	F = 0	
B = 84	E = 0	
Send Data	**Meaning for CAN**	**Range of value**
	CAN Receive ID Part 1 (LSB)	252
	CAN Receive ID Part 2 (MSB)	162
	Send data 1: Set Node ID	17
	Send data 2: Node Address	1...127

Table 42: Message Monitor LSS/LMT (2) > Set Node Address



MESSAGE OUTPUT Counter 1

Message Header

RX 16 TX 3
LN 5 NR 0
A 0 F 0
B 83 E 0

Telegram Header

Device Adr. Data Area
Data Adr. Data Idx.
Data Count Data Type
Function □ enable

Receive data

	0	1	2	3	4	5	6	7	8	9
0	252	131	17	0	0					
10										
20										
30										
40										
50										
60										
70										

MESSAGE INPUT Counter 2

Message Header

RX 3 TX 255
LN 4 Auto NR □ NR 0
A 0 F 0
B 84 E 0

Telegram Header

Device Adr. Data Area
Data Adr. Data Idx.
Data Count Data Type
Function □ enable

Send data

	0	1	2	3	4	5	6	7	8	9
0	252	162	17	1						
10										
20										
30										
40										
50										
60										
70										

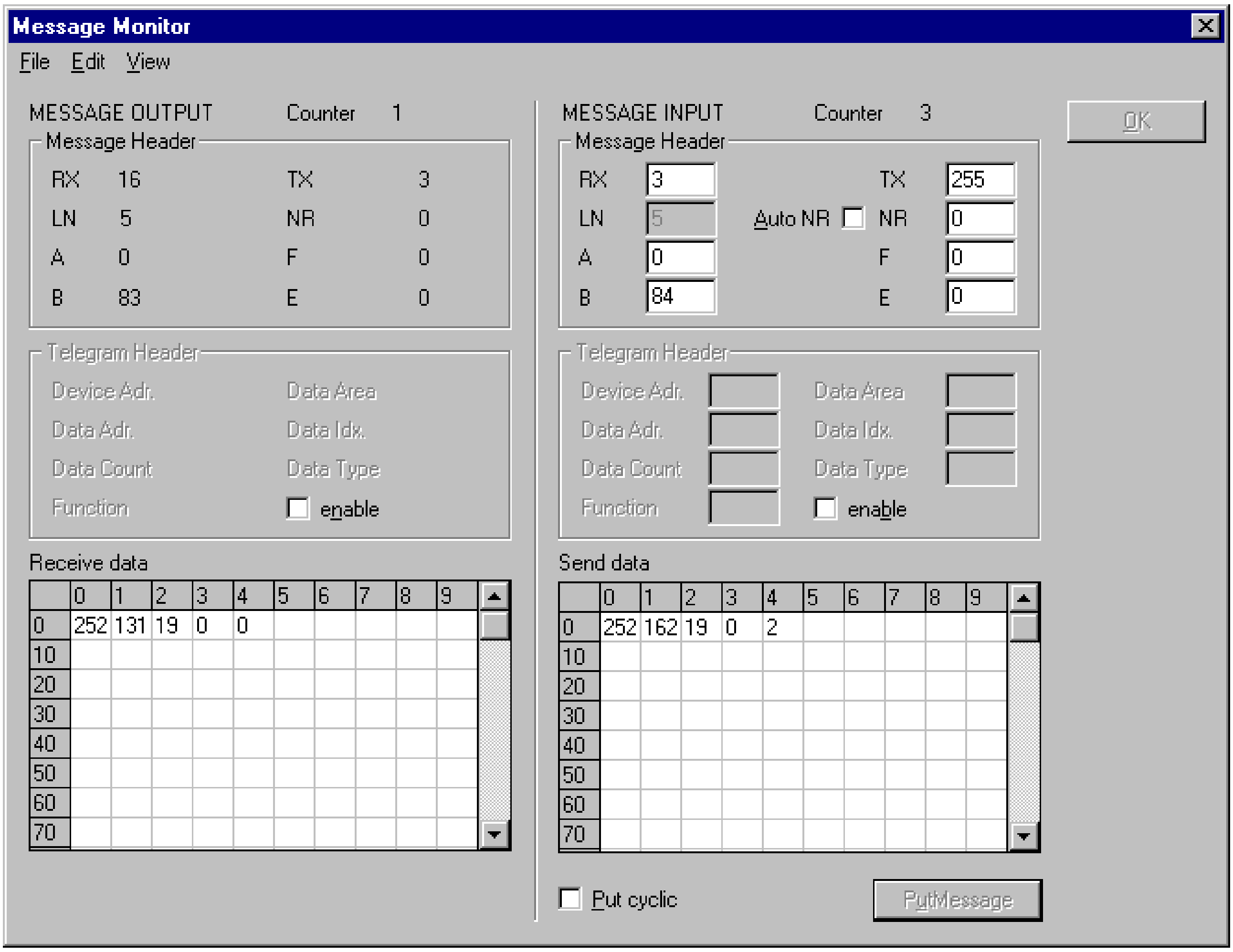
□ Put cyclic

Figure 71: Message Monitor LSS/LMT (2) > Set Node Address

3. Set Baud Rate

Message for Sending via CAN (Layer 2)		
Message Header		
Rx = 3 (fixed)	Tx = 255	
Ln = (is calculated)	Nr = 0...255	
A = 0	F = 0	
B = 84	E = 0	
Send Data	**Meaning for CAN**	**Range of value**
	CAN Receive ID Part 1 (LSB)	252
	CAN Receive ID Part 2 (MSB)	163
	Send data 1: Set Baud Rate	19
	Send data 2: Table	0 (Standard Table) 128...255
	Send data 3: Baud Rate	In case of table 0: 0 = 1 Mbit/s 1 = 800 kbit/s 2 = 500 kbit/s 3 = 250 kbit/s 4 = 125 kbit/s 5 = 50 kbit/s 6 = 20 kbit/s

Table 43: Message Monitor LSS/LMT (3) > Set Baud Rate



MESSAGE OUTPUT Counter 1

Message Header

RX	16	TX	3
LN	5	NR	0
A	0	F	0
B	83	E	0

Receive data

	0	1	2	3	4	5	6	7	8	9
0	252	131	19	0	0					
10										
20										
30										
40										
50										
60										
70										

MESSAGE INPUT Counter 3

Message Header

RX	3	TX	255
LN	5	Auto NR □ NR	0
A	0	F	0
B	84	E	0

Send data

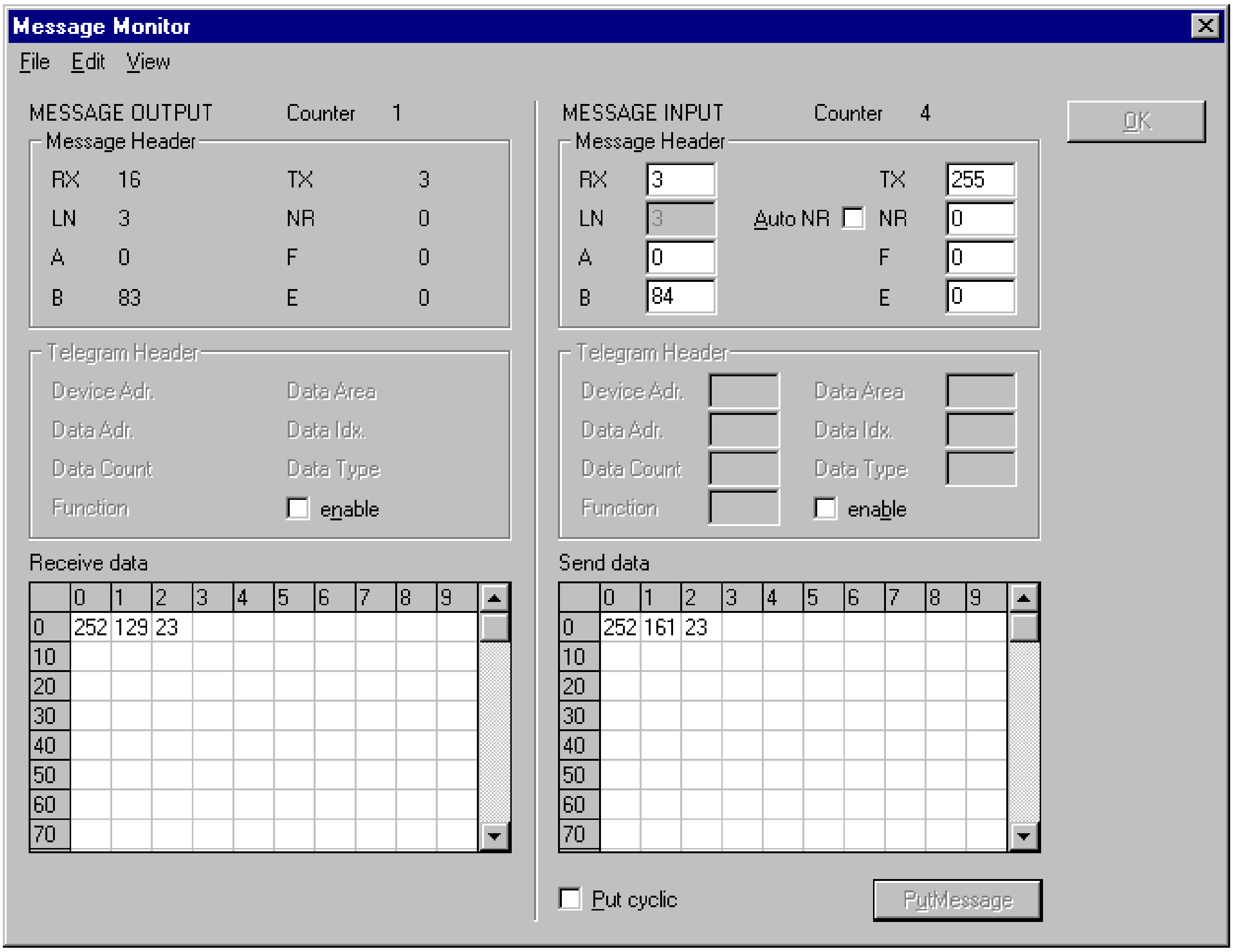
	0	1	2	3	4	5	6	7	8	9
0	252	162	19	0	2					
10										
20										
30										
40										
50										
60										
70										

Figure 72: Message Monitor LSS/LMT (3) > Set Baud Rate

4. Save Configuration

Message for Sending via CAN (Layer 2)		
Message Header		
Rx = 3 (fixed)	Tx = 255	
Ln = (is calculated)	Nr = 0...255	
A = 0	F = 0	
B = 84	E = 0	
Send Data	**Meaning for CAN**	**Range of value**
	CAN Send ID Part 1	252
	CAN Send ID Part 2	161
	Send data 1: Save Configuration	23

Table 44: Message Monitor LSS/LMT (4) > Save Configuration



MESSAGE OUTPUT Counter 1

Message Header

RX	16	TX	3
LN	3	NR	0
A	0	F	0
B	83	E	0

Telegram Header

Device Adr. Data Area

Data Adr. Data Idx.

Data Count Data Type

Function □ enable

Receive data

	0	1	2	3	4	5	6	7	8	9
0	252	129	23							
10										
20										
30										
40										
50										
60										
70										

MESSAGE INPUT Counter 4

Message Header

RX	3		TX	255
LN	3	Auto NR □	NR	0
A	0		F	0
B	84		E	0

Telegram Header

Device Adr. Data Area

Data Adr. Data Idx.

Data Count Data Type

Function □ enable

Send data

	0	1	2	3	4	5	6	7	8	9
0	252	161	23							
10										
20										
30										
40										
50										
60										
70										

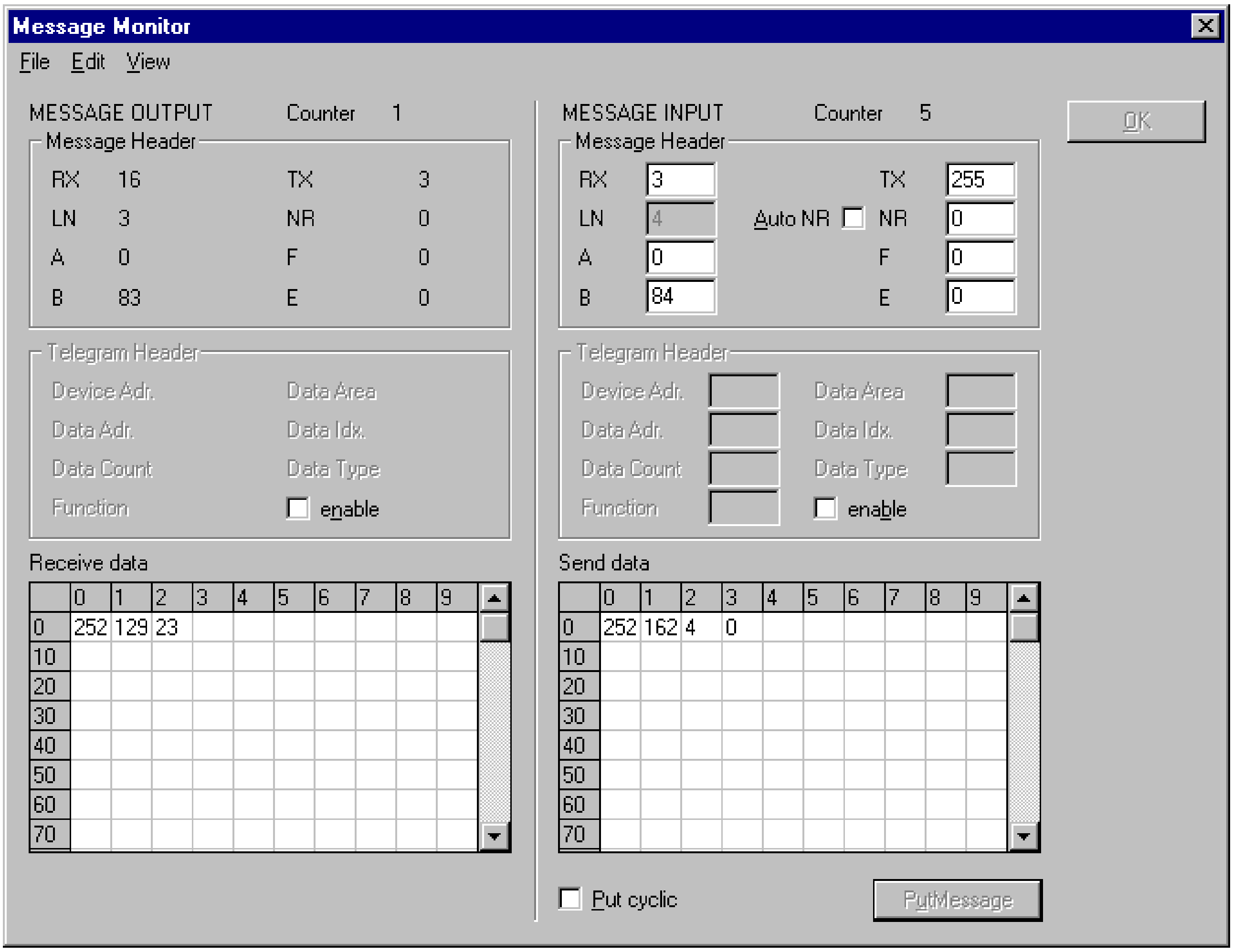
□ Put cyclic

Figure 73: Message Monitor LSS/LMT (4) > Save Configuration

5. Switch in Operating Mode

Message for Sending via CAN (Layer 2)		
Message Header		
Rx = 3 (fixed)	Tx = 255	
Ln = (is calculated)	Nr = 0...255	
A = 0	F = 0	
B = 84	E = 0	
Send Data	**Meaning for CAN**	**Range of value**
	CAN Send ID Part 1	252
	CAN Send ID Part 2	161
	Send data 1: Save Configuration	23

Table 45: Message Monitor LSS/LMT (5) > Switch in Operating Mode



Message Monitor

MESSAGE OUTPUT Counter 1

Message Header

RX	16	TX	3
LN	3	NR	0
A	0	F	0
B	83	E	0

Receive data

	0	1	2	3	4	5	6	7	8	9
0	252	129	23							
10										
20										
30										
40										
50										
60										
70										

MESSAGE INPUT Counter 5

Message Header

RX	3	TX	255
LN	4	Auto NR □ NR	0
A	0	F	0
B	84	E	0

Send data

	0	1	2	3	4	5	6	7	8	9
0	252	162	4	0						
10										
20										
30										
40										
50										
60										
70										

□ Put cyclic

Figure 74: Message Monitor LSS/LMT (5) > Switch in Operating Mode

6.6.2 Message Monitor for Sending or Receiving Transparent CAN Telegrams

Sending and receiving of CAN telegrams (Layer 2) is possible on basis of Messages.

