User Manual for

HE800DNM650/HEDNM650 & HE800DNS600/HEDNS600

DeviceNet

Master and Slave SmartStack™ Modules

PREFACE

This manual explains how to use the Horner APG HSyCon software product.

For user manual updates and technical support contact :

Horner APG *(USA)*	***Horner APG*** *(Europe)*
Technical Support (317) 916-4274	*Technical Support +353-21-4321266*
web-site www.horner-apg.com.	*web-site www.horner-apg.com*

LIMITED WARRANTY AND LIMITATION OF LIABILITY

Horner APG, LLC. ("HE-APG") warrants to the original purchaser that the Operator Station manufactured by HE is free from defects in material and workmanship under normal use and service. The obligation of HE-APG under this warranty shall be limited to the repair or exchange of any part or parts which may prove defective under normal use and service within two (2) years from the date of manufacture or eighteen (18) months from the date of installation by the original purchaser whichever occurs first, such defect to be disclosed to the satisfaction of HE-APG after examination by HE-APG of the allegedly defective part or parts. THIS WARRANTY IS EXPRESSLY IN LIEU OF ALL OTHER WARRANTIES EXPRESSED OR IMPLIED INCLUDING THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR USE AND OF ALL OTHER OBLIGATIONS OR LIABILITIES AND HE-APG NEITHER ASSUMES, NOR AUTHORIZES ANY OTHER PERSON TO ASSUME FOR HE-APG, ANY OTHER LIABILITY IN CONNECTION WITH THE SALE OF THE Operator Station. THIS WARRANTY SHALL NOT APPLY TO THE Operator Station OR ANY PART THEREOF WHICH HAS BEEN SUBJECT TO ACCIDENT, NEGLIGENCE, ALTERATION, ABUSE, OR MISUSE. HE MAKES NO WARRANTY WHATSOEVER IN RESPECT TO ACCESSORIES OR PARTS NOT SUPPLIED BY HE. THE TERM "ORIGINAL PURCHASER", AS USED IN THIS WARRANTY, SHALL BE DEEMED TO MEAN THAT PERSON FOR WHOM THE Operator Station IS ORIGINALLY INSTALLED. THIS WARRANTY SHALL APPLY ONLY WITHIN THE BOUNDARIES OF THE CONTINENTAL UNITED STATES.

In no event, whether as a result of breach of contract, warranty, tort (including negligence) or otherwise, shall HE-APG or its suppliers be liable of any special, consequential, incidental or penal damages including, but not limited to, loss of profit or revenues, loss of use of the products or any associated equipment, damage to associated equipment, cost of capital, cost of substitute products, facilities, services or replacement power, down time costs, or claims of original purchaser's customers for such damages.

List of Revisions

Index	Date	Version	Chapter	Revision
	21-May-04	01	All	Initial Release
	18-Mar-07	02	3	Added Cscape Configuration Section
	10-Jul-08	03	9	Added Explicit Messaging chapter

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1 Scope

This manual shows how to connect and configure the DeviceNet Master or Slave Smartstack Modules. *HSyCon,* a Windows™-based software package, is an easy-to-use configuration package for use with the SmartStack COM range of fieldbus modules and Cscape or Cbreeze, a Windows™ based package for use with the OCS/ TIU ranges. The software user's guide is contained in this manual.

A basic level of understanding of Microsoft Windows™ technology and operation is assumed. The manual assumes that the user is familiar with Windows 95, Windows 98, Windows NT, Win 2000 or XP.

1.2 Introduction

The Fieldbus Smartstack module range require only three stages to get them operational, these are:

1. Physical installation and connection.
2. Configuration of the fieldbus interface.
3. Configuration of Cscape / Cbreeze to map the fieldbus data.

The system is comprised of two separate software software functions, the fieldbus interface software running independently in the COM module and the OCS/TIU firmware running in the main module. Data and commands are exchanged via a dual port ram interface. The configuration of the COM module is via the RS232 serial port on the module. For correct operation the number of registers assigned in the OCS must match the number required by the Master or Slave module configuration.

The Smartstack module should be configured with the OCS/TIU software first as otherwise it will be held in reset and cannot be configured.

1.3 Installing and Removing a SmartStack Module

The following section describes how to install and remove a SmartStack Module.

Caution: To function properly and avoid possible damage, do not install more than four SmartStack™ Modules per OCS or RCS.
Do not attempt to install or remove a SmartStack module with the units powered on.

1.4 Installing SmartStack Modules

1. Hook the tabs. Each SmartStack Module has two tabs that fit into slots located on the OCS. (The slots on the OCS are located on the back cover.)

2. Press the SmartStack Module into the “locked” position, making sure to align the SmartStack Module fasteners with the SmartStack receptacles on the OCS.

1.5 Removing SmartStack Modules

1. Using a flathead screwdriver, lever up the end of the SmartStack Module (opposite end to tabs) and swing the module out.
2. Lift out the tabs of the module.

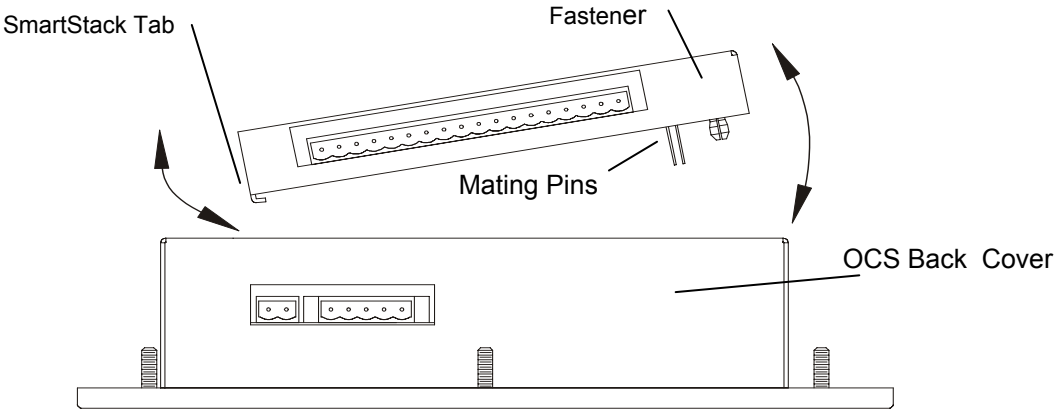


Figure 2.1 – Installing a SmartStack Module in an OCS

1.6 Status LED definitions

Signal	LED Colour	State	Definition
NET	RED	ON	Critical Link Failure
		Flashing	Connection Time out
		OFF	Device Not Powered
	GREEN	ON	ON-LINE Link OK
		Flashing	ON-LINE, Not connected
		OFF	Device Not Powered
MOD	RED	ON	Unrecoverable Fault
		FLASHING	Minor Fault
		OFF	No Power
	GREEN	ON	Normal Operation
		FLASHING	Configuration Failure
		OFF	No Power
RDY	YELLOW	ON	COM ready
		FLASHING CYCLIC	Bootstrap Loader Active
		FLASHING NON CYCLIC	Hardware or System Error
		OFF	Hardware Error
RUN	GREEN	ON	Communication Running
		FLASHING NON CYCLIC	Parameter Error
		OFF	Communication Stopped

Status Signals of the DNM650 and DNS600

1.7 Main Functions

The main functions of the DEVICENET System Configurator are:

- Universal Fieldbus Configurator - Configuration of the complete Fieldbus range with one package.
- Documents Fieldbus system - detailed documentation of the Fieldbus network may be printed.
- Standardised configuration files – allows use of protocol specific standardised configuration files.
- Diagnostic tool – upon configuration download the software may be switched into diagnostic mode.