6.6.2.1 Message Monitor for Sending CAN Telegrams (transparent)

In the following the Message Monitor for sending CAN telegrams to the Hilscher Master and/or Hilscher Node is described.

For sending you have to type in the following in the Message Monitor:

Message for Sending via CAN (Layer 2)		
Message Header		
Rx = 3 (fixed)	Tx = 255	
Ln = (is calculated)	Nr = 0...255	
A = 0	F = 0	
B = 84	E = 0	
Send Data	**Meaning for CAN**	**Range of value**
	CAN Send ID Part 1	0...255
	CAN Send ID Part 2	0...255
	Send data 1, if available	0...255
	Send data 2, if available	0...255
	Send data 3, if available	0...255
	Send data 4, if available	0...255
	Send data 5, if available	0...255
	Send data 6, if available	0...255
	Send data 7, if available	0...255
	Send data 8, if available	0...255

Table 46: Message Monitor for Sending CAN Telegrams (transparent)

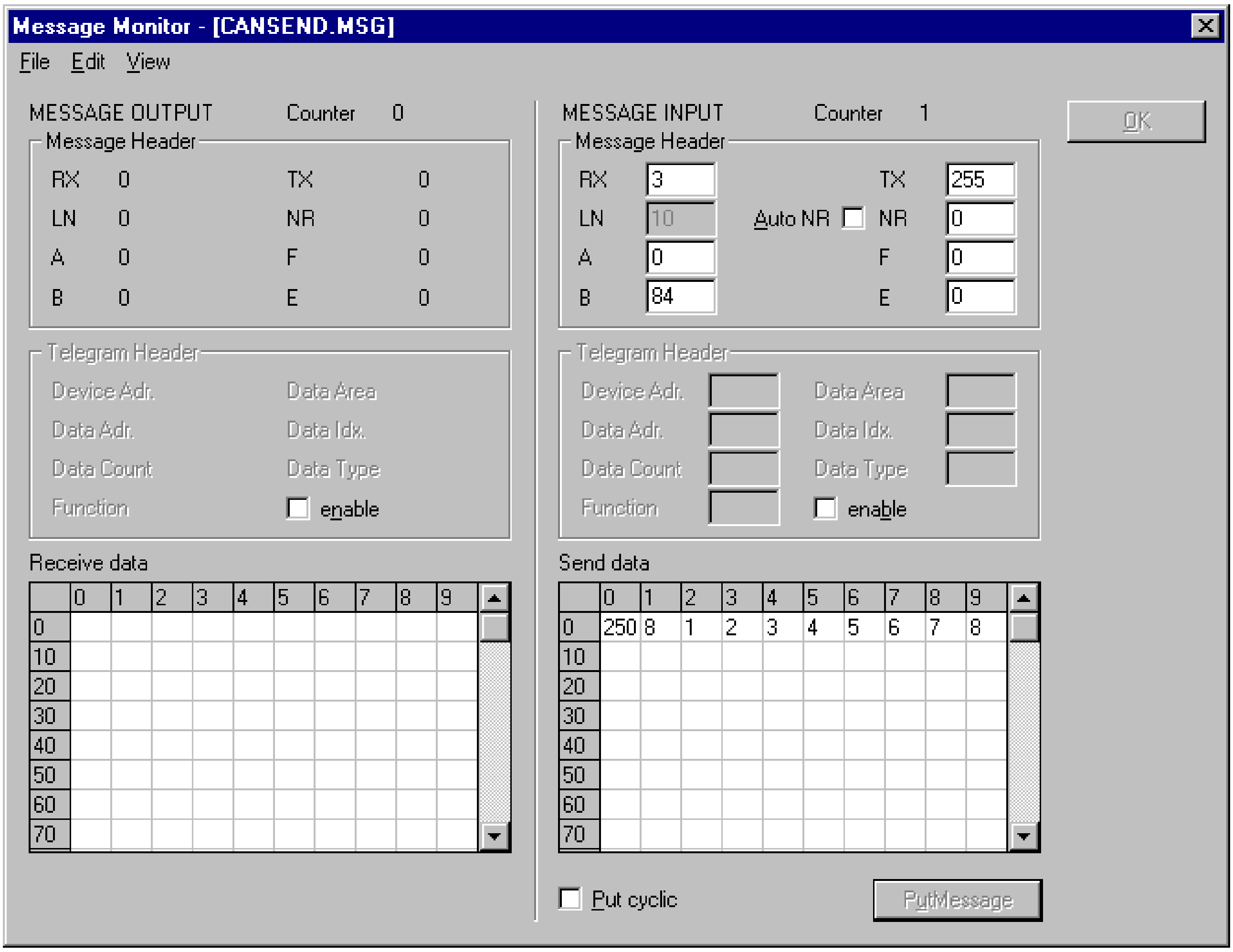
Note: If the Hilscher device is used simultaneously as CANopen device, then Identifier are already used. The user is responsible that it comes to no conflicts here.

The CAN Send ID consists of two Bytes and is formed as follows:

The CAN ID (in range of value 0 to 2047) is multiplicated with 32 and the data length (in range of value 0 to 8) is added up. CAN Send ID Part 1 is then the byte with high order and CAN Send ID Part 2 then is the byte of low order.

Example: If the CAN telegram with CAN ID 2000 with 8 byte user data should be sent, the following results: 2000 x 32 + 8 = 64008 and/or FA08H. Then the CAN Send IP Part 1 is 250 and/or FAH and CAN Send IP Part 2 is 8 and/or 08H.

The following picture shows the Sending of the CAN ID 2000 with 8 bytes user data. The user data here are 1, 2, 3, 4, 5, 6, 7 and 8.



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 □ enable

Receive data

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

MESSAGE INPUT Counter 1

Message Header

RX	3		TX	255
LN	10	Auto NR □	NR	0
A	0		F	0
B	84		E	0

Telegram Header

Device Adr. Data Area
Data Adr. Data Idx.
Data Count Data Type
Function □ enable

Send data

	0	1	2	3	4	5	6	7	8	9
0	250	8	1	2	3	4	5	6	7	8
10										
20										
30										
40										
50										
60										
70										

□ Put cyclic

Figure 75: Message Monitor for Sending CAN telegrams (transparent)

Note: The sending of the telegram is not confirmed and the error number 2025 appears.

6.6.2.2 Message Monitor for Receiving CAN Telegrams (transparent)

For the receiving of CAN telegrams it has to be informed, which CAN Identifier are permissible for receiving. For this the receive filter is set, to inform, which CAN identifier are passed through.

In the following the Message Monitor for setting the receive filter at the Hilscher Master and/or Hilscher Node is described.

For setting a receive filter you have to type in the following in the Message Monitor:

Message for Setting the Receive Filter CAN (Layer 2)		
Message Header		
Rx = 3 (fixed)	Tx = 255	
Ln = (is calculated)	Nr = 0...255	
A = 0	F = 0	
B = 82	E = 0	
Send Data	**Meaning for CAN**	**Range of value**
	CAN Receive ID Part 1	0...255
	CAN Receive ID Part 2	0...7

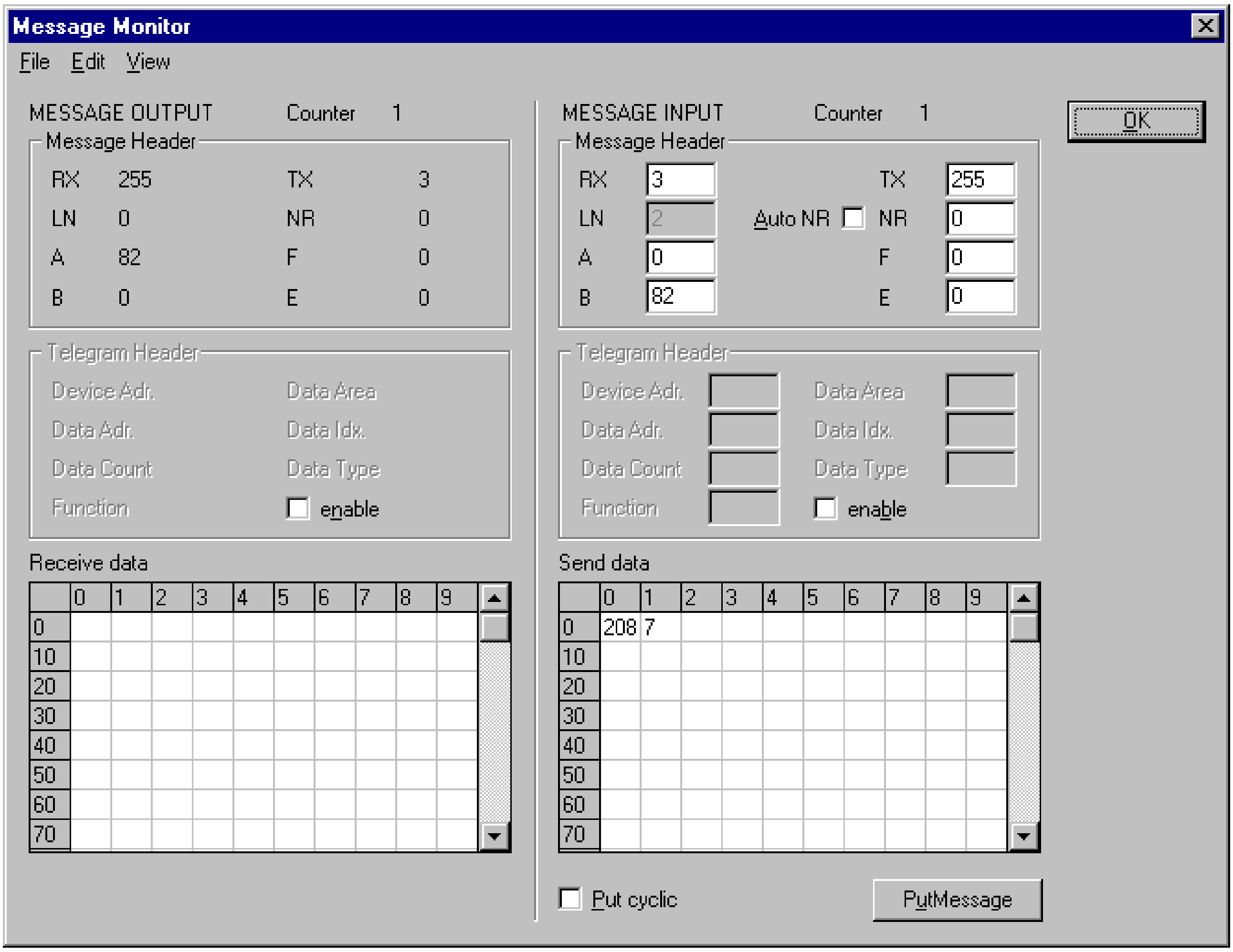
Table 47: Message Monitor for Setting the Receive Filter

The CAN Receive ID consists of two bytes and is formed as follows:

The CAN ID (in range of value 0 to 2047) is segmented in a low order byte and a high order byte. Then the CAN Receive IP Part 1 is the low order byte and the CAN Receive ID Part 2 the high order byte.

Example: If the CAN telegram with CAN ID 2000 should be received, it results this: 2000 and/or 07D0H. Then CAN Receive ID Part 1 is equal to 208 and/or D0H and CAN Receive ID Part 2 is then 7 and/or 07H.

The following picture shows the setting of the receive filter for CAN ID 2000.



Message Monitor

MESSAGE OUTPUT Counter 1

Message Header

RX	255	TX	3
LN	0	NR	0
A	82	F	0
B	0	E	0

Telegram Header

Device Adr.	Data Area
Data Adr.	Data Idx.
Data Count	Data Type
Function	□ enable

Receive data

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

MESSAGE INPUT Counter 1

Message Header

RX	3		TX	255
LN	2	Auto NR □	NR	0
A	0		F	0
B	82		E	0

Telegram Header

Device Adr.		Data Area	
Data Adr.		Data Idx.	
Data Count		Data Type	
Function		□ enable	

Send data

	0	1	2	3	4	5	6	7	8	9
0	208	7								
10										
20										
30										
40										
50										
60										
70										

□ Put cyclic

Figure 76: Message Monitor for Setting the Receive Filter

In the following the Message Monitor with the Receive of CAN telegrams at the Hilscher Master and/or Hilscher Node is described .

With the receiving the following appears in the Message Monitor:

Message for Setting the Receive Filter CAN (Layer 2)		
Message Header		
Rx = 16 (fixed)	Tx = 3	
Ln = (is calculated)	Nr = 0...255	
A = 0	F = 0	
B = 83	E = 0	
Send Data	**Meaning for CAN**	**Range of value**
	CAN Receive ID Part 1	0...255
	CAN Receive ID Part 2	0...255
	Receive data 1, if available	0...255
	Receive data 2, if available	0...255
	Receive data 3, if available	0...255
	Receive data 4, if available	0...255
	Receive data 5, if available	0...255
	Receive data 6, if available	0...255
	Receive data 7, if available	0...255
	Receive data 8, if available	0...255

Table 48: Message Monitor for Receiving of CAN telegrams (transparent)

Note: This is only possible via the dual-port memory (CIF Device Driver). It is not possible via a serial connection (CIF Serial Driver).

The CAN Receive ID consists of two bytes and contains the CAN telegram ID and the data length. It is evaluated like follows:

CAN Receive ID Part 1 is the byte with high order of receipt ID part of 2 is the low order byte.

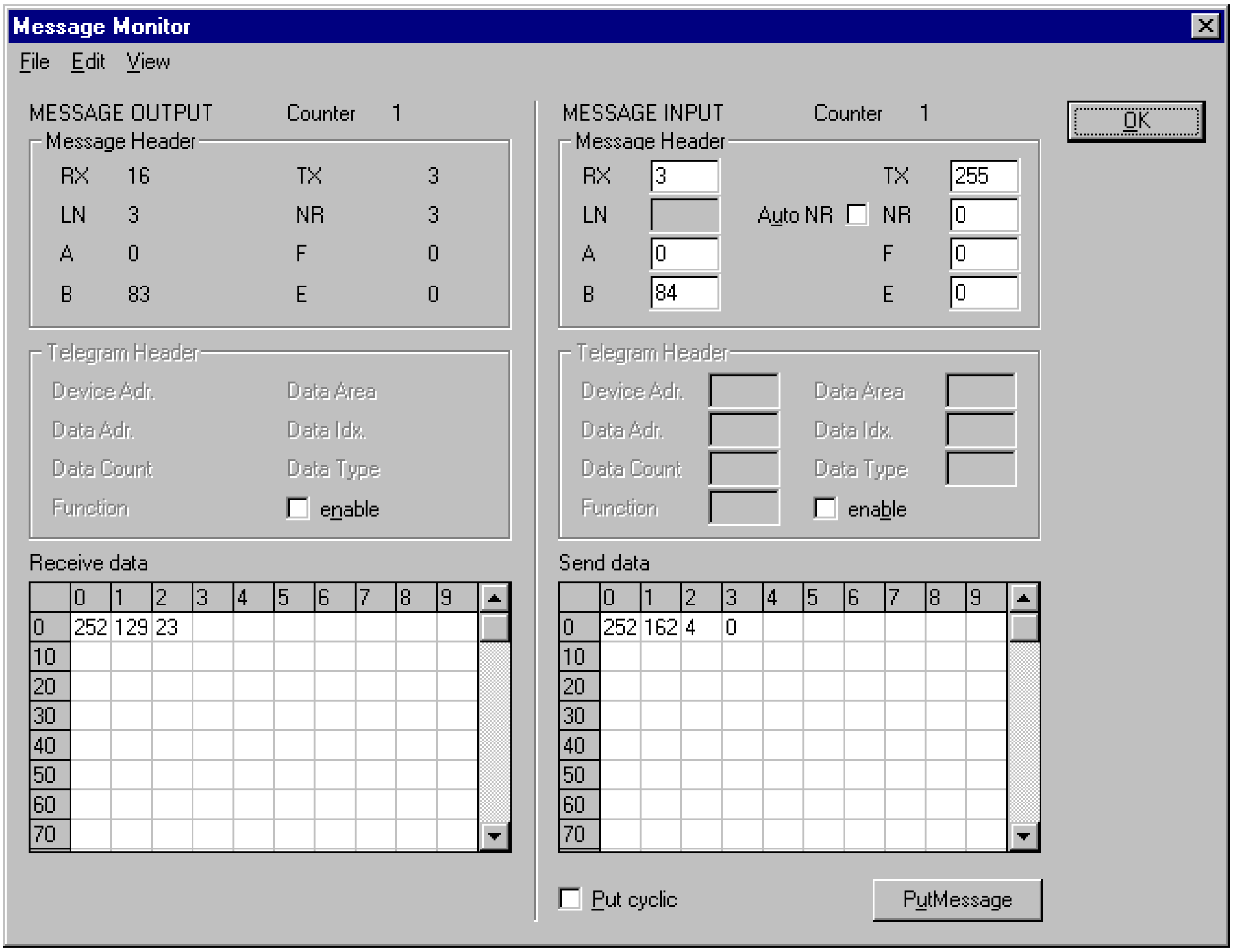
The result is (CAN Receive ID part 1) * 256 + (CAN Receive ID part 2). This result divided by 32 is the CAN telegram identifier.