CHAPTER 2: INSTALLATION

2.1 System Requirments

- PC with 486-, Pentium processor or higher.
- Windows 95/98/ME, Windows NT/2000/XP.
- Free disk space: 30 - 80 Mbyte.
- CD ROM drive.
- RAM: min. 16 Mbyte.
- Graphic resolution: min. 800 x 600 pixel.
- Windows 95: Service Pack 1 or higher.
- Windows NT: Service Pack 3 or higher.
- Keyboard and Mouse.

2.2 System Installation

It is recommended that all application programs on the system are closed before installation begins. Insert the Cscape CD in the local CD ROM drive. The HSycon installation program will start automatically when installation of Cscape is complete (see Fig 2.1)

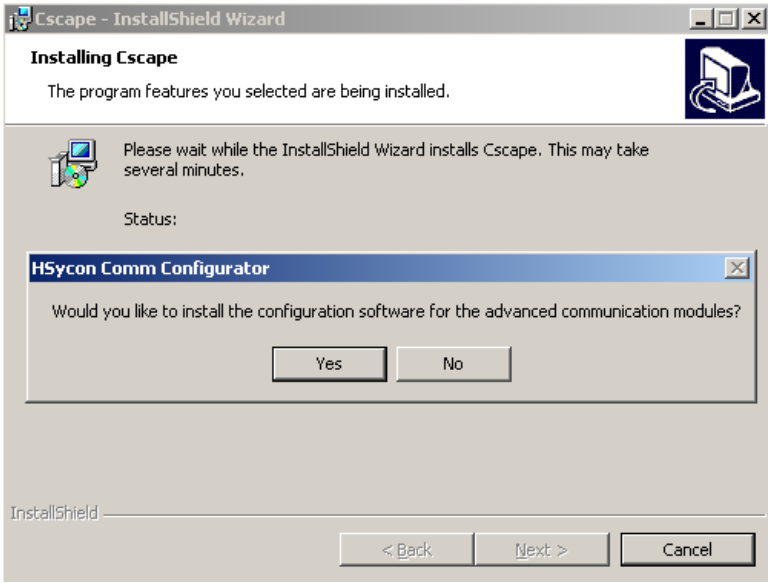


Fig 2.1

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

The installation program asks for the components to install. Answer these questions with **Yes** or **No**. Tick 'No' for the OPC Server function, it is not included with this installation pack.