The length is included in the 4 below bits of the CAN Receive ID part 2.

Example: CAN Receive ID part 1 is received with 250 respectively FAH and CAN Receive ID part 2 is received with 8 respectively with 08H. Then results: 250 * 256 + 8 = 64008. 64008 / 32 = 2000,25. Hence the telegram identifier is 2000.

The CAN Receive ID part 2 is 8 respectively 08H. With this the 4 bellower bit have the value 8. The CAN Receive Telegram includes 8 byte user data.

The following figure shows the receiving of the CNA ID 2000 with 8 bytes user data. The user data here are 1, 2, 3, 4, 5, 6, 7 and 8.



MESSAGE OUTPUT Counter 1

Message Header

RX	16	TX	3
LN	3	NR	3
A	0	F	0
B	83	E	0

Telegram Header

Device Adr. Data Area
Data Adr. Data Idx.
Data Count Data Type
Function □ enable

Receive data

	0	1	2	3	4	5	6	7	8	9
0	252	129	23							
10										
20										
30										
40										
50										
60										
70										

MESSAGE INPUT Counter 1

Message Header

RX	3	TX	255
LN		Auto NR □ NR	0
A	0	F	0
B	84	E	0

Telegram Header

Device Adr. Data Area
Data Adr. Data Idx.
Data Count Data Type
Function □ enable

Send data

	0	1	2	3	4	5	6	7	8	9
0	252	162	4	0						
10										
20										
30										
40										
50										
60										
70										

□ Put cyclic

Figure 77: Message Monitor for Receiving CAN Telegrams (transparent)

7 File, Print, Edit, Export 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

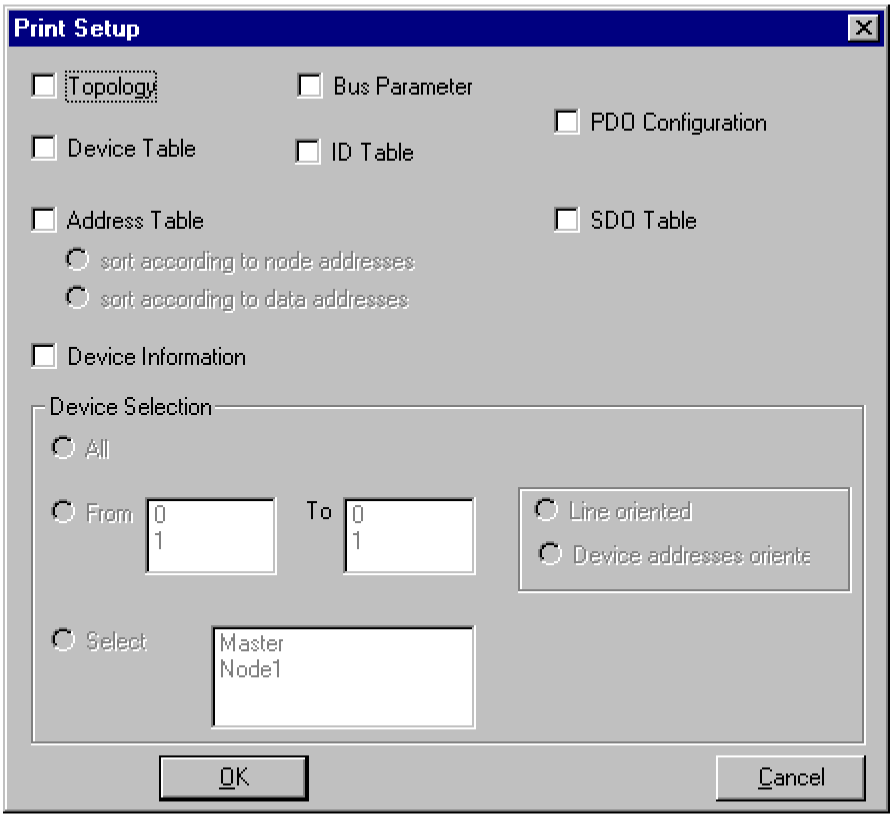
When the file name is known, the configuration can be saved under the **File > Save** menu, otherwise the **File > Save As** menu must be selected.

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 > Print Setup** menu, the configuration can be printed out under the **File > Print** menu. For a page view, select the **File > Print Preview** menu.



Print Setup

□ Topology
□ Bus Parameter
□ PDO Configuration
□ Device Table
□ ID Table
□ Address Table
□ SDO Table
○ sort according to node addresses
○ sort according to data addresses
□ Device Information

Device Selection

○ All
○ From 0 1 To 0 1
○ Line oriented
○ Device addresses oriente
○ Select Master Node1

Figure 78: File > Print

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

Topology prints the topology of the bus system.

Bus Parameter prints the bus parameters of the bus system.

Address table prints the address table of the Master.

Device table prints the device table.

ID Table prints the ID table.

PDO Configuration prints the PDO configuration.

SDO Table prints the SDO 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 (*.CO Microsoft Access Format) in a DBM file (Hilscher binary format). This DBM file can be retrieved in the DOS Compro program. The configuration is stored in the Project directory in the path of the SyCon Installation with the extension *.dpm.

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 parameters:

Parameter	**Meaning**
Stationaddress	The Station address is the unique device address of the Slave on the bus.
RecordType	The RecordType defines the version of the following structure and is always 2.
IdentNumber	This number is the unique device number of the Slave.
VendorNumber	The VendorNumber is the clear number of the vendor (if available).
VendorName	Here the name of the vendor is shown (max. 32 characters).
Device	Name of the device (max. 32 characters).
Description	This is the description of the device, which is set by the user (max. 32 characters).
MasterAddress	This is the number of the Master Address, where the devices are related to.
Settings	Contains information about the addressing mode and the storage format of the process data (words, double words and floats) see section *Description of the Parameter Settings*.
Reserved	reserved
ModulCount	Number of the modules of the device. For each module the parameters data type, data size, data position and offset address are given. It can be follow max 60 modules. The parameters for module 1 are marked with ..._0 and of the module 60 are marked with ..._59.
DataSize_0	Number of bytes, which were used by the module.
DataType_0	The DataType, which is used in the configuration. The code for this you find below this table in section *Description of the Parameter DataType.*
DataPosition_0	The byte DataPosition, which is used in the configuration. The code for this you find below this table in section *Description of the Parameter DataPosition.*
Address_0	Offset Address in the Dual-port memory
...	...
DataSize_59	if used, see at the top
DataType_59	if used, see at the top
DataPosition_59	if used, see at the top
Address_59	if used, see at the top

Table 49: CSV Export - Meaning of the values

7.3.2.1 Description of the Parameter Settings

D7	D6	D5	D4	D3	D2	D1	D0
Reserved Area						Format	Address Mode
							0 byte Address 1 word Address
						1 little Endian (LSB/MSB) 0 big Endian (MSB/LSB)	
reserved							

Table 50: CSV-Export - Description of the Byte Settings

7.3.2.2 Description of the Parameter DataType

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

Table 51: CSV Export > DataType Code

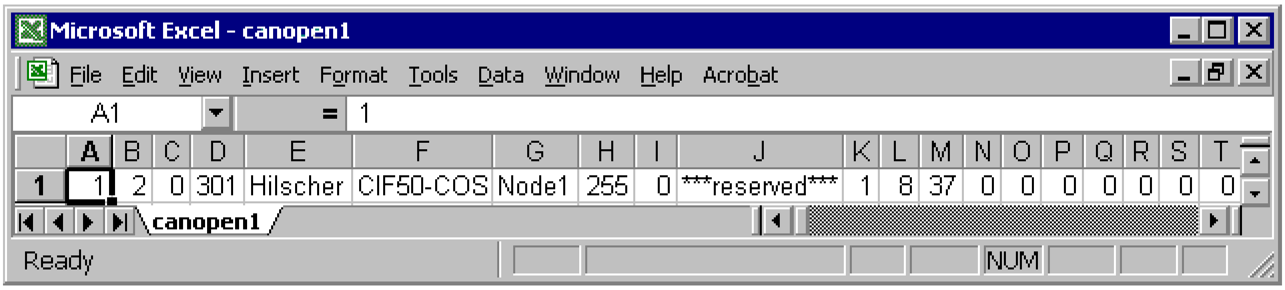
7.3.2.3 Description of the Parameter DataPosition

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

Table 52: CSV Export > DataPosition Code

7.3.2.4 Example of a CSV file

Example of a CSV file which was exported in Excel:


Microsoft Excel - canopen1
File Edit View Insert Format Tools Data Window Help Acrobat
A1
=
1
canopen1
Ready
NUM

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	1	2	0	301	Hilscher	CIF50-COS	Node1	255	0	***reserved***	1	8	37	0	0	0	0	0	0	0

Figure 79: Example of a CSV File in Excel

Cell	Parameter	Value	Meaning
A1	StationAddress	1	Station address of the CANopen Node.
B1	RecordType	2	The RecordType is always 2.
C1	IdentNumber	0	IdentNumber of the Node.
D1	VendorNumber	301	The vendor number is 301.
E1	VendorName	Hilscher	Vendor name of the device.
F1	Device	CIF 50-COS	Description of the device.
G1	Description	Node1	Description of the device which is also shown in SyCon as the name of the device.
H1	MasterAddress	255	Address of the related Master.
I1	Settings	0	The addressing mode (byte- or word addressing) and the data format of the process data are shown. The description you see in section *Description of the Parameter Settings*.
J1	reserved	reserved	reserved
K1	ModulCount	1	Number of the modules of the device. For each module the information with datatype, data size, data position and the offset address follow. The information for module 1 you find in the cells L1, M1, N1, O1 and for module 2 in the cells P1, Q1, R1, S1 and so on.
L1	DataSize	8	The size of the module is 8 bytes.
M1	DataType	37	Input; Datatype unsigned Integer 8
N1	DataPosition	0	Output; Datatype unsigned Integer 8
O1	Offset address	0	The Offset address is 0.
P1...IQ1	DataSize	0	The modules 2 till 59 are not used for this device and so a 0 is shown.

Table 53: Example of a CSV File in Excel

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

7.4 Edit

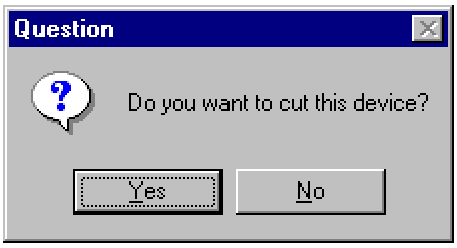
7.4.1 Cut, Copy and Paste

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

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

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

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

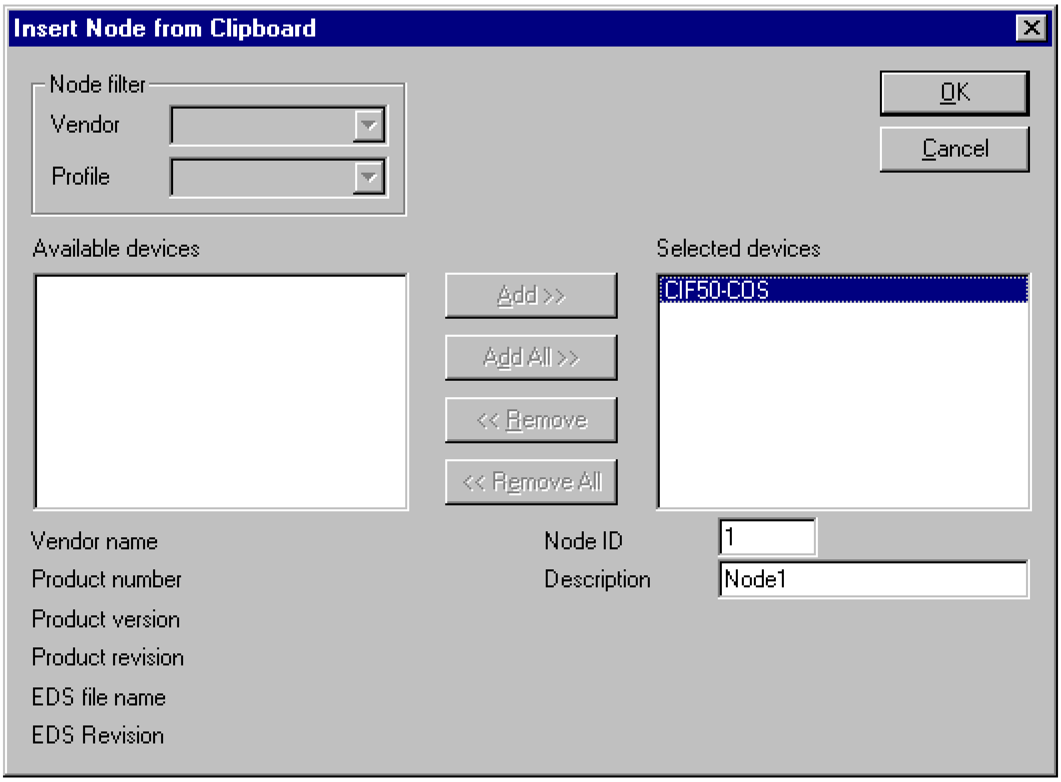


Do you want to cut this device?

Figure 80: Security question cut device

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

With the menu **Edit > Paste** and clicking at the position where the device should be inserted, a window opens where the cut/copied device can be selected.



Selected devices: CIF50-COS

Node ID: 1

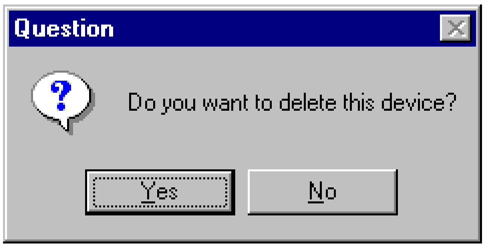
Description: Node1

Figure 81: Edit > Paste cut/copied device

When you click on the **OK** button the device will be inserted in the configuration.

7.4.2 Delete

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



Question

Do you want to delete this device?

Figure 82: Security question delete device

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

7.4.3 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 *39*. If you want to replace a Slave device look in section *Replace Node* at page *66*.

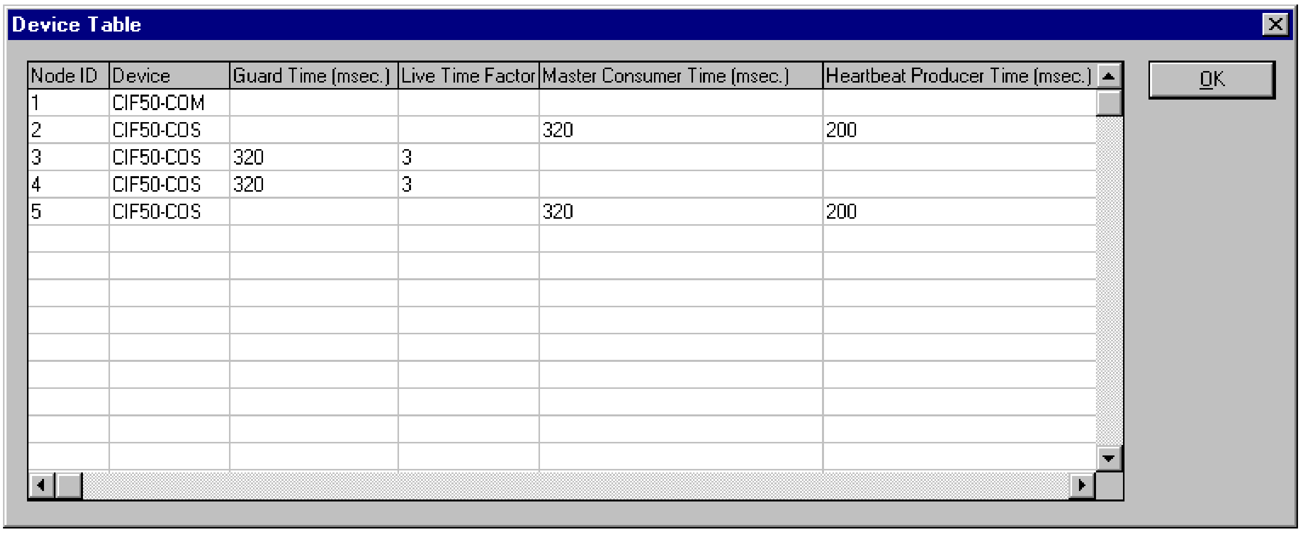
7.5 View of the Configuration

The configuration can be displayed in a table. The following tables are available:

- Device Table
- Address Table
- ID Table
- SDO Table

7.5.1 Device Table

The list of all added devices is displayed with the menu item **View > Device Table**. Apart from the Node address the name of the device displayed with the pertinent alterable description. Look at section *Node Configuration* at page *42*.



Device Table

Node ID	Device	Guard Time (msec.)	Live Time Factor	Master Consumer Time (msec.)	Heartbeat Producer Time (msec.)
1	CIF50-COM				
2	CIF50-COS			320	200
3	CIF50-COS	320	3		
4	CIF50-COS	320	3		
5	CIF50-COS			320	200

Figure 83: View > Device Table

- **Node ID**

  Shows the Node ID of the device.
- **Device**

  Name of the device.
- **Guard Time (msec.)**

  Information about the Guard Time for the Node Guarding Protocol in ms.
- **Live Time Factor**

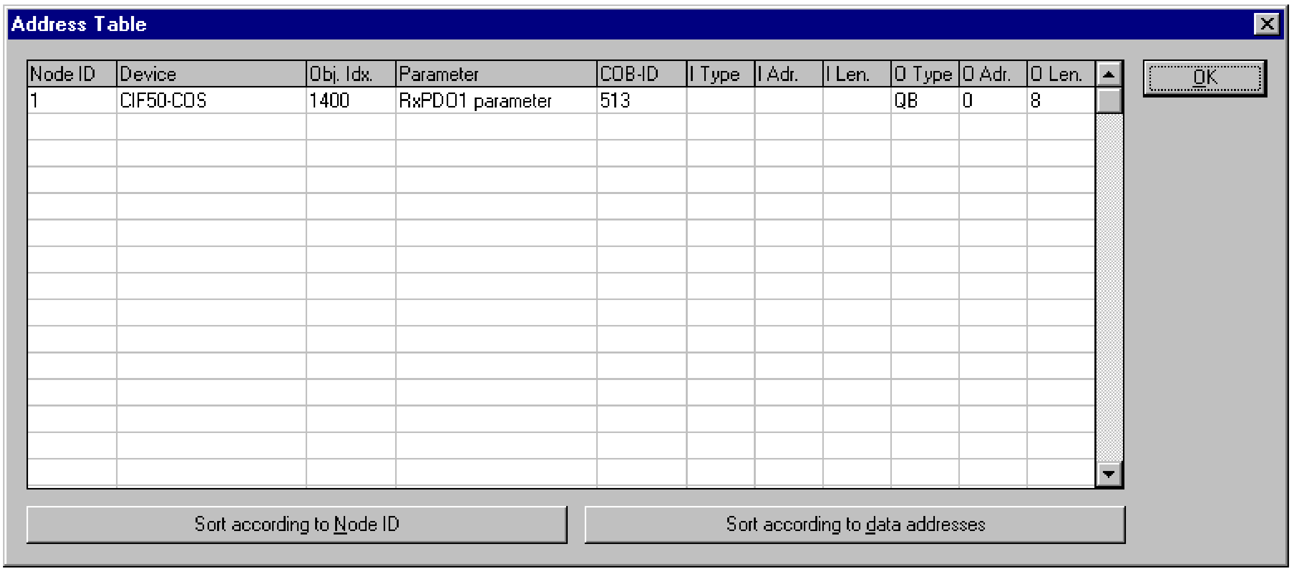
  Displays the Life Time Factor for the Life Guarding.
- **Master Consumer Time (msec.)**

  Displays the Master Consumer Time for the Heartbeat Protocol in ms.
- **Heartbeat Producer Time (msec.)**

  Displays the Heartbeat Producer Time for the Heartbeat Protocol in ms.

7.5.2 Address Table

With the menu item **View > Address Table** you get an overview of all configured PDOs and the booked start addresses in the process image including their length.



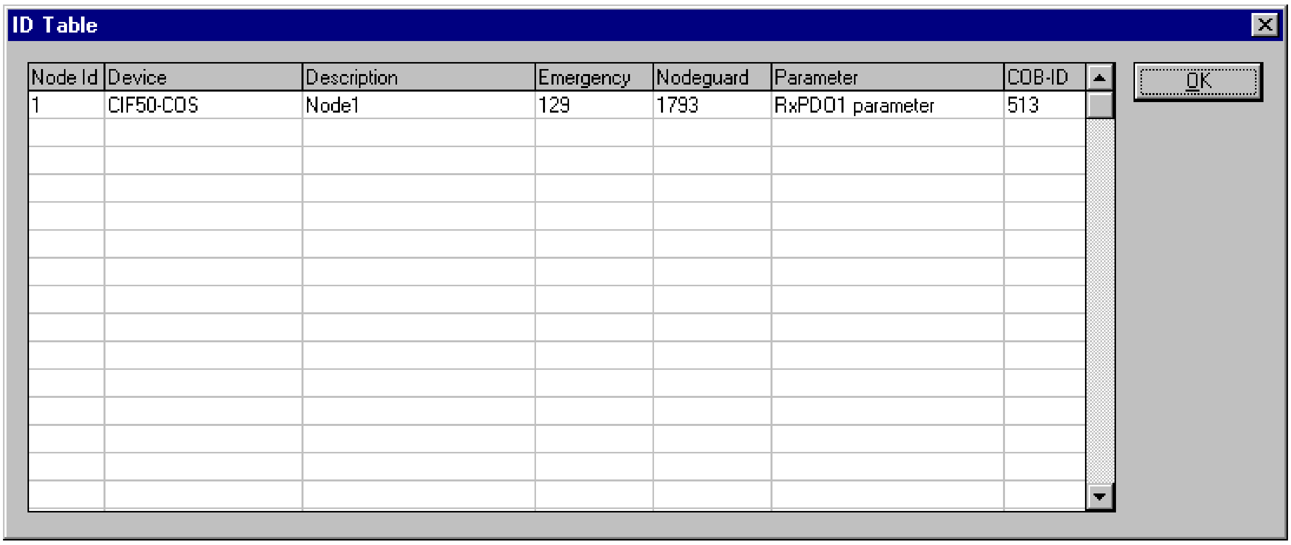
Node ID	Device	Obj. Idx.	Parameter	COB-ID	I Type	I Adr.	I Len.	O Type	O Adr.	O Len.
1	CIF50-COS	1400	RxPDO1 parameter	513				QB	0	8

Figure 84: View > Address Table

It is possible to sort the addresses according to Station Addresses or according to Data Addresses.

7.5.3 ID Table

With the menu item **View > ID Table** is sorted listed for each Node, which message numbers in the CAN network are occupied by the respective Nodes. This are the Emergency ID, Nodeguard ID and the IDs of the PDOs.

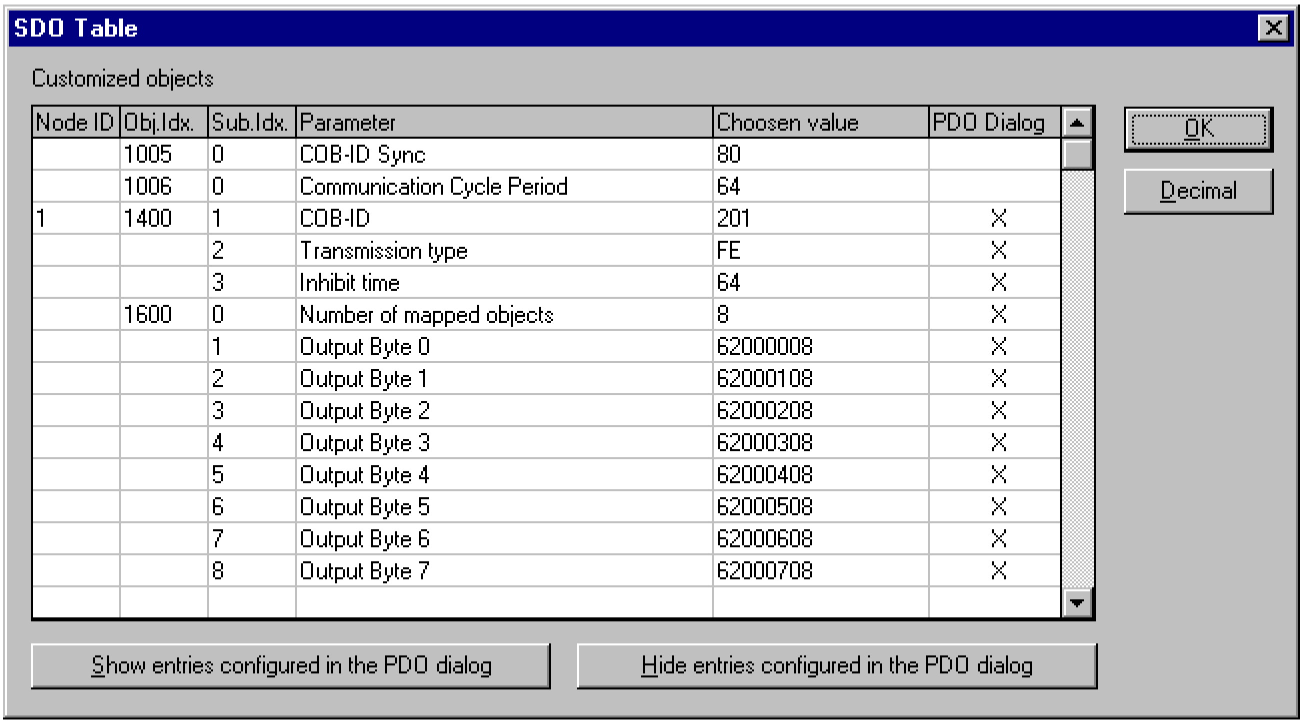


Node Id	Device	Description	Emergency	Nodeguard	Parameter	COB-ID
1	CIF50-COS	Node1	129	1793	RxPDO1 parameter	513

Figure 85: View > ID Table

7.5.4 SDO Table

With the menu item **View > SDO Table** you get an overview of the transmitted objects during the Node BootUp phase for each Node. Apart from the Node Address for each entry the Object- and Sub index is displayed with the pertinent value. Thereby if a line contains a cross in the column PDO Dialog, then the entry was created automatically when inserting a PDOs by SyCon and can be changed in section *Node Configuration* (see at page *42*). If an entry does not contain a cross in the column PDO Dialog, then the appropriate object is manually created in the *Object Configuration* (see at page *64*) and can be changed there. Exceptions here form the entries COB-ID SYNC and Communication Cycle Period, which can be changed in the dialog *Bus Parameter* (see at page *77*). It is possible to hid or to display the configured objects of the PDO Dialog. The representation method of the object values can be selected between decimal and hexadecimal.



SDO Table
Customized objects
OK
Decimal
Show entries configured in the PDO dialog
Hide entries configured in the PDO dialog

Node ID	Obj.Idx.	Sub.Idx.	Parameter	Choosen value	PDO Dialog
	1005	0	COB-ID Sync	80	
	1006	0	Communication Cycle Period	64	
1	1400	1	COB-ID	201	X
		2	Transmission type	FE	X
		3	Inhibit time	64	X
	1600	0	Number of mapped objects	8	X
		1	Output Byte 0	62000008	X
		2	Output Byte 1	62000108	X
		3	Output Byte 2	62000208	X
		4	Output Byte 3	62000308	X
		5	Output Byte 4	62000408	X
		6	Output Byte 5	62000508	X
		7	Output Byte 6	62000608	X
		8	Output Byte 7	62000708	X

Figure 86: View > SDO Table

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 (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 / PKV50 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 The communication board is not initialized by the driver. No or wrong configuration found for the given board, check the driver configuration. Driver function used without calling DevOpenDriver() first.
-2	Driver: Error in internal ‘Init state‘
-3	Driver: Error in internal ‘Read state‘
-4	Driver: Command on this channel is active
-5	Driver: Unknown parameter in function occurred
-6	Driver: Version is incompatible The device driver version does not correspond to the driver DLL version. From version V1.200 the internal command structure between DLL and driver has changed. Make sure to use the same version of the device driver and the driver DLL.
-10	Device: Dual port memory RAM not accessible (board not found) Dual-ported RAM (DPM) not accessible / no hardware found. This error occurs, when the driver is not able to read or write to the Dual-port memory. Check the BIOS setting of the PC Memory address conflict with other PC components. Try another memory address, check the driver configuration for this board, check the jumper setting of the board.
-11	Device: Not ready (RDY flag=Ready flag failed) Board is not ready. This could be a hardware malfunction or another program writes inadmissible to the dual-port memory.
-12	Device: Not running (RUN flag=Running flag failed) The board is ready but not all tasks are running, because of an initialization error. No data base is loaded into the device or a wrong parameter can causes that a task can't initialize.
-13	Device: Watch dog test failed
-14	Device: Signals wrong Operating System version No license code found on the communication board. Device has no license for the used operating system or customer software. No firmware or no data base to the device is loaded.

Table 54: CIF Device Driver Error Numbers (-1..-14)

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

Table 55: CIF Device Driver Error Numbers (-15..-19)

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

Table 56: CIF Device Driver Error Numbers (-20..-27)

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

Table 57: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 58: 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 NULL
-46	User: Buffer too short
-47	User: Setup error

Table 59: CIF Serial Driver Error Numbers (-20..-47)

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

Table 60: 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 61: 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 62: 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 existent
22	Number for message wrong during download
23	Received number of data during download does not match to that in the command message
24	Sequence identifier wrong during download
25	Checksum after download and checksum in command message do not match
26	Write/Read access of data base segment
27	Download/Upload or erase of configured data base type is not allowed
28	The state of the data base segment indicated an error. Upload not possible
29	The access to the data base segment needs the bootstraploader. The bootstraploader is not present
30	Trace buffer overflow
31	Entry into trace buffer too long
37	No or wrong license. The OEM license of the System Configurator allows only communication to devices that have the same license inside
38	The data base created by the System Configurator and the data base expected by the firmware is not compatible
39	DBM module missing

Table 63: 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 64: 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 65: Database Access Error Numbers (100..130)

9.6 SyCon Error Number (235)

Error Number	Description
235	Project file with the same name already opened. Two project files with the same name can't be opended at the same time.

Table 66: SyCon Error Number (235)

9.7 Online Data Manager Error Numbers

9.7.1 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 67: Online Data Manager Error numbers (1000..1018)

9.7.2 Message Handler Error Numbers (2010 .. 2027)

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

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

Table 68: Error Numbers of the Message Handler of the Online Data Manager (2010..2027)

9.7.3 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 69: Error Numbers of the Driver Functions of the Online Data Manager (2501..2512)

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

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

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

Table 70: Sub function Error Numbers of the Driver Functions of the Online Data Manager (8001..8035)

9.8 Data Base Functions Error Numbers (4000 .. 4098)

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

Error Number	Description
4000	File does not exist
4001	Success in comprimizing
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 71: Error numbers of converting functions (4000..4029)

Error Number	Description
4030	Path too long
4031	Directory changed
4032	Directory created
4034	Length of converting stream is 0
4035	Non equal data set found
4036	Non equal data set found
4037	Non equal data set found
4038	Data set has length 0
4039	The function DbmInit has assigned a Zero pointer during RCS initialization
4040	Printer not ready
4041	The data base is used from 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 72: 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 73: Error numbers of converting functions (4061..4075)

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

Table 74: Error numbers of converting functions (4082..4199)

9.9 Converting Functions Error Numbers (5001 .. 5008)

The following table lists the error numbers of converting functions.