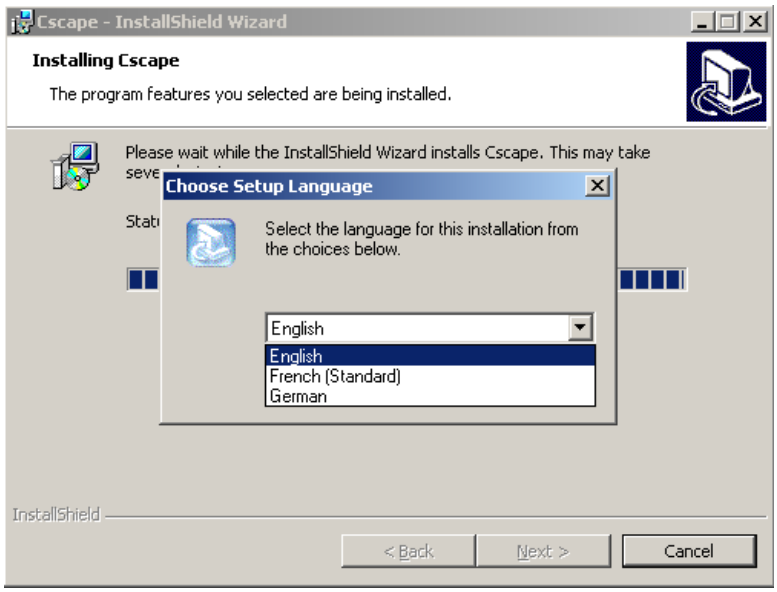


Fig 2.2

2.3 Installation of the System Configurator HSyCon

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

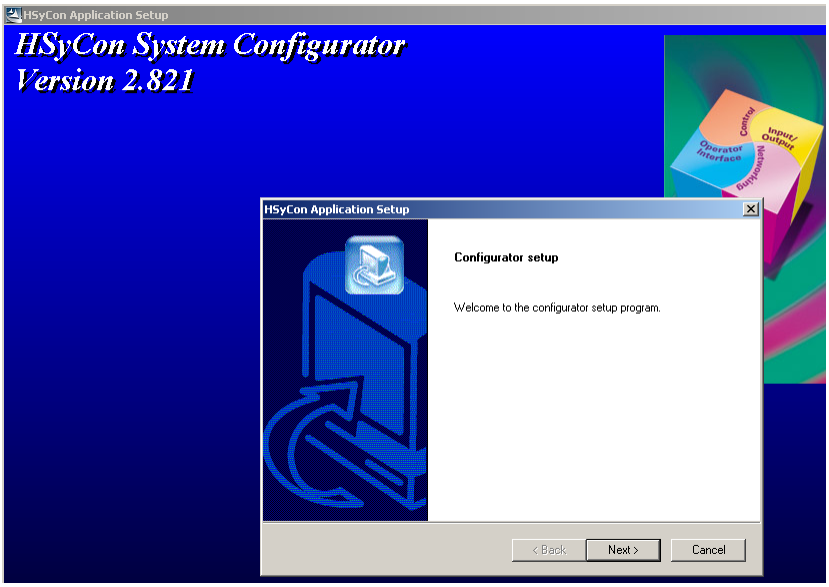


Fig 2.3

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

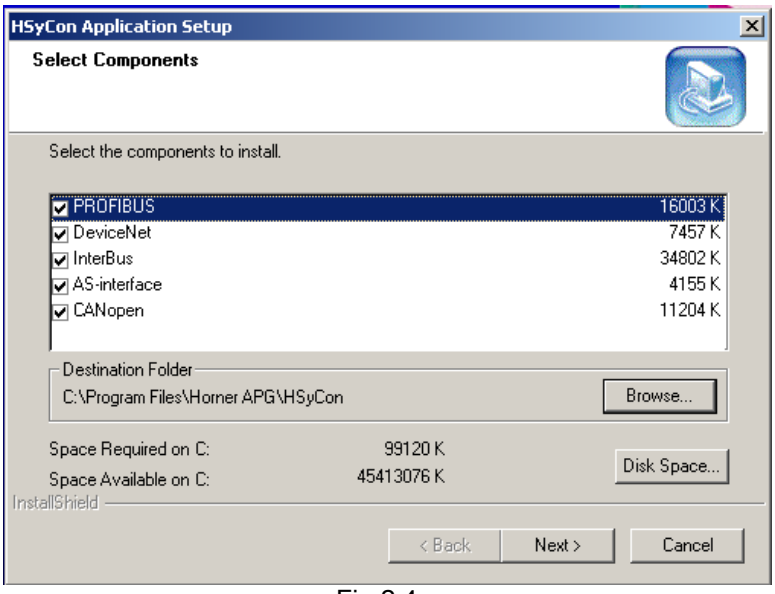


Fig 2.4

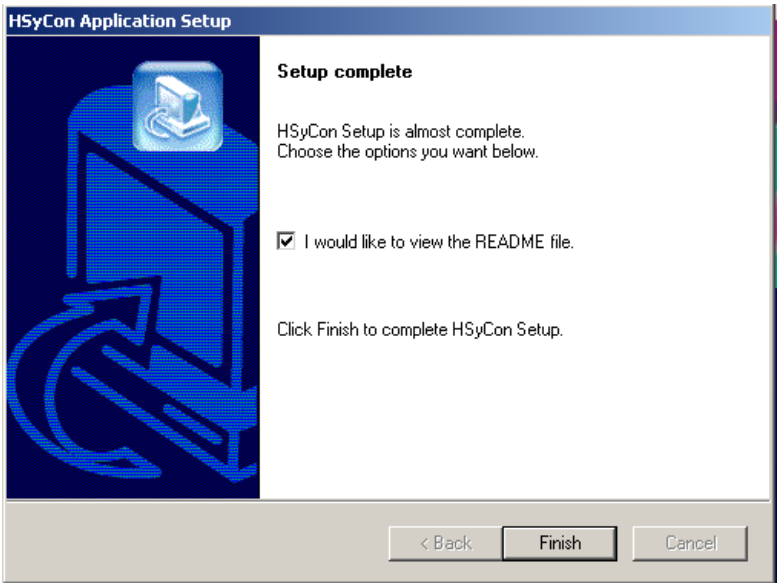


Fig 2.5

Finally the following files are entered in the system Registry.

- System DLLs
- The application

NOTES

CHAPTER 3: GETTING STARTED – CSCAPE CONFIGURATION

3.1 Scope

This chapter describes the procedures for configuring the DeviceNet Master and slaves. This includes loading EDS files, saving, downloading and assigning I/O.

3.2 Configuring Cscape.

The following describes the steps involved to setup Cscape. Attach the communications module to the appropriate OCS unit. Open Cscape. All I/O is setup through the I/O Configure Menu in Cscape:

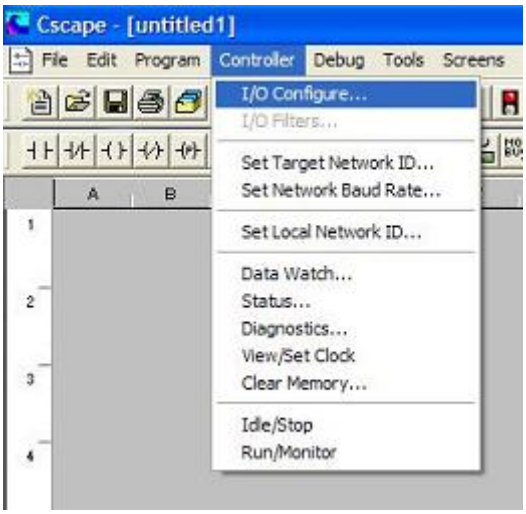


Figure 3.1

The following window is displayed. Select the CONFIG button adjacent to the first empty slot (nearest the main unit).

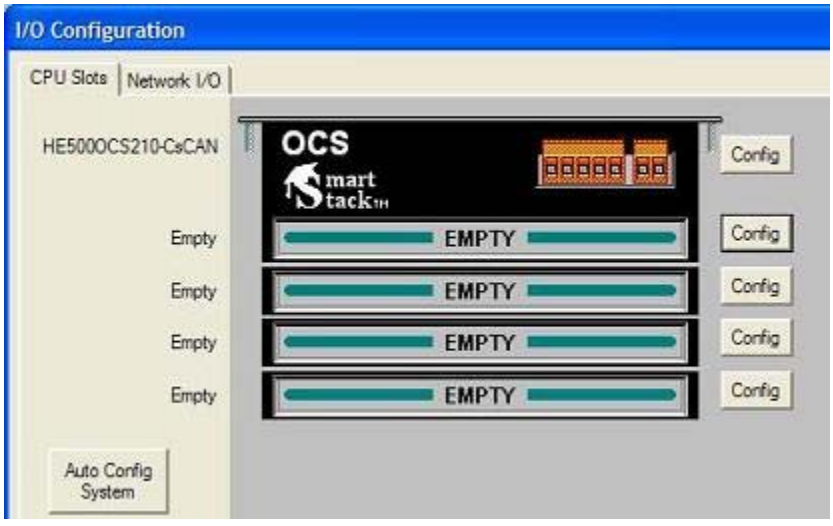


Figure 3.2

Select the COMM Tab. From here select the appropriate DeviceNet Module and click OK.

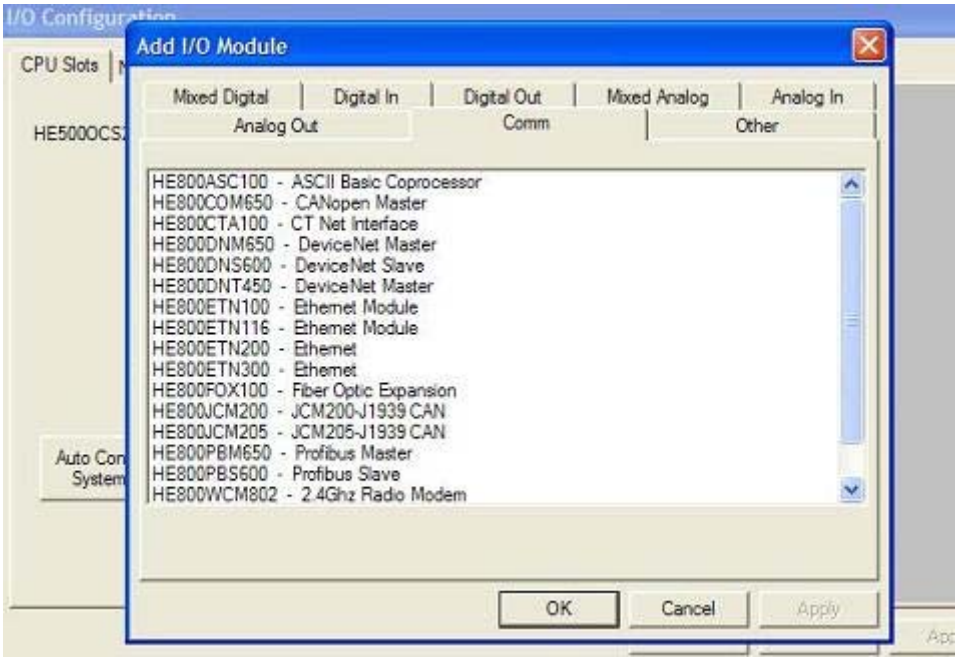


Figure 3.3

The selected module is now visibly attached to the main unit and can be configured.

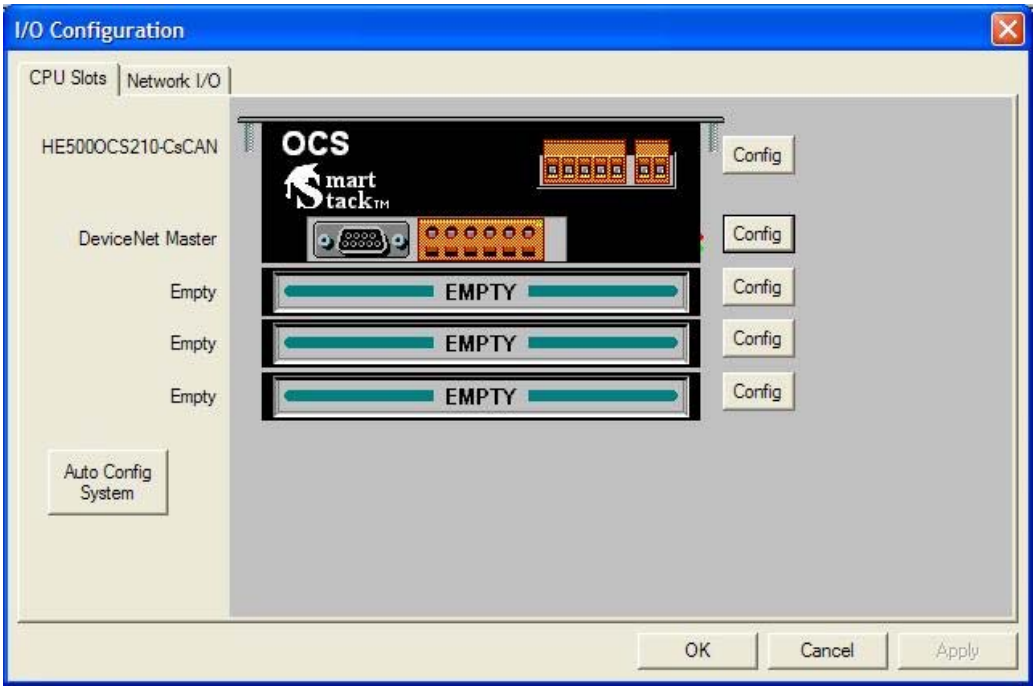


Figure 3.4

Select the CONFIG button adjacent to the module. Then select the MODULE SETUP tab.

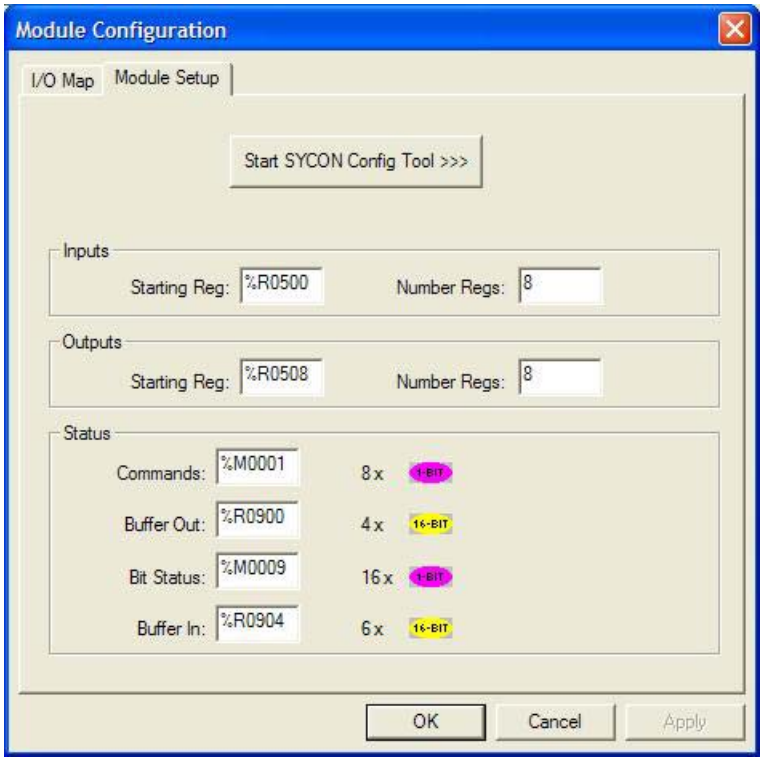


Figure 3.5

Configure the Inputs and Outputs.

NOTE:

INPUTS: means data coming FROM the Network VIA the DNM/DNS Module to the OCS Registers.

OUTPUTS: means data going TO the NETWORK VIA the DNM/DNS Module from the OCS Registers.

In Figure 3.5 above, For both Inputs and Outputs, 8 %R registers are used. The OCS %R registers are retentive, general purpose, 16 bit registers.

It is VERY important that the number of registers used for both Inputs and Outputs in Cscape is identical to the number setup in the Hsycon software when setting up the DNM650 and DNS600 modules. See Figure 3.6 below.

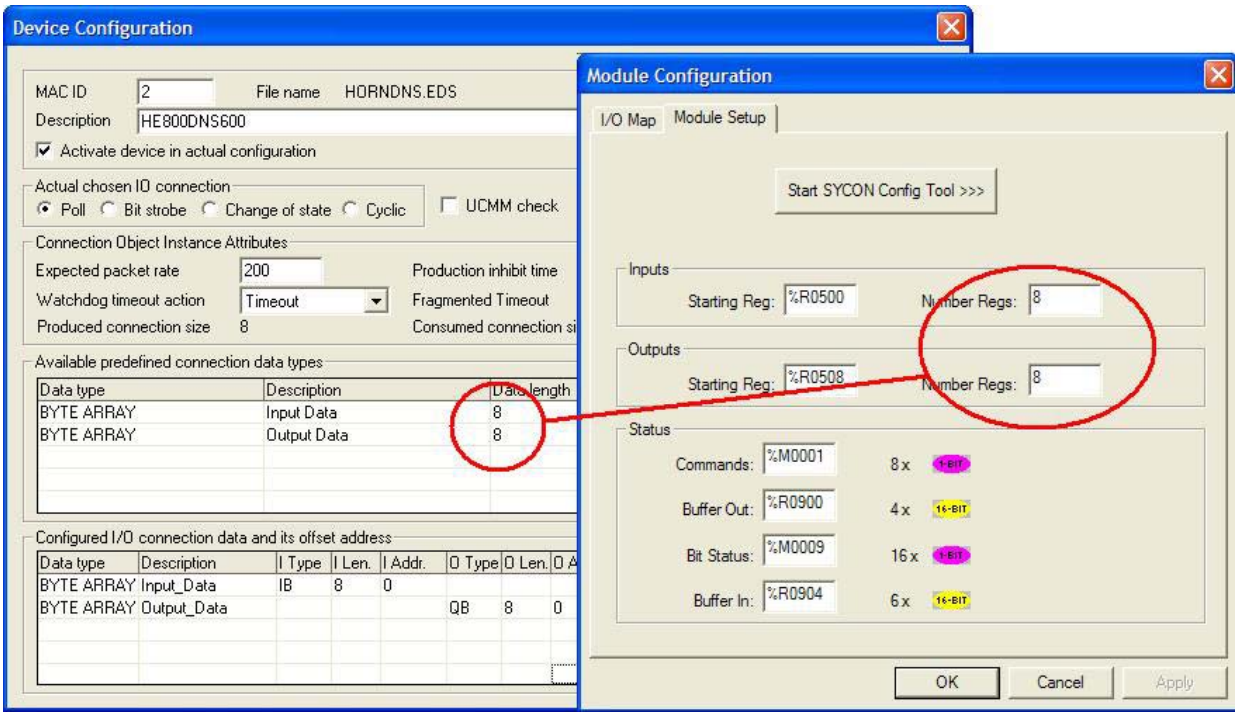


Figure 3.6

3.3 Configuring a SmartStack DeviceNet Master with any DeviceNet Slave

Table 4 below describes the steps to configure a Horner Smartstack DeviceNet Master to any DeviceNet Device (Slave).