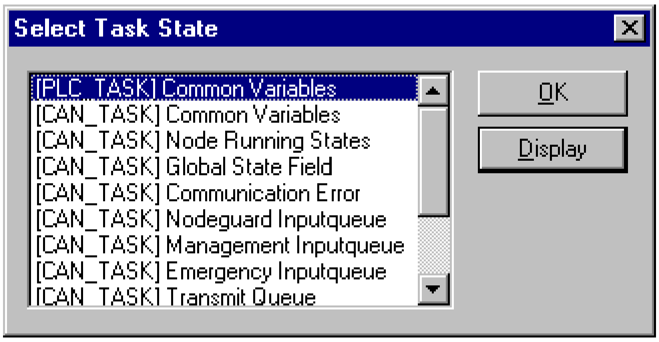
Error Number	Description
5000	Function PackLongToByteShort: Not enough space in pvD (Number of elements greater than reserved memory)
5001	Function PackLongToByteShort: Not enough space in pvD. Detected during converting of pvS
5002	Function PackLongToByteShort: Not enough space in pvD
5003	Function 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 75: Error Numbers of data base functions (5000 .. 5012)

10 Appendix

10.1 Extended Device Diagnostic Master

The menu item **Online > Extended Device Diagnostic** helps to find possible network and configuration faults while trying to get the network fully operative, when the normal debugger does not rudicate any helpful information any more to get the fault location. This menu activates a list of available structures. The listed structures can be displayed to show the values. The structures will be resetted after power on or after a cold or warmstart command.



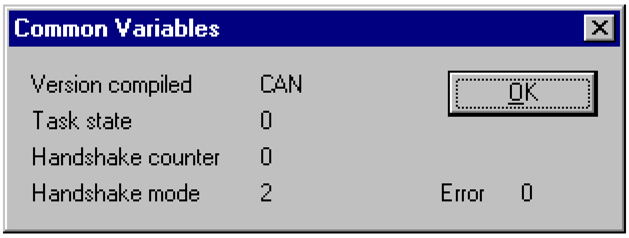
Select Task State

[PLC_TASK] Common Variables
[CAN_TASK] Common Variables
[CAN_TASK] Node Running States
[CAN_TASK] Global State Field
[CAN_TASK] Communication Error
[CAN_TASK] Nodeguard Inputqueue
[CAN_TASK] Management Inputqueue
[CAN_TASK] Emergency Inputqueue
[CAN_TASK] Transmit Queue

Figure 87: Online > Extended Device Diagnostic

This points contain online counters, values, parameters and statuses. Several task states are available

10.1.1 PLC_TASK Common Variables



Common Variables

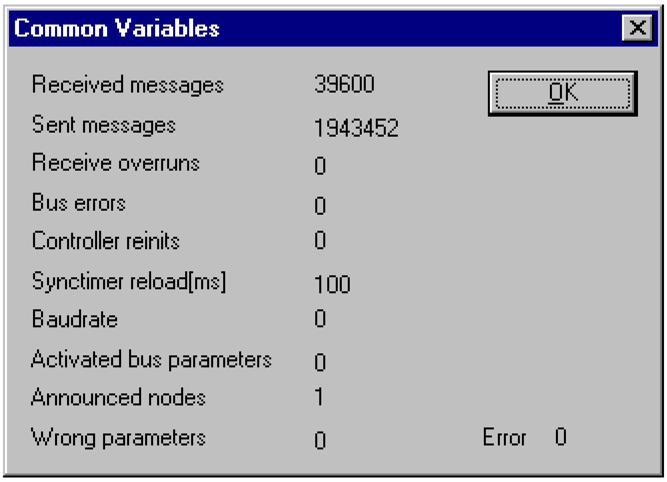
Version compiled CAN
Task state 0
Handshake counter 0
Handshake mode 2
Error 0

Figure 88: PLC_TASK Common Variables

Variable	Meaning
Version compiled	indicates the hardware version the software is compiled for
Task state	is always filled up with value 0
Handshake counter	number of process data handshakes ever done with the application
Handshake mode	represent the actual process data handshake mode the card is actual running with. The mode can be switched in the menu **Settings > CANopen Master Settings**

Table 76: PLC_TASK Common Variables

10.1.2 CAN_TASK Common Variables



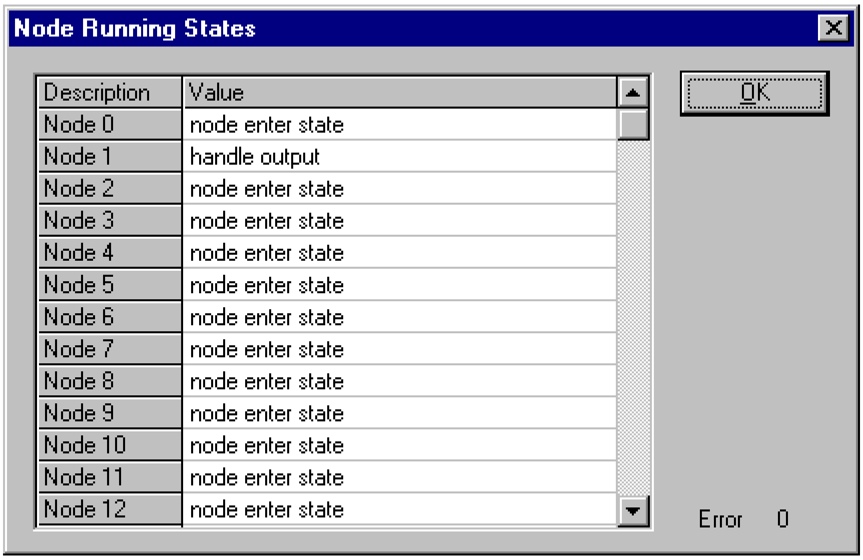
Received messages	39600
Sent messages	1943452
Receive overruns	0
Bus errors	0
Controller reinits	0
Synctimer reload[ms]	100
Baudrate	0
Activated bus parameters	0
Announced nodes	1
Wrong parameters	0
Error	0

Figure 89: CAN_TASK Common Variables

Variable	Meaning
Received messages	Number of received CAN-Messages
Sent messages	Number of sent CAN-Messages
Receive overruns	This counter is incrementing when to much incoming CAN messages overload the master. An incremented counter will always cause lost CAN message data, so it should normally contain the value 0.
Bus errors	Our used CAN controller has two internal error frame counter for detected transmission errors one for receive and one for transmit messages. If one of these error counter oversteps a defined value, the bus error counter is incremented by a value of 1.
Controller reinits	If the internal CAN controller error frame counter overstep a defined limit the controller goes into the bus off state. If this occurs we reinitialize the controller again to be preoperative and increment this counter value. A value unequal 0 is an indication for bad transmission quality, for unsatisfied bus wiring or for low power in the CAN-controller interface driver.
Synctimer reload	This value represent the value that was configured via the menu **Settings > Bus parameter** in SyCon and shows the actual configured and handled value.
Baudrate	This value shows numeric the actual baudrate the master is working with: 0 = 1Mbaud, 1 = 800Kbaud, 2 = 500kBaud, 3 = 250Kbaud, 4 = 125kBaud, 5 = 100Kbaud, 6 = 50kBaud, 7 = 20kBaud, 8 = 10kBaud
Activated bus parameter	Value 0, the master device 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 Node data sets in the download database.
Wrong parameters	This value indicates, if the master has detected any error in a Node data set which was a containment of the actual downloaded data base. For each Node which has a wrong entry in there the counter is incremented by 1.

Table 77: CAN_TASK Common Variables

10.1.3 CAN_TASK Node Running State



Description	Value
Node 0	node enter state
Node 1	handle output
Node 2	node enter state
Node 3	node enter state
Node 4	node enter state
Node 5	node enter state
Node 6	node enter state
Node 7	node enter state
Node 8	node enter state
Node 9	node enter state
Node 10	node enter state
Node 11	node enter state
Node 12	node enter state

Error 0

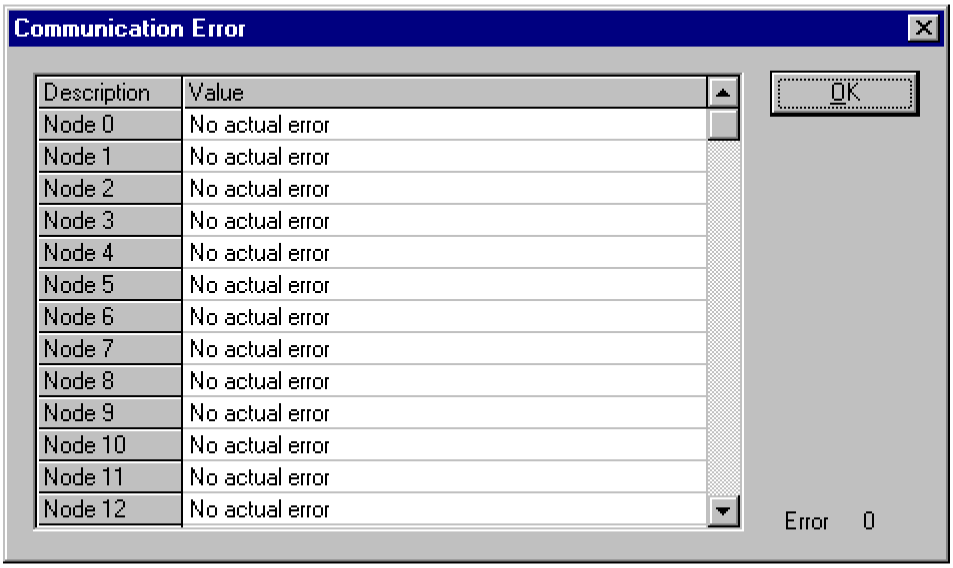
Figure 90: CAN_TASK Node Running State

To handle the Nodes in their different states and requirements the master device has a so-called Node handler running, where each Node has its own actual state. SyCon interpret now the actual state of each Node and print it on the screen in textual form.

10.1.4 CAN_TASK Global State Field

See section *Global State Field* at page *103*.

10.1.5 CAN_TASK Communication Error



Description	Value
Node 0	No actual error
Node 1	No actual error
Node 2	No actual error
Node 3	No actual error
Node 4	No actual error
Node 5	No actual error
Node 6	No actual error
Node 7	No actual error
Node 8	No actual error
Node 9	No actual error
Node 10	No actual error
Node 11	No actual error
Node 12	No actual error

Error 0

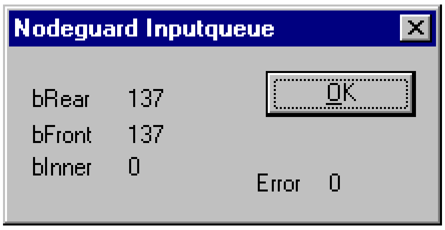
Figure 91: CAN_TASK Communication Error

For each Node the master has an internal online error buffer. SyCon interprets now the actual error condition and print it on the screen in textual form.

10.1.6 Queues

The different incoming CAN specific identifier with their message containment are assigned to different input message queues. A received message is interpreted directly after it was received by the CAN controller and stored into its corresponding queue as well as the messages that shall be sent are stored in a queue while the CAN controller is busy in sending a message. A main loop then interpret these messages and dequeue them or the interrupt handler send the next message. The message queue handler has three parameters and their containments are shown by SyCon. **bInner** is the number of actual stored messages. **bFront** is the pointer where the next message will be store and **bRear** is the pointer where the next message will be dequeue from the queue body area. In a running system the value **bInner** should normally decrease automatically to 0 and **bFront** and **bRear** should be equal.

10.1.6.1 CAN_TASK Nodeguard Inputqueue



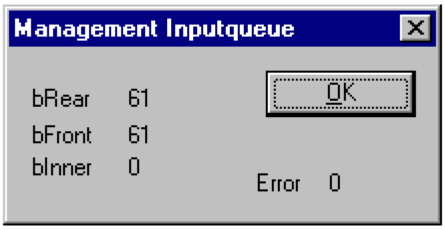
bRear 137
bFront 137
bInner 0
Error 0

Figure 92: CAN_TASK Nodeguard Inputqueue

Variable	Meaning
bRear	Position of the next read access to the puffer of received CAN telegrams
bFront	Position of the next write access
bInner	Number of CAN telegrams which are actual included in the puffer

Table 78: CAN_TASK Nodeguard Inputqueue

10.1.6.2 CAN_TASK Management Inputqueue



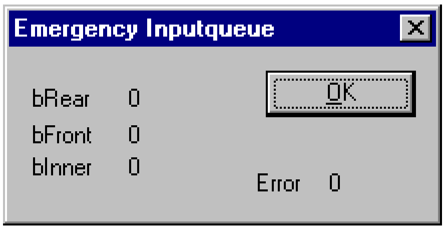
bRear 61
bFront 61
bInner 0
Error 0

Figure 93: CAN_TASK Management Inputqueue

Variable	Meaning
bRear	Position of the next read access to the puffer of received CAN telegrams
bFront	Position of the next write access
bInner	Number of CAN telegrams which are actual included in the puffer

Table 79: CAN_TASK Management Inputqueue

10.1.6.3 CAN_TASK Emergency Inputqueue



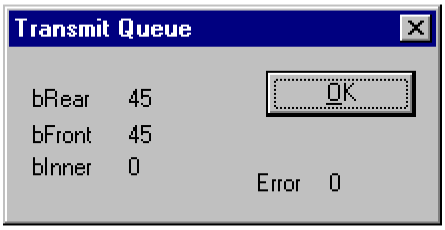
bRear 0
bFront 0
bInner 0
Error 0

Figure 94: CAN_TASK Emergency Inputqueue

Variable	Meaning
bRear	Position of the next read access to the puffer of received CAN telegrams
bFront	Position of the next write access
bInner	Number of CAN telegrams which are actual included in the puffer

Table 80: CAN_TASK Emergency Inputqueue

10.1.6.4 CAN_TASK Transmit Queue



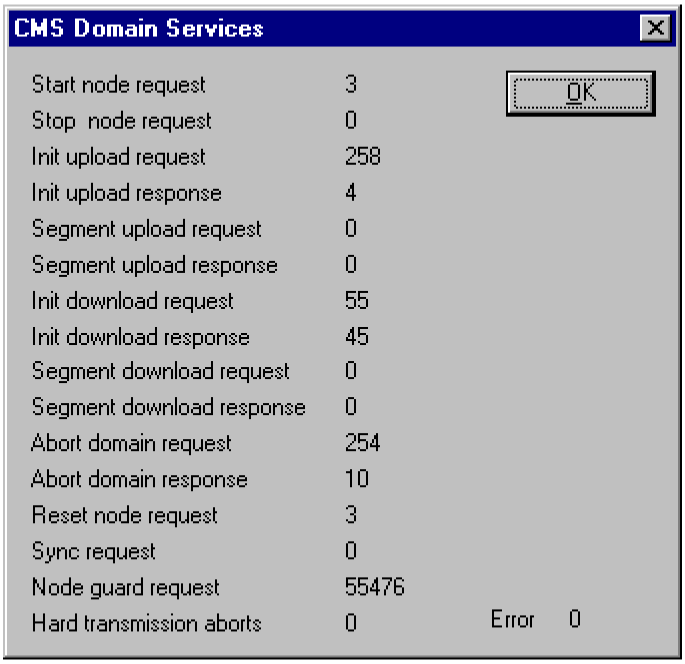
bRear 45
bFront 45
bInner 0
Error 0

Figure 95: CAN_TASK Transmit Queue

Variable	Meaning
bRear	Position of the next read access to the puffer of received CAN telegrams
bFront	Position of the next write access
bInner	Number of CAN telegrams which are actual included in the puffer

Table 81: CAN_TASK Transmit Queue

10.1.7 CAN_TASK CMS Domain Services



CMS Domain Services

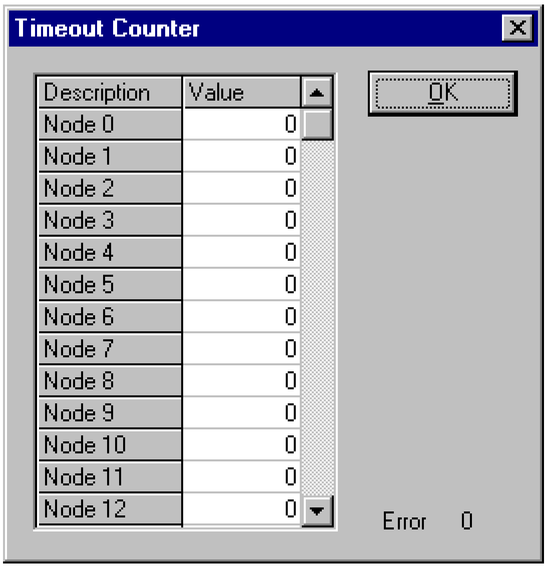
Start node request	3
Stop node request	0
Init upload request	258
Init upload response	4
Segment upload request	0
Segment upload response	0
Init download request	55
Init download response	45
Segment download request	0
Segment download response	0
Abort domain request	254
Abort domain response	10
Reset node request	3
Sync request	0
Node guard request	55476
Hard transmission aborts	0

Error 0

Figure 96: CAN_TASK CMS Domain Services

The CANopen protocol defines different services which are summarized under the name **Domain Services**. All Domain Services that are transmitted and were received are counted in this table and shown online by SyCon. A special value is the **Hard transmission abort** counter. Each CAN message which is inserted into the CAN controller to be sent is supervised by a simultaneously started timer. If the CAN controller cannot sent the message because it don’t find any other CAN controller active in the connected network who is acknowledging him the message, the message can’t be sent and the timer expires. If so the message is thrown away and the next message of the queue is inserted to be sent. So an incrementing **Hard transmission abort** counter is directly an indication for a physical hardware problem in the network. A possible fault that is often made is a wrong configured baud rate for example that causes such an error too.