Action	Menu in the System Configurator
• Create a new project	**File > New > DeviceNet**
• Copy EDS file of the DeviceNet device (Slave), if the device is not in the selection list.	**File > Copy EDS**
• Choose Horner DeviceNet Master and provide MAC ID.	**Insert > Master**
• Choose DeviceNet device and provide MAC ID address	**Insert > Device**
• Assign the input and output modules **(*1)**	Mark the Device (left Mouse click), then
• Assign the offset addresses	**Settings > Device Configuration**
• Assign the Device Parameter data, if the Device needs Parameter data	Mark the Device (left Mouse click), then **Settings > Device Configuration > Parameter Data**
• Set the Bus parameter	Mark the Master (left Mouse click), then **Settings > Bus Parameters**
• Set device assignment if no automatic assignment has occurred	Mark the Master (left Mouse click), then **Settings > Device Assignment**
• Save project	**File > Save**
• Download	Mark the Master (left Mouse click), then **Online > Download**
• Live list	Mark the Master (left Mouse click), then **Online > Live List**
• Start Debugger	Mark the Master (left Mouse click), then **Online > Start Debug Mode**
• Device diagnostic	Mark the Slave (left Mouse click), then **Online > Device Diagnostic**
• Stop Debugger	**Online > Stop Debug Mode**
• Global Diagnostic	Mark the Master (left Mouse click), then **Online > Global State Field**
• Transfer user data: Write output, read input	Mark the Master (left Mouse click), then **Online > I/O Monitor**

Table 4: Steps to Configure DNM650 with any DeviceNet Slave.

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

3.4 Configuring a SmartStack DeviceNet Slave with any DeviceNet Master

Table 5 below describes the steps to configure a Horner DeviceNet Slave to any DeviceNet Master as it is typical for many cases.

Action	Menu in the System Configurator

• Create a new project	**File > New > DeviceNet**
• Choose DeviceNet Master **(*1)** and provide MAC ID address	**Insert > Master**
• Choose Horner DeviceNet device (Slave) and provide Mac address	**Insert > Device**
• Assign the input and output modules **(*2)**	Mark the Device (left Mouse click), then **Settings > Device Configuration**
• Set device assignment if no automatic assignment has occurred	Mark the Device (left Mouse click), then **Settings > Device Assignment**
• Save project	**File > Save**
• Download	Mark the Slave (left Mouse click), then **Online > Download**
• Device diagnostic	Mark the Slave (left Mouse click), then **Online > Device Diagnostic**
• Transfer user data: Write output, read input	Mark the Device (left Mouse click), then **Online > I/O Monitor**

Table 5: Steps to configure a DNS600 with any DeviceNet Master.

Note (*1): Insert a Horner SmartStack DeviceNet Master (HE800DNM650). This Master is a place holder and it is not necessary to match the connected Master.

Note (*2): The Offset addresses assigned in the Slave configuration are always related to the DeviceNet Master.

3.5 Configuring a Horner DeviceNet Master with a Horner DeviceNet Slave

Table 6 below describes the steps to configure a SmartStack DeviceNet Master to a Smartstack DeviceNet slave :

Action	Menu in the System Configurator
• Create a new project	**File > New > DeviceNet**
• Choose Horner DeviceNet Master and provide MAC ID address	**Insert > Master**
• Choose Horner DeviceNet Device (Slave) and provide MAC ID address	**Insert > Device**
• Assign the input and output modules	Mark the Device (left Mouse click), then
• Assign the offset addresses **(*1)**	**Settings > Device Configuration**
• Set the Bus parameter	Mark the Master (left Mouse click), then **Settings > Bus Parameters**
• Set device assignment for the Master if no automatic assignment has occurred	Mark the Master (left Mouse click), then **Settings > Device Assignment**
• Set device assignment for the Device (Slave) if no automatic assignment has occurred	Mark the Device (left Mouse click), then **Settings > Device Assignment**
• Save project	**File > Save**
• Download to the Master	Mark the Master (left Mouse click), then **Online > Download**
• Download to the Device (Slave)	Mark the Device (left Mouse click), then **Online > Download**
• Live list	Mark the Master (left Mouse click), then **Online > Live List**
• Start Debugger	Mark the Master (left Mouse click), then **Online > Start Debug Mode**
• Device diagnostic	Mark the Device (left Mouse click), then **Online > Device Diagnostic**
• Stop Debugger	**Online > Stop Debug Mode**
• Global Diagnostic	Mark the Master (left Mouse click), then **Online > Global State Field**
• Transfer user data: Write output, read input	Mark the Master (left Mouse click), then **Online > I/O Monitor**

Table 6: Steps to configure a DNM650 with DNS600's.

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

CHAPTER 4: CONFIGURATION OF DEVICENET WITH HSYCON

4.1 Setting up the DEVICENET Configuration

To create a new configuration, choose the **File > New** menu. This will offer a selection list of fieldbus systems. Choose **DEVICENET**. If only the DeviceNet 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 (Electronic Data Sheet Files) Introduction

An Electronic Data Sheet (EDS) provides information necessary to access and alter the configurable parameters of a device. An Electronic Data Sheet (EDS) is an external file that contains information about configurable attributes for the device, including object addresses of each parameter.
The application objects in a device represent the destination addresses for configuration data. These addresses are encoded in the EDS. The figure below shows a general block diagram of a sample EDS.

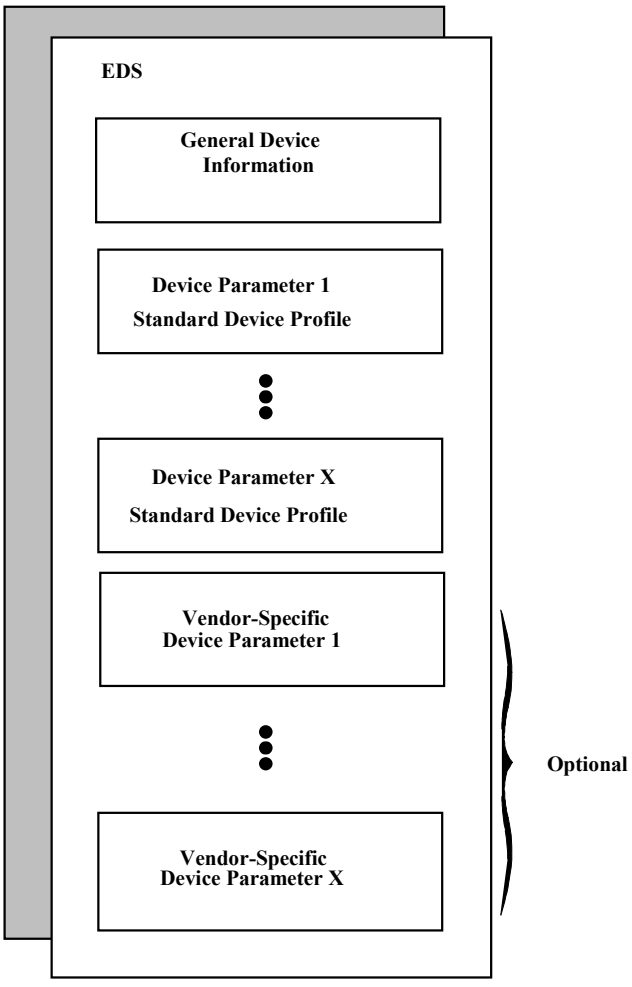


Figure 1: General block diagram of an EDS file

4.3 EDS Files and HSyCon

When HSyCon is started, it automatically retrieves all the EDS files stored in the EDS directory. The device names are placed in an internal list. During the configuration, the device-specific data is retrieved directly from the EDS files.
If a DeviceNet device does not appear in the selection list (Insert Master or Insert Device), the required EDS file may be copied into the EDS directory with **File > Copy EDS**. Another method is to copy the EDS file into the HSyCon EDS directory with Windows Explorer and then access the EDS files in the EDS directory with **Settings > Path** and **OK**.

Horner devices: The EDS files for Horner devices are included and already installed.
Other devices: The respective device manufacturer provides the EDS files for other devices.

The EDS files of some vendors are available on the DeviceNet homepage http://www.odva.org or visit the homepage of the manufacturer.
The EDS directory is adjustable. In order to alter the directory from a previous setting in another directory, use the menu **Settings > Path**. All EDS files must be placed in this directory.

4.4 Insert Master

To insert a Master into the configuration, choose the **Insert > Master** menu, this will open the selection window, or click on the symbol :

Insert > Master Symbol


Figure 1: Symbol Insert > Master

The mouse pointer automatically changes to the Insert Master pointer.

Insert Master Mouse Pointer


Figure 2: Mousepointer insert Master

Click on the position where the Master should be inserted. The dialog box from which one or more Masters can be chosen opens. The following types of Masters may be selected:

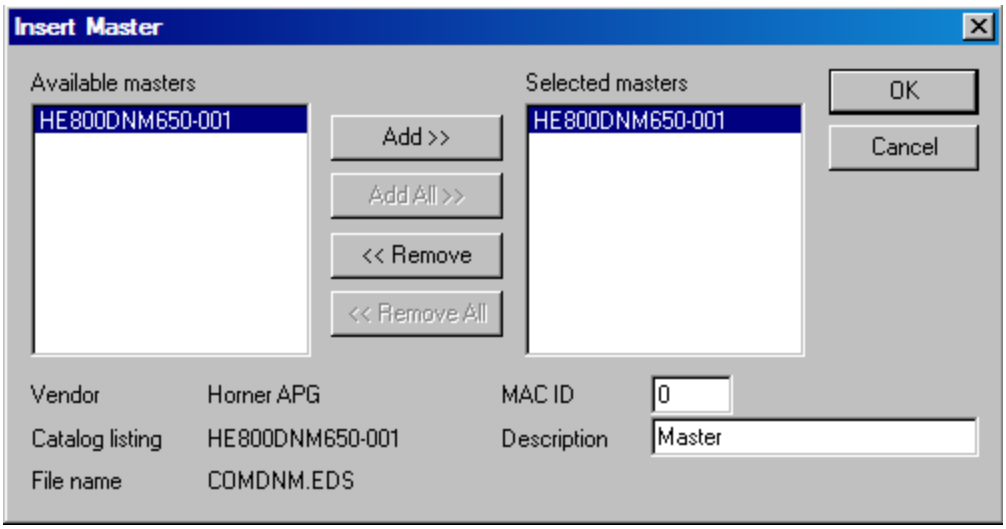


Figure 3: Selectable Master types

Select a Master device from the **Available Masters** list by clicking on it. By clicking the **Add** button the Master is shown in the list **Selected Masters**. Click the **OK** button, the Master will be inserted at the top of the configuration.
This example shows an HE500DNM650-001. The Master gets the description Master at first. This may be changed in the **Description** field. The **MAC ID** of the Master may also be changed here.

4.5 Insert Device (Slave)

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

Insert > Device Symbol


Table 4: Insert > Device Symbol

The mouse cursor changes automatically to the insert device cursor.

Insert Device Mouse Pointer


Table 5: Insert Device Mouse Pointer

Click on the position at which the new device should be inserted. A dialogue box appears from which one or more devices may be chosen.

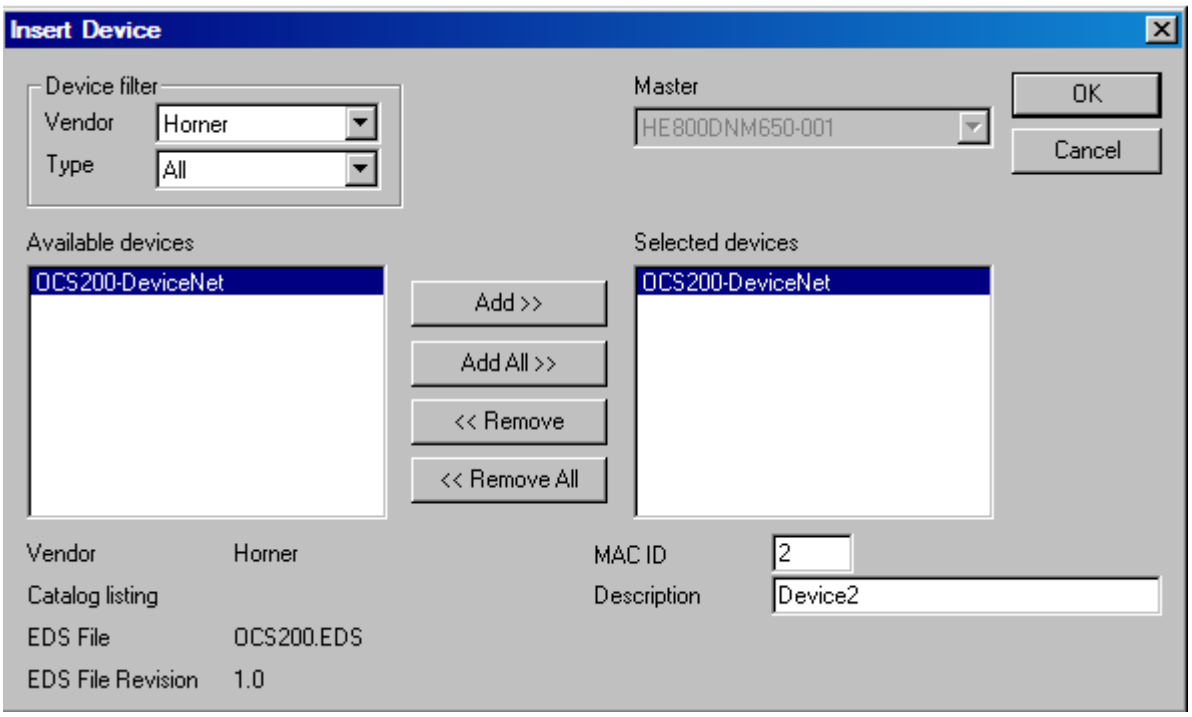


Figure 2: Insert > Device (Slave)

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

All devices in the right-hand list are assigned to the current insert point that is also shown in this window. If the Slaves in the right-hand list are chosen one after the other (a mouse click), then every Slave can be assigned a name in the **Description** field.
With each new selected device in the right list the MAC ID is incremented by one but it can be changed by the user in the field **MAC ID**.