10.1.8 CAN_TASK Timeout Counter



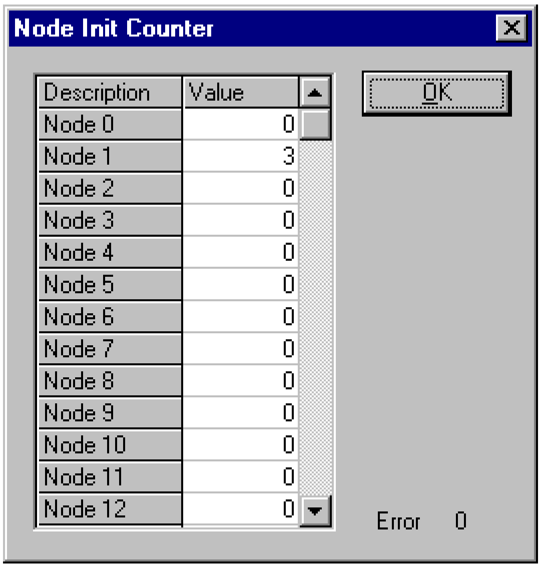
Description	Value
Node 0	0
Node 1	0
Node 2	0
Node 3	0
Node 4	0
Node 5	0
Node 6	0
Node 7	0
Node 8	0
Node 9	0
Node 10	0
Node 11	0
Node 12	0

Error 0

Figure 97: CAN_TASK Timeout Counter

Normally PDOs are transferred in CANopen protocol without sending back any kind of acknowledge message. But if input PDO data is polled by the master with a remote request telegram each addressed Node has to response. If a Node do not response to an outstanding remote request, then the **Timeout Counter** of the corresponding Node is incremented by a value of 1. So if a counter shows a value unequal 0 this can be seen as an indication that the remote request rate is to high for the Node which cannot answer to every request. In such case decrease the Node request poll rate in the Node configuration window.

10.1.9 CAN_TASK Node Init Counter



Node Init Counter

Description	Value
Node 0	0
Node 1	3
Node 2	0
Node 3	0
Node 4	0
Node 5	0
Node 6	0
Node 7	0
Node 8	0
Node 9	0
Node 10	0
Node 11	0
Node 12	0

Error 0

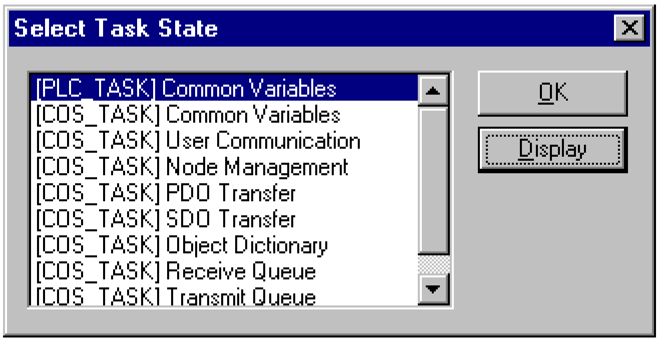
Figure 98: CAN_TASK Node Init Counter

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

10.2 Extended Device Diagnostic Node

The menu item **Online > Extended Device Diagnostic** helps to find possible bus and configuration faults while trying to get the bus fully operative, when the normal debugger does not rudicate any helpful information any more to get the fault localization. This menu activates a list of available structures. The listed structures can be displayed to show the values.

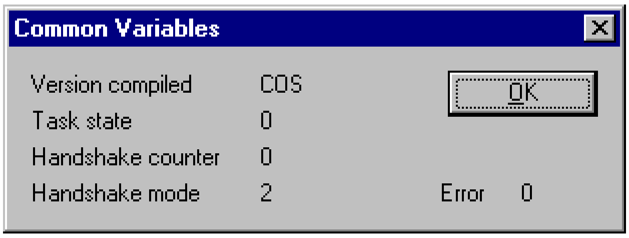
To activate the extended device diagnostic for Hilscher Nodes, click with the right mouse button to the Node and select the menu Select as actual master. Then select the menu **Online > Extended Device Diagnostic**. This menu activates the following list:



[PLC_TASK] Common Variables
[COS_TASK] Common Variables
[COS_TASK] User Communication
[COS_TASK] Node Management
[COS_TASK] PDO Transfer
[COS_TASK] SDO Transfer
[COS_TASK] Object Dictionary
[COS_TASK] Receive Queue
[COS_TASK] Transmit Queue

Figure 99: Extended Device Diagnostic Node

10.2.1 PCL_TASK Common Variables



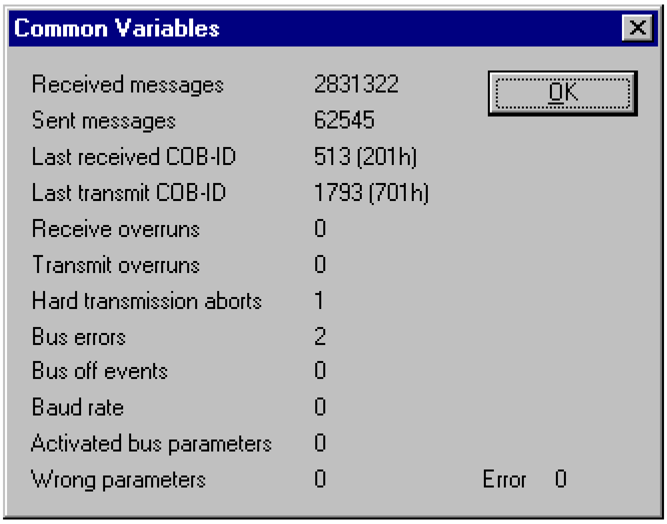
Version compiled	COS
Task state	0
Handshake counter	0
Handshake mode	2
Error	0

Figure 100: PCL_TASK Common Variables

Variable	Meaning
Version compiled	Holds a static text ('COS') indicating the hardware the PLC task was compiled for
Task state	Internal state of the PLC task
Handshake counter	Number handshake cycles executed
Handshake mode	Currently activated handshake mode (1, 2 or 3)

Table 82: PCL_TASK Common Variables

10.2.2 COS_TASK Common Variables



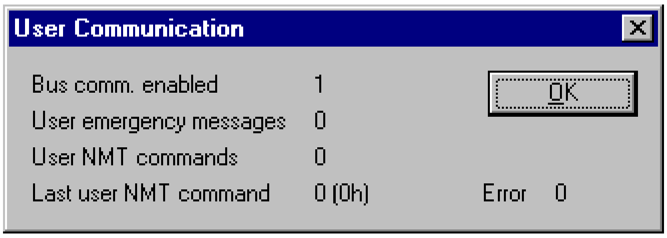
Received messages	2831322
Sent messages	62545
Last received COB-ID	513 (201h)
Last transmit COB-ID	1793 (701h)
Receive overruns	0
Transmit overruns	0
Hard transmission aborts	1
Bus errors	2
Bus off events	0
Baud rate	0
Activated bus parameters	0
Wrong parameters	0
Error	0

Figure 101: COS_TASK Common Variables

Variable	Meaning
Received messages	Number of received CAN messages
Sent messages	Number of messages handed over to CAN chip for transmission (This does not necessarily mean these messages were sent over the bus, see below).
Last received COB-ID	COB ID of most recently received CAN message
Last transmit COB-ID	COB ID of most recently CAN message handed over to CAN chip
Receive overruns	Number of overrun situations in the CAN chip internal receive queue
Transmit overruns	Reserved for future use
Hard transmission aborts	Number of discarded messages because no acknowledging partner could be found on the bus
Bus errors	Counter for bus events detected by the CAN chip. This includes warnings, bus off situations and receive queue overruns. The latter two are also counted in separate variables (see below/above).
Bus off events	Number of bus off events. These indicate severe communication problems on the CAN bus. When the CAN chip detects such a situation it goes to disabled state and is not involved in bus operations anymore. To resume to work it must be re-initialized which is done by the firmware automatically.
Baud rate	Current baud rate the CAN chip is operating at on the bus 0 - 1 Mbit/s 1 - 800 Kbit/s 2 - 500 Kbit/s 3 - 250 Kbit/s 4 - 125 Kbit/s 5 - 100 Kbit/s 6 - 50 Kbit/s 7 - 20 Kbit/s 8 - 10 Kbit/s
Activated bus parameters	0 - valid configuration data received 255 - current configuration data is invalid
Wrong parameters	Reserved for future use

Table 83: COS_TASK Common Variables

10.2.3 COS_TASK User Communication



User Communication

Bus comm. enabled 1

User emergency messages 0

User NMT commands 0

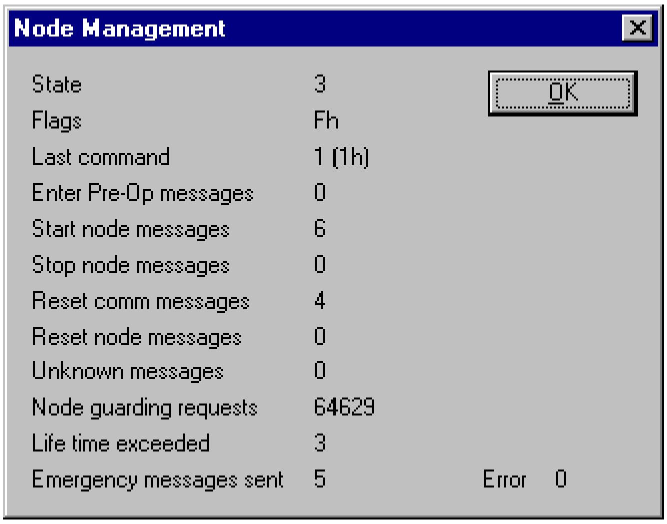
Last user NMT command 0 (0h) Error 0

Figure 102: COS_TASK User Communication

Variable	**Meaning**
Bus comm. enabled	0 - bus communication is disabled because of Not Ready bit set in DPM or HOST watchdog error 1 - bus communication is enabled
User emergency messages	Number of user generated emergency messages sent
User NMT commands	Number of NMT commands received from user
Last user NMT command	NMT command received most recently from user

Table 84: COS_TASK User Communication

10.2.4 COS_TASK Node Management



State	3
Flags	Fh
Last command	1 (1h)
Enter Pre-Op messages	0
Start node messages	6
Stop node messages	0
Reset comm messages	4
Reset node messages	0
Unknown messages	0
Node guarding requests	64629
Life time exceeded	3
Emergency messages sent	5

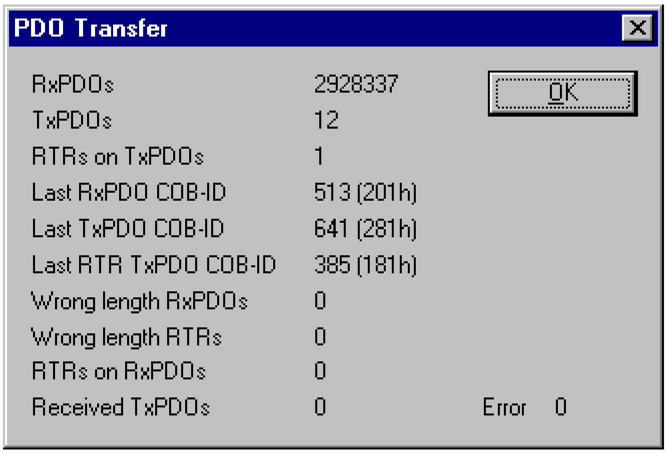
Error 0

Figure 103: COS_TASK Node Management

Variable	Meaning
State	Current Node management state of the DEVICE 0 - Init 1 - Pre-Operational 2 - Prepared 3 - Operational
Flags	Some flags Bit 0 - PDO communication enable Bit 1 - SDO communication enable Bit 2 - Node guarding enable Bit 3 - life guarding enable Bit 7 - Node guarding toggle bit
Last command	Last Node management command received
Enter Pre-Op messages	Number of Enter Pre-Operational State messages received
Start Node messages	Number of Start Node messages received
Stop Node messages	Number of Stop Node messages received
Reset comm messages	Number of Reset Communication messages received
Reset Node messages	Number of Reset Node messages received
Unknown messages	Number of unknown (and ignored) Node management messages received
Node guarding requests	Number of Node guarding requests received
Life time exceeded	Number of life time supervision failures
Emergency messages sent	Number of emergency messages sent by the DEVICE

Table 85: COS_TASK Node Management

10.2.5 COS_TASK PDO Transfer



PDO Transfer

RxPDOs	2928337
TxPDOs	12
RTRs on TxPDOs	1
Last RxPDO COB-ID	513 (201h)
Last TxPDO COB-ID	641 (281h)
Last RTR TxPDO COB-ID	385 (181h)
Wrong length RxPDOs	0
Wrong length RTRs	0
RTRs on RxPDOs	0
Received TxPDOs	0

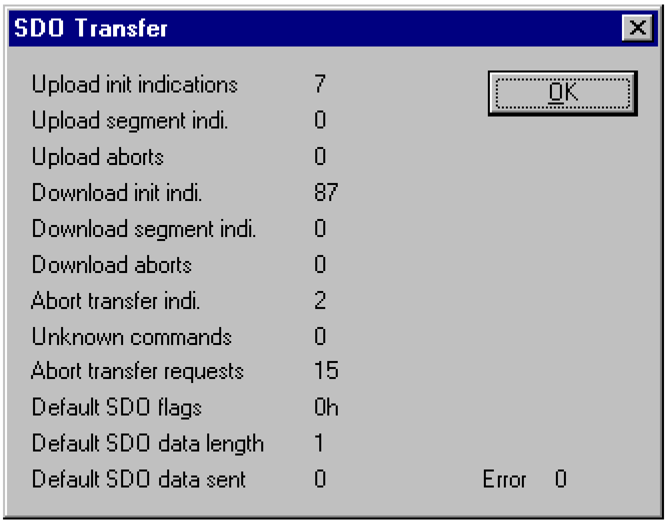
Error 0

Figure 104: COS_TASK PDO Transfer

Variable	**Meaning**
RxPDOs	Number of valid RxPDOs received
TxPDOs	Number of TxPDOs sent
RTRs on TxPDOs	Number of valid RTRs on TxPDOs received
Last RxPDO COB-ID	COB ID of RxPDO most recently received
Last TxPDO COB-ID	COB ID of RxPDO most recently sent
Last RTR TxPDO COB-ID	COB ID of RTR on TxPDO most recently received
Wrong length RxPDOs	Number of RxPDOs with wrong length (unequal internally configured length of this PDO)
Wrong length RTRs	Number of RTRs on TxPDOs with wrong length (unequal 0)
RTRs on RxPDOs	Number of RTRs on RxPDOs
Received TxPDOs	Number TxPDOs received (and ignored)

Table 86: COS_TASK PDO Transfer

10.2.6 COS_TASK SDO Transfer



SDO Transfer

Upload init indications	7
Upload segment indi.	0
Upload aborts	0
Download init indi.	87
Download segment indi.	0
Download aborts	0
Abort transfer indi.	2
Unknown commands	0
Abort transfer requests	15
Default SDO flags	0h
Default SDO data length	1
Default SDO data sent	0

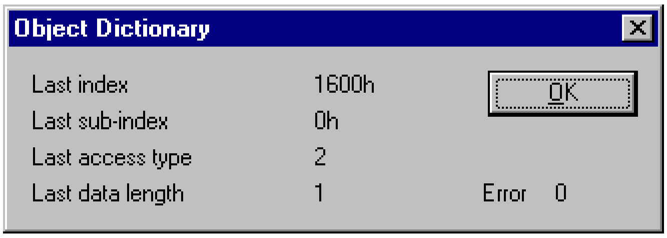
Error 0

Figure 105: COS_TASK SDO Transfer

Variable	Meaning
Upload init indications	Number of upload initial segment indications received
Upload segment indi.	Number of upload segment indications received
Upload aborts	Number of upload operations aborted
Download init indi.	Number of download initial segment indications received
Download segment indi.	Number of download segment indications received
Download aborts	Number of download operations aborted
Abort transfer indi.	Number of abort transfer indications received
Unknown commands	Number of unknown SDO commands received
Abort transfer requests	Number of abort requests sent
Default SDO flags	Flags indicating the current state of the default SDO Bit 0 - upload operation running Bit 1 - download operation running Bit 7 - SDO transfer toggle bit
Default SDO data length	**Download:** number of bytes received during last download **Upload:** number of bytes to be uploaded
Default SDO data sent	**Download:** 0 **Upload:** number of bytes already uploaded

Table 87: COS_TASK SDO Transfer

10.2.7 COS_TASK Object Dictionary



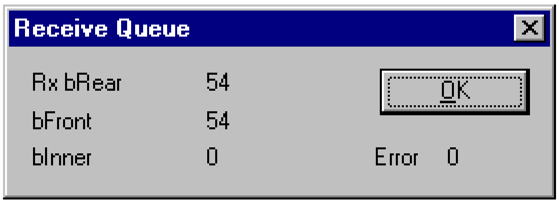
Last index 1600h
Last sub-index 0h
Last access type 2
Last data length 1
Error 0

Figure 106: COS_TASK Object Dictionary

Variable	**Meaning**
Last index	Index of most recent access to object dictionary
Last sub-index	Sub-index of most recent access to object dictionary
Last access type	Operation type of most recent access to object dictionary 1 - read operation 2 - write operation
Last data length	Number of bytes transferred during most recent access to object dictionary

Table 88: COS_TASK Object Dictionary

10.2.8 COS_TASK Receive Queue



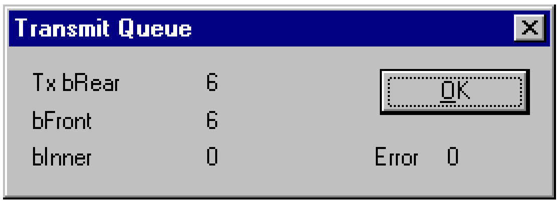
Rx bRear 54
bFront 54
bInner 0
Error 0

Figure 107: COS_TASK Receive Queue

Variable	**Meaning**
Rx bRear	Position of the next message to be read from the queue of incoming CAN messages
bFront	Position of the next incoming CAN message to be inserted into the queue
bInner	Number of messages currently in the queue

Table 89: COS_TASK Receive Queue

10.2.9 COS_TASK Transmit Queue



Transmit Queue

Tx bRear 6

bFront 6

bInner 0

Error 0

Figure 108: COS_TASK Transmit Queue

Variable	Meaning
Tx bRear	Position of the next message to be read from the queue of CAN messages to be sent
bFront	Position of the next CAN message to be inserted into the queue, messages to be sent are queued only if the CAN chip is not able to immediately send the message because the preceding message transfer is pending
bInner	Number of messages currently in the queue

Table 90: COS_TASK Transmit Queue

10.3 COB-ID (Predefined Connection Set)

COB-ID means Communication Object Identifier. This is the 11 bit telegram identifier of the CAN telegram. The higher 4 bits (bit 10 to 8) is the function code and the lower 7 bits (bit 7 to 0) is the bus address of the Node.

Broadcast Objects:

Object	**Function Code**	**COB ID hex**	**COB ID dec**	**Index in the Object Directory**
NMT	0000	00H	0	-
SYNC	0001	80H	128	1005H, 1006H, 1007H
TIME STAMP	0010	100H	256	1012H, 1013H

Table 91: COB ID - Broadcast Objects

Peer-to-Peer Objects:

Object	**Function Code**	**COB ID hex**	**COB ID dec**	**Index in the Object Directory**
Emergency	0001	81H-FFH	129-255	1014H, 1015H
PDO 1 (tx)	0011	181H-1FFH	385-511	1800H
PDO 1 (rx)	0100	201H-27FH	513-639	1400H
PDO 2 (tx)	0101	281H-2FFH	641-767	1801H
PDO 2 (rx)	0110	301H-37FH	769-895	1401H
PDO 3 (tx)	0111	381H-3FFH	897-1023	1802H
PDO 3 (rx)	1000	401H-47FH	1025-1151	1402H
PDO 4 (tx)	1001	481H-4FFH	1153-1279	1803H
PDO 4 (rx)	1010	501H-57FH	1281-1407	1403H
SDO (tx)	1011	581H-5FFH	1409-1535	1200H
SDO (rx)	1100	601H-67FH	1537-1663	1200H
NMT Error Control	1110	701H-77FH	1793-1919	1016H, 1017H

Table 92: COB ID - Peer-to-Peer Objects

10.4 Object Dictionary

The Object Dictionary is a collection of data, which have influence on the application and the communication of a CANopen device and the device can be configured with this data collection. The entries are structured by the index and the sub index.

10.4.1 Object Name and Object Code

The following table shows a list of the Object Codes:

Object Name	Comment	Object Code
NULL	A dictionary entry with no data fields	0
DOMAIN	Large variable amount of data e.g. executable program code	2
DEFTYPE	Denotes a type definition such as a Boolean, UNSIGED 16, float and so on	5
DEFSTRUCT	complex Data type definition, e.g. PDO Mapping-Structure	6
VAR	A single value such as Unsigned 8, Boolean, visible string etc.	7
ARRAY	A multiple data field object where each data field is a simple variable of the same basic data type e.g. array of Unsigned 16 Sub-index 0 is of Unsigned 8 and therefore not part of the array data.	8
RECORD	A multiple data field object where the data fields may be any combination of simple variables. Sub-index 0 is of Unsigned 8 and therefore not part of the record data.	9

Table 93: Object Codes

Note: The list of the Object Names and Object Codes is no information that the Master or the Node support the respective data type.

10.4.2 Object Dictionary Data Types

The following table is a survey of the data types and an extract of the CANopen specification.

Note: The list of the data types is no information that the Master or the Node support the respective data type.

Range	Index (Hex)	Object	Name
Reserved	0000	reserved	reserved
Static data types	0001	DEFTYPE	BOOLEAN
	0002	DEFTYPE	INTERGER 8
	0003	DEFTYPE	INTERGER 16
	0004	DEFTYPE	INTERGER 32
	0005	DEFTYPE	UNSIGNED 8
	0006	DEFTYPE	UNSIGNED 16
	0007	DEFTYPE	UNSIGNED 32
	0008	DEFTYPE	REAL 32
	0009	DEFTYPE	VISIBLE_STRING
	000A	DEFTYPE	OCTET_STRING
	000B	DEFTYPE	UNICODE_STRING
	000C	DEFTYPE	TIME_OF_DAY
	000D	DEFTYPE	TIME_DIFFERENCE
	000E	DEFTYPE	BIT_STRING
	000F	DEFTYPE	DOMAIN
	0010	DEFTYPE	INTERGER 24
	0011	DEFTYPE	REAL 64
	0012	DEFTYPE	INTERGER 40
	0013	DEFTYPE	INTERGER 48
	0014	DEFTYPE	INTERGER 56
	0015	DEFTYPE	INTERGER 64
	0016	DEFTYPE	UNSIGNED 24
	0017	reserved	reserved
	0018	DEFTYPE	UNSIGNED 40
	0019	DEFTYPE	UNSIGNED 48
	001A	DEFTYPE	UNSIGNED 56
	001B	DEFTYPE	UNSIGNED 64
	001C-001F	reserved	reserved

Continuation see next page.

Complex data types	0020	DEFSTRUCT	PDO_COMMUNICATION_PARAMETER
	0021	DEFSTRUCT	PDO_MAPPING
	0022	DEFSTRUCT	SDO_PARAMETER
	0023	DEFSTRUCT	IDENTITY
	0024-003F	reserved	reserved
Manufacturer specific complex data types	0040-005F	DEFSTRUCT	Manufacturer specific complex data types
Device profile data types	0060-007F	DEFTYPE	Device profile (0) specific standard data types
	0080-009F	DEFSTRUCT	Device profile (0) specific complex data types
	00A0-00BF	DEFTYPE	Device profile 1 specific standard data types
	00C0-00DF	DEFSTRUCT	Device profile 1 specific complex data types
	00E0-00FF	DEFTYPE	Device profile 2 specific standard data types
	0100-011F	DEFSTRUCT	Device profile 2 specific complex data types
	0120-013F	DEFTYPE	Device profile 3 specific standard data types
	0140-015F	DEFSTRUCT	Device profile 3 specific complex data types
	0160-017F	DEFTYPE	Device profile 4 specific standard data types
	0180-019F	DEFSTRUCT	Device profile 4 specific complex data types
	01A0-01BF	DEFTYPE	Device profile 5 specific standard data types
	01C0-01DF	DEFSTRUCT	Device profile 5 specific complex data types
	01E0-01FF	DEFTYPE	Device profile 6 specific standard data types
	0200-021F	DEFSTRUCT	Device profile 6 specific complex data types
	0220-023F	DEFTYPE	Device profile 7 specific standard data types
	0240-025F	DEFSTRUCT	Device profile 7 specific complex data types
Reserved	0300-0FFF	reserved	reserved

Table 94: Object Dictionary Data Types

10.4.3 Object Dictionary Profile

The following table is a survey of the profile object dictionary and an extract of the CANopen specification.

Note: The list of the single objects is no information that the Master or the Node support the respective object and the function which is associated with it.

Range	Index (Hex)	Object	Name	Type / Access
Communication Profile	1000	VAR	Device Type	Unsigned 32 / ro
	1001	VAR	Error Register	Unsigned 8 / ro
	1002	VAR	Manufacturer Status Register	Unsigned 32 / ro
	1003	ARRAY	Pre-defined Error Field	Unsigned 32 / ro
	1004	-	Reserved	-
	1005	VAR	COB-ID SYNC	Unsigned 32 / rw
	1006	VAR	Communication Cycle Period	Unsigned 32 / rw
	1007	VAR	Synchronous Window Length	Unsigned 32 / rw
	1008	VAR	Manufacturer Device Name	Visible_string / c
	1009	VAR	Manufacturer Hardware Version	Visible_string / c
	100A	VAR	Manufacturer Software Version	Visible_string / c
	100B	-	Reserved	-
	100C	VAR	Guard Time	Unsigned 32 / rw
	100D	VAR	Life Time Factor	Unsigned 32 / rw
	100E	-	Reserved	-
	100F	-	Reserved	-
	1010	VAR	Store Parameters	Unsigned 32 / rw
	1011	VAR	Restore Default Parameters	Unsigned 32 / rw
	1012	VAR	COB-ID TIME	Unsigned 32 / rw
	1013	VAR	High Resolution Time Stamp	Unsigned 32 / rw
	1014	VAR	COB-ID EMCY	Unsigned 32 / rw
	1015	VAR	Inhibit Time EMCY	Unsigned 16 / rw
	1016	ARRAY	Consumer Heartbeat Time	Unsigned 32 / rw
	1017	VAR	Producer Heartbeat Time	Unsigned 16 / rw
	1018	RECORD	Identity Object	Identity / ro
	1018 / 0		Number of Entries	Unsigned 8
	1018 / 1		Vendor Information	Unsigned 32
	1018 / 2		Product Code	Unsigned 32
	1018 / 3		Revision Number	Unsigned 32
	1018 / 4		Serial Number	Unsigned 32
	1019-11FF	-	Reserved	-

Continuation see next page.

Range	Index (Hex)	Object	Name	Type / Access
Communication Profile	1200	RECORD	Server 1. SDO Parameter	SDO_Parameter / ro
	1200 / 0		Number of Entries	Unsigned 8
	1200 / 1		COB-ID Client -> Server	Unsigned 32
	1200 / 2		COB-ID Client <- Server	Unsigned 32
	1200 / 3		NodeID	Unsigned 8
	1201-1277	RECORD	Server 2. to 127. SDO Parameter	SDO_Parameter / rw
	1280-12FF	RECORD	Client SDO Parameter	SDO_Parameter / rw
	1300-13FF	-	Reserved	-
	1400-15FF	RECORD	Receive PDO Parameter	PDO_Com_Para / rw
	1400 / 0		Number of Entries	Unsigned 8
	1400 / 1		COB-ID	Unsigned 32
	1400 / 2		Transmission Type	Unsigned 8
	1400 / 3		Transmit Prohibited Time	Unsigned 16
	1400 / 4		Reserved	Unsigned 8
	1400 / 5		Event Timer	Unsigned 16
	1600-17FF	ARRAY	Receive PDO Mapping	PDO_Mapping / rw
	1600 / 0		Number of Entries	Unsigned 8
	1600 / 1		1. Object	Unsigned 32
	1600 / 2		2. Object	Unsigned 32
	1600 / ...		n. Object	Unsigned 32
	1600 / 40		64. Object	Unsigned 32
	1800-19FF	RECORD	Transmit PDO Parameter	PDO_Com_Para / rw
	1A00-1BFF	ARRAY	Transmit PDO Mapping	PDO_Mapping / rw
	1C00-1FFF	-	Reserved	-
Manufacturer Specific Profile	2000-5FFF			
Standardized Device Profiles	6000-67FF		Device Profile 1	
	6800-6FFF		Device Profile 2	
	7000-77FF		Device Profile 3	
	7800-7FFF		Device Profile 4	
	8000-87FF		Device Profile 5	
	8800-8FFF		Device Profile 6	
	9000-97FF		Device Profile 7	
	9800-9FFF		Device Profile 8	
Reserved	A000-FFFF	-	Reserved	-

Table 95: Object Dictionary Profile

10.5 Communication Profile, Device Profile and Device Type

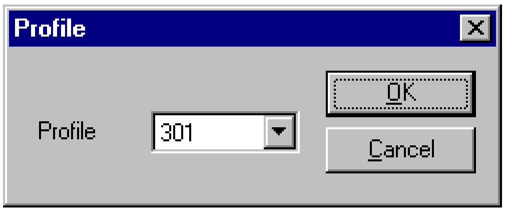
The Communication Profile DS 301 specifies, how to communicate. The Device Profiles DS 401ff specify, what is communicated.

Device Profile	Description
301	Common communication profile according to DS301
401	Device profile for I/O modules
402	Device profile for drives
405	Device Profile for IEC 61131-3 Programmable Devices
406	Device profile for encoder

Table 96: Device Profile and Device Type

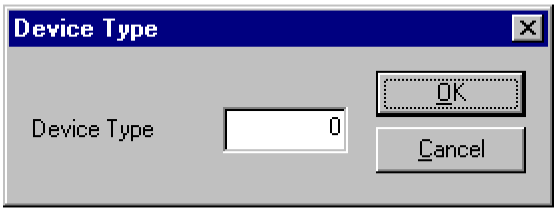
10.5.1 Communication Profile 301

The communication profile DS 301 is a common profile. It is the basic of CANopen communication and lays down, how the device on the CANopen communicates with each other.