Note: It is possible to select the same device more than once, however, each device must have a unique MAC ID to distinguish it in the network.

4.6 Replace Slave

To replace an existing Slave device first mark the device to be replaced. Then proceed as described in the section 4.5 Replace Master.

4.7 Device Configuration

To enter the Device Configuration set the focus on the device (left mouse click) and select the menu **Settings > Device Configuration** or set the focus on the device (left mouse click) and use the right mouse button at the device or simply double-click on the device.
The device's I/O are assigned to logical addresses in the process data image of the Master in the device configuration. Note that the device offsets set here will be used in the application to read inputs and write outputs.

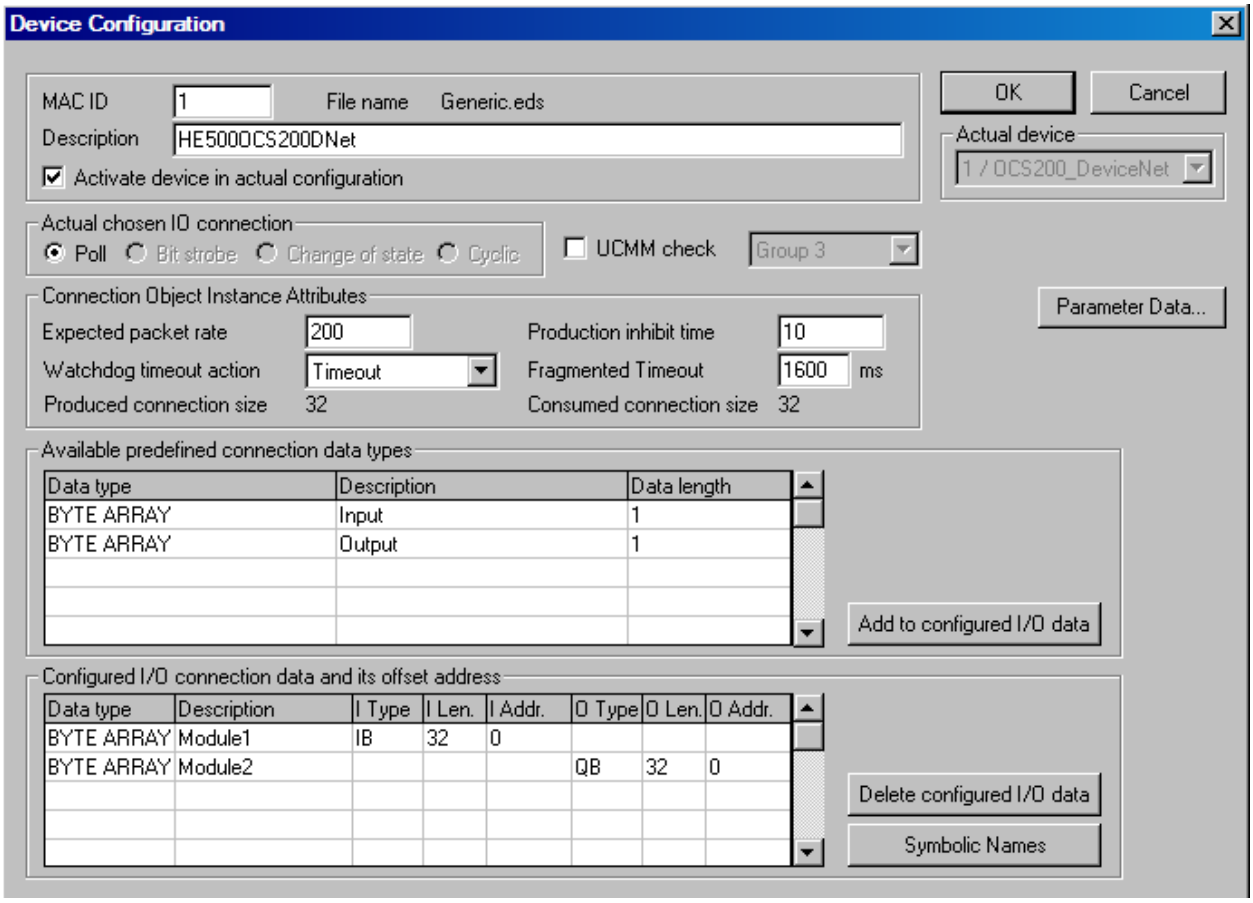


Figure 3: Settings > Device Configuration

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

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

The **File Name** displayed is the EDS file for the device.
The **Description** and **MAC ID** fields display the entries made during the selection phase of the Slave device. Both entries can be set/changed here.
The checkbox **Activate Device in actual configuration** decides, whether the Master tries to establish the communication with the Node or not. If a Node is physically not present in the network but will be present in future then the checkbox should not be checked. This suppresses unnecessary requests by the Master to devices that do not exist, but the device insertion reserves process data in the process data image of the Master.

4.8 MAC ID (Device network address)

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

4.9 Actual chosen IO Connection

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

I/O connection
Poll
Bit Strobe
Change of State
Cyclic

Table 6: Overview I/O Connections

The different connections types are :

- **Polled I/O Connection** - One poll command from the Master sends a number of output data to a single, specific device (point-to-point). The device receives (consumes) the poll command and processes the output data. If it has input data configured for this poll connection it reacts by sending (producing) back a number of input data and/or status information to the Master. Before a polled I/O connection is initiated by the Master, it reads the **Consumed** and **Produced Connection Size** of the data from the Slave first and compares each value with the internally configured one. If the Master detects differences the connection cannot be established. Sending a poll command can happen at any time the Master wants to and has timer or event dependencies. A device has to respond if it has consumed and understood the poll command request of the Master, even if it has no input data. Otherwise the Master will report a timeout error. Polling data to many devices has the disadvantage that the network traffic rate is very high and most data which is transferred has not changed since the last transmission. Furthermore the higher the bus load more communication errors can occur if the bus is disturbed by external influences.
- **Bit Strobe I/O Connection** - Bit strobe command and response messages rapidly move small amounts of I/O data between the Master device and one/some/all Slave devices. The bit strobe message contains a bit string of 64 bits of output data, one output bit per possible device. Each bit is assigned to one device address in the network. This service has broadcast functionality that means more than one Slave device can be addressed by one command. Because all addressed Slave devices get this command at the same time, this command is normally used to synchronize data transfer to several Slave devices. A Slave device can take its corresponding output bit as a real output information to give it to the peripheral connections (e.g. an LED) and/or use the bit as a trigger to send back its input data with a poll response message. The data that can be send back from each Slave after a bit strobe command was received is limited to 8 bytes in length. Bit strobe connections reduce the bus loading.
- **Change of State/Cyclic I/O Connection** - The Master device sends a number of output data to a single, specific device (point-to-point). Data production is triggered by either a determined changed value in the output data or the cyclic timer expiration. Depending on how the Slave behaviour is configured, the Slave can send back an acknowledge message, containing a number of input data and/or status information. The Slave device sends a number of input data to the Master, if the data is either changed or the cyclic timer has expired. The Master itself can acknowledge this message with output data if configured.
- Change of state only production of data hold down the bus load as small as possible, while data than can be transmitted as fast as possible by each device because bus conflicts are less possible. High performance data transmission can be achieved with comparatively low baud rates.