Profile 301

Figure 109: Device Profile 301

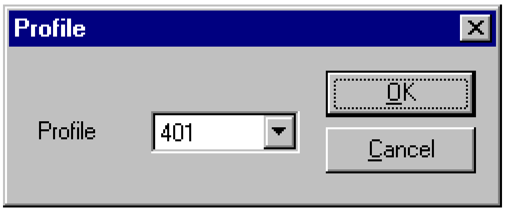


Device Type 0

Figure 110: Selection of the Device Type in case of Device Profile 301

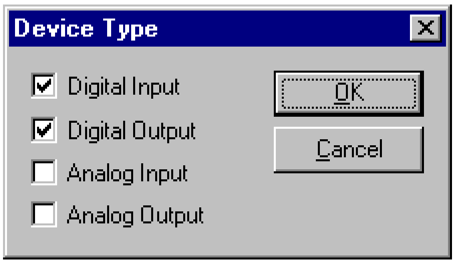
10.5.2 Device Profile 401 - Device Profile for I/O Modules

The device profile DS 401 is a profile for I/O modules.



Profile: 401

Figure 111: Device Profile 401



☑ Digital Input
☑ Digital Output
☐ Analog Input
☐ Analog Output

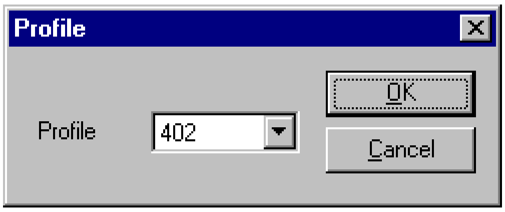
Figure 112: Selection of the Device Type in case of Device Profile 401

Device Profile	Device Type	Description
401	Digital Input Digital Output Analog Input Analog Output	Device Profile for I/O Modules

Table 97: Device Profile for I/O Modules

10.5.3 Device Profile 402 - Device Profile for Drives

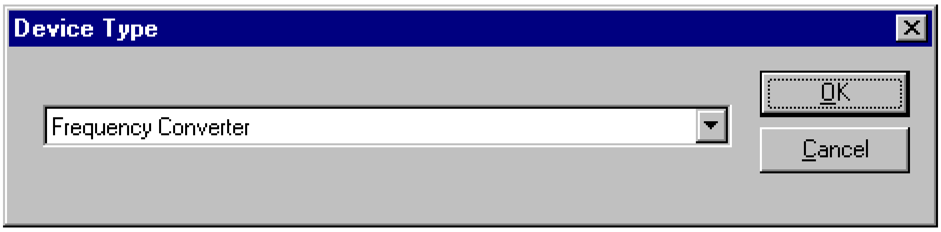
The device profile DS 402 is a profile for drives.



Profile

Profile 402

Figure 113: Device Profile 402



Device Type

Frequency Converter

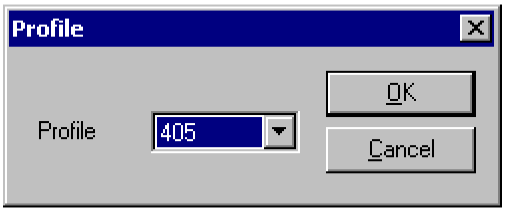
Figure 114: Selection Device Type in case of Device Profile 402

Device Profile	Device Type	Description
402	Frequency Converter Servo Drive Stepper Motor I/O Module Multi device module	Device profile for drives

Table 98: Device Profile for Drives

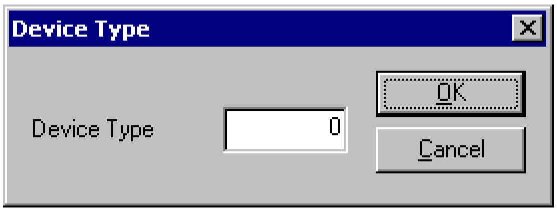
10.5.4 Device Profile 405 - Device Profile for IEC 61131-3 Programmable Devices

The device profile DPS-405 is a profile for IEC 61161-3 programmable devices like PLCs.



Profile: 405

Figure 115: Device Profile 405



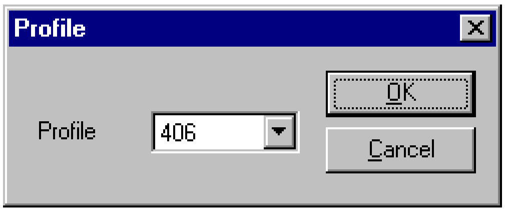
Device Type: 0

Figure 116: Selection Device Type in case of Device Profile 405

The meaning of the device type is manufacturer specific and not universally applicable in the specification. For further information see the device description of the manufacturer.

10.5.5 Device Profile 406 - Device Profile for Encoder

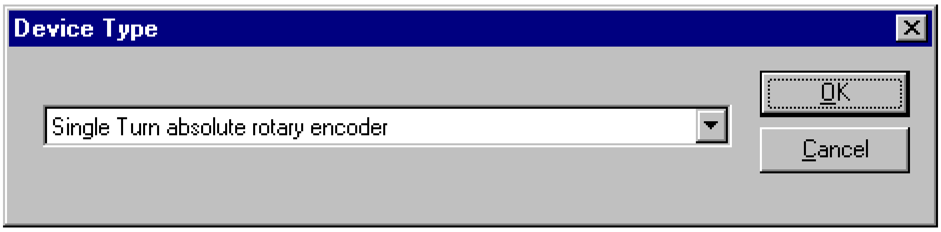
The device profile DS 406 is a profile for encoder.



Profile

Profile 406

Figure 117: Device Profile 406



Device Type

Single Turn absolute rotary encoder

Figure 118: Selection of the Device Type in case of Device Profile 406

Device Profile	Device Type	Description
406	Single Turn absolute rotary encoder Multi Turn absolute rotary encoder Single Turn absolute rotary encoder with electronic turn count Incremental rotary encoder Incremental rotary encoder with electronic counting Incremental linear encoder Incremental linear encoder with electronic counting Absolute linear encoder Absolute linear encoder with cyclic coding	Device profile for encoder

Table 99: Device Profile for Encoder

10.6 PDO Mapping Method

The PDO Mapping with degree of freedom was fixed in the specification DS301 V3. The System Configurator produces the following PDO Mapping:

- Sub index 0 the number of objects (value N) is entered in object 16xx (and object 1Axx respectively).
- Sub index 1 to N are entered in the objects which are to be mapped in object 16xx (and object 1Axx respectively).

The PDO Mapping was laid down more exactly in the specification DS301 V4. Thereby particular the fist mapped information in the Node is deleted, then it is described new and after this it is set to valid.

- To delete the information of the PDO Mapping in the Node (and to set it back to the default mapping respectively), in object 16xx (and object 1Axx respectively) Sub index 0 the value 0 is written down.
- The objects which are to be mapped are entered in object 16xx (and respectively object 1Axx) Sub index 1 to N.
- The number of objects is entered (value N) in object 16xx (and respectively object 1Axx).

10.7 NMT State Machine (State Diagram)

NMT stands for Network Management.

The following diagram shows the possible states of a CANopen Node.

Note: The following figure contains the english terms from the CANopen specification.

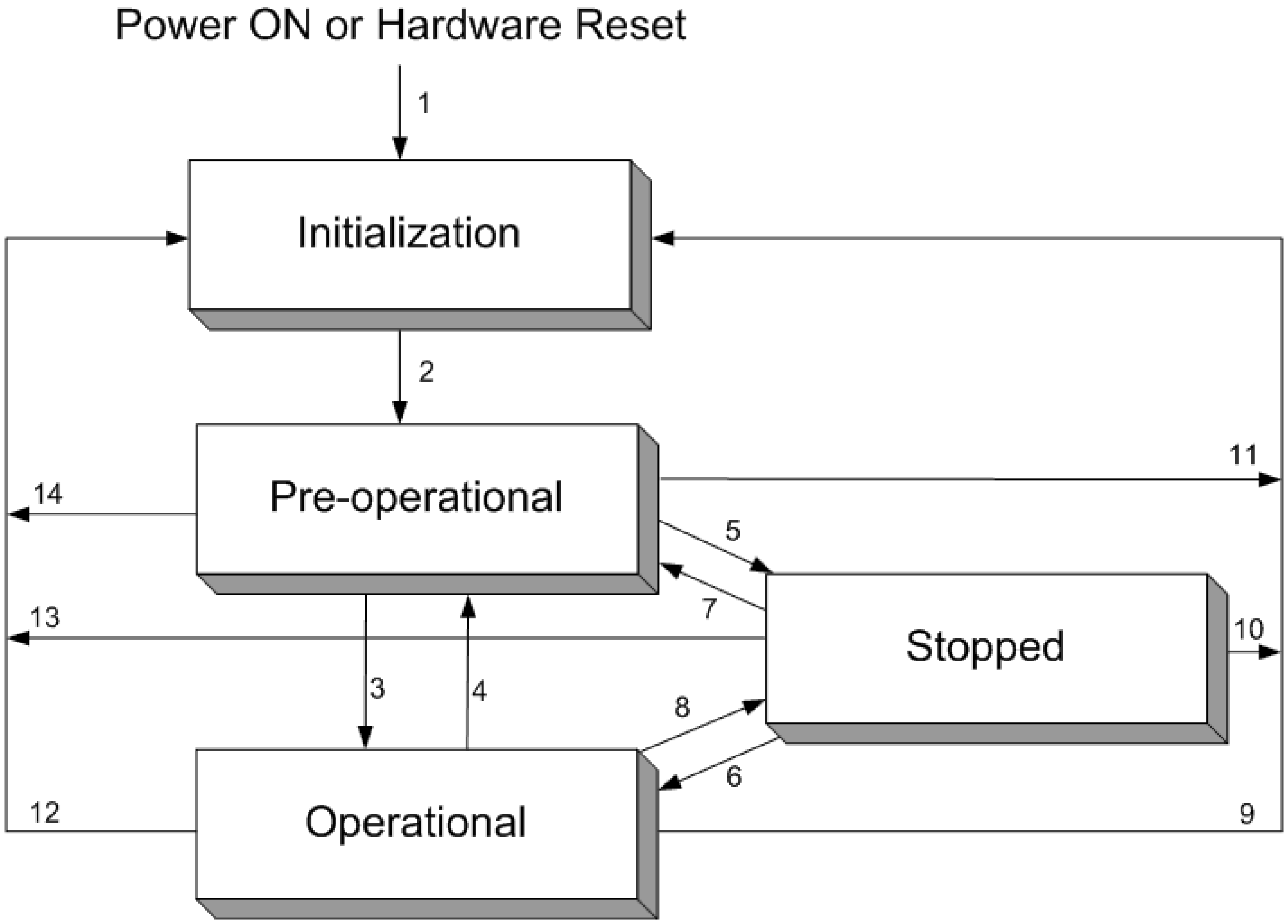


Figure 119: NMT-State Machine

Number	Meaning
1	At Power on the initialization state is entered autonomously
2	Initialization finished -enter PRE_OPERATIONAL automatically
3, 6	Start_Remote_Node indication
4, 7	Enter_PRE-OPERATIONAL_State indication
5, 8	Stop_Remote_Node indication
9, 10, 11	Reset_Node indication
12, 13, 14	Reset_Communication indication

Table 100: Description NMT-State Machine

10.7.1 Communication Characteristics in the different NMT States

The following table shows the possible communication in the respective NMT states.

Communication	Initialization	Pre-Operational	Operational	Stopped
PDO			X	
SDO		X	X	
SYNC		X	X	
Time Stamp		X	X	
EMCY		X	X	
BootUp	X			
NMT		X	X	X

Table 101: Communication in the different NMT States

10.8 LSS/LMT Services

LSS stands for Layer Setting Services, LMT stands for Layer Management and is an older designation.

LSS/LMT supports access to the basic parameter like

- Baud Rate
- Node ID

via the CAN network also without mechanical setting possibilities on the Node. The communication is based on a Master/Slave relationship and uses the COB-ID 2020 (07E4H, Slave to Master) and 2021 (07E5H, Master to Slave).

The LSS/LMT Slave need to be in the NMT state Stop, to perform the LSS/LMT services. The LSS/LMT Slave is able to take the following both states

- Operation Mode = Operating mode with valid parameters and
- Configuration Mode = Configuration Mode.

Note: It is permitted to couple only one Node to the Master at a time.

10.9 Emergency Telegrams

Emergency Telegrams are sent by the Node in case of a Node internal event.

The Emergency Telegram has the following structure:

Byte 1	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**	**Byte 8**
Error Code LSB	Error Code MSB	Error Register	Comm. Error	Device Error	Emergency Trigger	Info 0	Info 1

Table 102: Emergency Telegram (Structure)

- Error Code (Byte 1 and 2): See section *Emergency Telegram Error Codes* on page *192*.
- Error Register (Byte 3): Object 1001H. See device description of the Node manufacturer.

Bit	**Meaning**
0	generic error
1	current
2	voltage
3	temperature
4	communication error (overrun, error state)
5	device profile specific
6	reserved
7	manufacturer specific

Table 103: Structure of the Error Register

- Manufacturer specific error field (Byte 4 to 8): See device description of the Node manufacturer.

10.9.1 Emergency Telegram Error Codes

The meaning of the Error Codes is shown in the following table:

Error Code (Hex)	Meaning
00xx	No error or reset
10xx	Generic error
20xx	Current
21xx	Current, device input side
22xx	Current inside the device
23xx	Current, device output side
30xx	Voltage
31xx	Mains voltage
32xx	Voltage, inside the device
33xx	Output Voltage
40xx	Temperature
41xx	Ambient temperature
42xx	Device temperature
50xx	Device Hardware
60xx	Device Software
61xx	Internal Software
62xx	User Software
63xx	Data Set
70xx	Additional Modules
80xx	Monitoring
81xx	Communication
8110	CAN Overrun (Object lost)
8120	CAN in Error Passive Mode
8130	Life Guarding Error or Heartbeat Error
8140	recover from bus off
82xx	Protocol Error
8210	PDO not processed due to length error
8220	PDO length exceeded
90xx	External Error
F0xx	Additional Functions
FFxx	Device specific

Table 104: Emergency Error Codes

Note: The table Emergency Error Codes is a common list. To see the exact meaning of these codes we refer to the Node manufacturer.

11 Lists

11.1 List of Figures

11.2 List of Tables

12 Glossary

COB-ID

Communication object identifier. Table in section *COB-ID* on page *176.*

LMT

Layer Management

LSS

Layer Setting Services

NMT

Network Management. This contains the functions configuration, initialization and supervision of the network devices.

SyCon

System Configurator.

Configuration- and Diagnostic Tool.

13 Contacts

Headquarters

Germany
Hilscher Gesellschaft für
Systemautomation mbH
Rheinstrasse 15
65795 Hattersheim
Phone: +49 (0) 6190 9907-0
Fax: +49 (0) 6190 9907-50
E-Mail: info@hilscher.com

Support
Phone: +49 (0) 6190 9907-99
E-Mail: de.support@hilscher.com

Subsidiaries

China
Hilscher Systemautomation (Shanghai) Co. Ltd.
200010 Shanghai
Phone: +86 (0) 21-6355-5161
E-Mail: info@hilscher.cn

Support
Phone: +86 (0) 21-6355-5161
E-Mail: cn.support@hilscher.com

France
Hilscher France S.a.r.l.
69500 Bron
Phone: +33 (0) 4 72 37 98 40
E-Mail: info@hilscher.fr

Support
Phone: +33 (0) 4 72 37 98 40
E-Mail: fr.support@hilscher.com

India
Hilscher India Pvt. Ltd.
New Delhi - 110 065
Phone: +91 11 43055431
E-Mail: info@hilscher.in

Italy
Hilscher Italia S.r.l.
20090 Vimodrone (MI)
Phone: +39 02 25007068
E-Mail: info@hilscher.it

Support
Phone: +39 02 25007068
E-Mail: it.support@hilscher.com

Japan
Hilscher Japan KK
Tokyo, 160-0022
Phone: +81 (0) 3-5362-0521
E-Mail: info@hilscher.jp

Support
Phone: +81 (0) 3-5362-0521
E-Mail: jp.support@hilscher.com

Korea
Hilscher Korea Inc.
Suwon, Gyeonggi, 443-734
Phone: +82 (0) 31-695-5515
E-Mail: info@hilscher.kr

Switzerland
Hilscher Swiss GmbH
4500 Solothurn
Phone: +41 (0) 32 623 6633
E-Mail: info@hilscher.ch

Support
Phone: +49 (0) 6190 9907-99
E-Mail: ch.support@hilscher.com

USA
Hilscher North America, Inc.
Lisle, IL 60532
Phone: +1 630-505-5301
E-Mail: info@hilscher.us

Support
Phone: +1 630-505-5301
E-Mail: us.support@hilscher.com