4.10 Connection Object Instance Attributes

The **Production Inhibit Time**, one for each connection, configures the minimum delay time between new data production in multiples of a millisecond. The timer is reloaded each time new data production through the established connection occurs. While the timer is running the device suppresses new data production until the timer has expired. This method prevents that the device is overloaded with to fast incoming requests.
The value 0 defines no inhibit time and data production can and will be done as fast as possible. If in polled mode for example a **Production Inhibit Time** of 1000dec is configured, then the poll request message to the device will be sent every second.
The **Expected Packet Rate**, one for each connection, is always transferred to the device before starting and doing the I/O transfer . The value is used by the device later to reload its 'Transmission Trigger' and 'Watchdog Timer'. The 'Transmission Trigger Timer' is used in a 'cyclic' I/O connection to control the time when the data shall be produced. Expiration of this timer then is an indication that the associated connection must transmit the corresponding I/O message. In 'change of state' connections the timer is used to avoid the watchdog timeout in this connection, when a production has not occurred since the timer was activated or reloaded.

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

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

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

4.11 UCMM Check

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

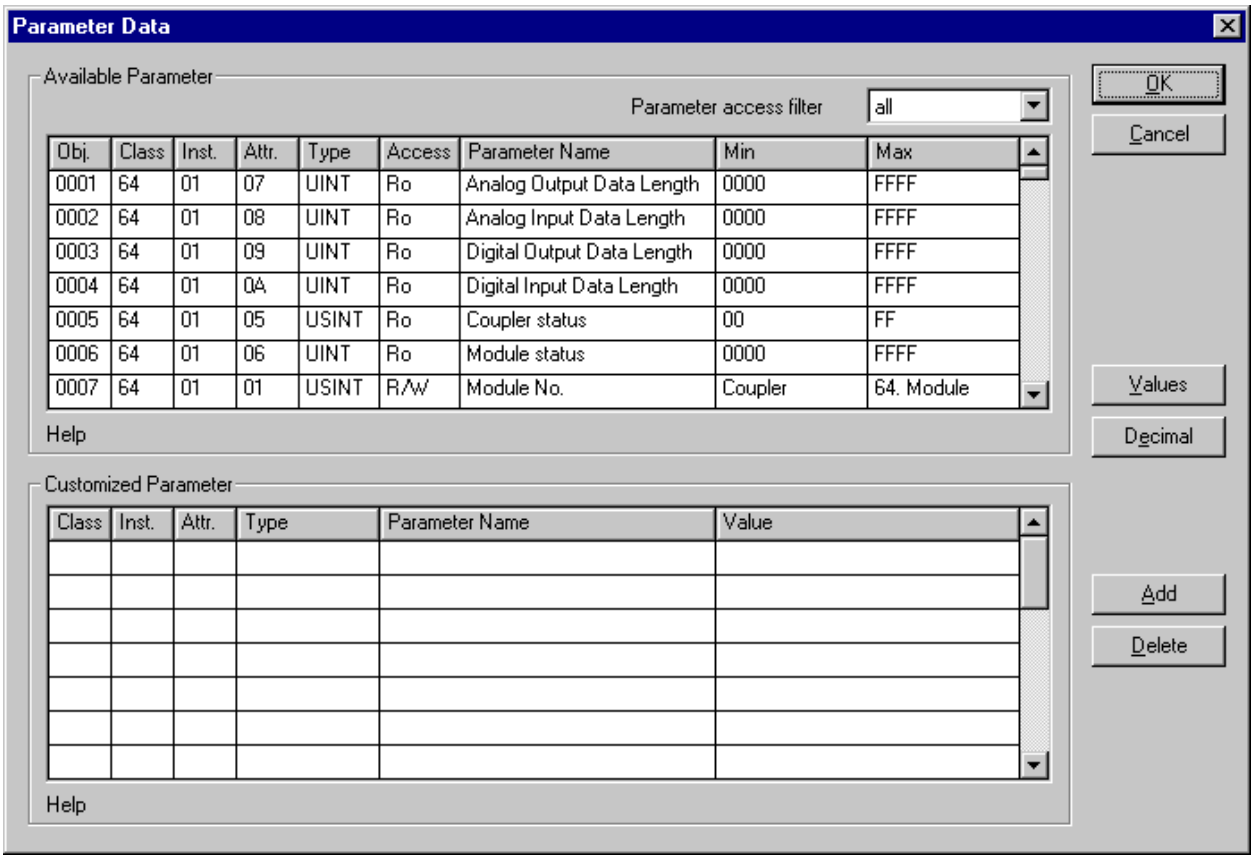
4.12 Fragmented Timeout

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

4.13 Parameter Data

The button **Parameter Data** can be selected in the Device Configuration window to edit the parameter data.
If default parameters are configured in the EDS file for this Node, they are inserted automatically when the menu is chosen the first time.

Some of devices need some further parameterisation data, to change for example a measurement limitation or a value range. These data is Node specific and their functionality can not be explained at this point.
The explanation can be normally found in the corresponding Node manual.
This window below shows an example of parameter data of a device.



Obj.	Class	Inst.	Attr.	Type	Access	Parameter Name	Min	Max
0001	64	01	07	UINT	Ro	Analog Output Data Length	0000	FFFF
0002	64	01	08	UINT	Ro	Analog Input Data Length	0000	FFFF
0003	64	01	09	UINT	Ro	Digital Output Data Length	0000	FFFF
0004	64	01	0A	UINT	Ro	Digital Input Data Length	0000	FFFF
0005	64	01	05	USINT	Ro	Coupler status	00	FF
0006	64	01	06	UINT	Ro	Module status	0000	FFFF
0007	64	01	01	USINT	R/W	Module No.	Coupler	64. Module

Class	Inst.	Attr.	Type	Parameter Name	Value

Figure 4: Settings > Device Configuration > Parameter Data

Two tables are available: one table with all available parameters and one table for customized parameters. These parameters can be selected from the available parameters to that table.

4.14 Process Data Configuration

- Fixed I/O data transferred

  DeviceNet handles I/O data transparent as a byte string without defining any data type in the transferred data. To be operative it defines only the number of bytes in consumed and produced direction that shall be transferred across a connection, nothing else. But HSyCon and the firmware now allows to assign modular each byte or a bunch of bytes of the transparent string to different data types. A list of the supported data types of the connection can be found in the middle table of the window called **Available Predefined Connection Data Types**.

The following data types are supported:

- Bit, Byte, Word, Dword, Byte Array

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

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

- Assigning the process data offset addresses

  The I/O offset addresses of each placed data type in the connection data table can be freely configured in a range of 0 to 3583 or they are set automatically by HSyCon. To enable or disable free configuration use the flag **Auto Addressing** in the menu **Settings - Auto Addressing**. If enabled HSyCon will place all configured I/O data, without spaces in physical order one after another based on the rising MAC-ID order. This is done during the download procedure. The assigned addresses can be checked then in the overview **Address Table** of the menu **View**. If the addresses are entered manually the default address 0 in the **input address** respectively the **output address** must be overwritten. Depending on the **Addressing mode** in the **DNM Master Settings** the addresses are byte addresses or word addresses. This is described in the chapter Addressing mode.

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

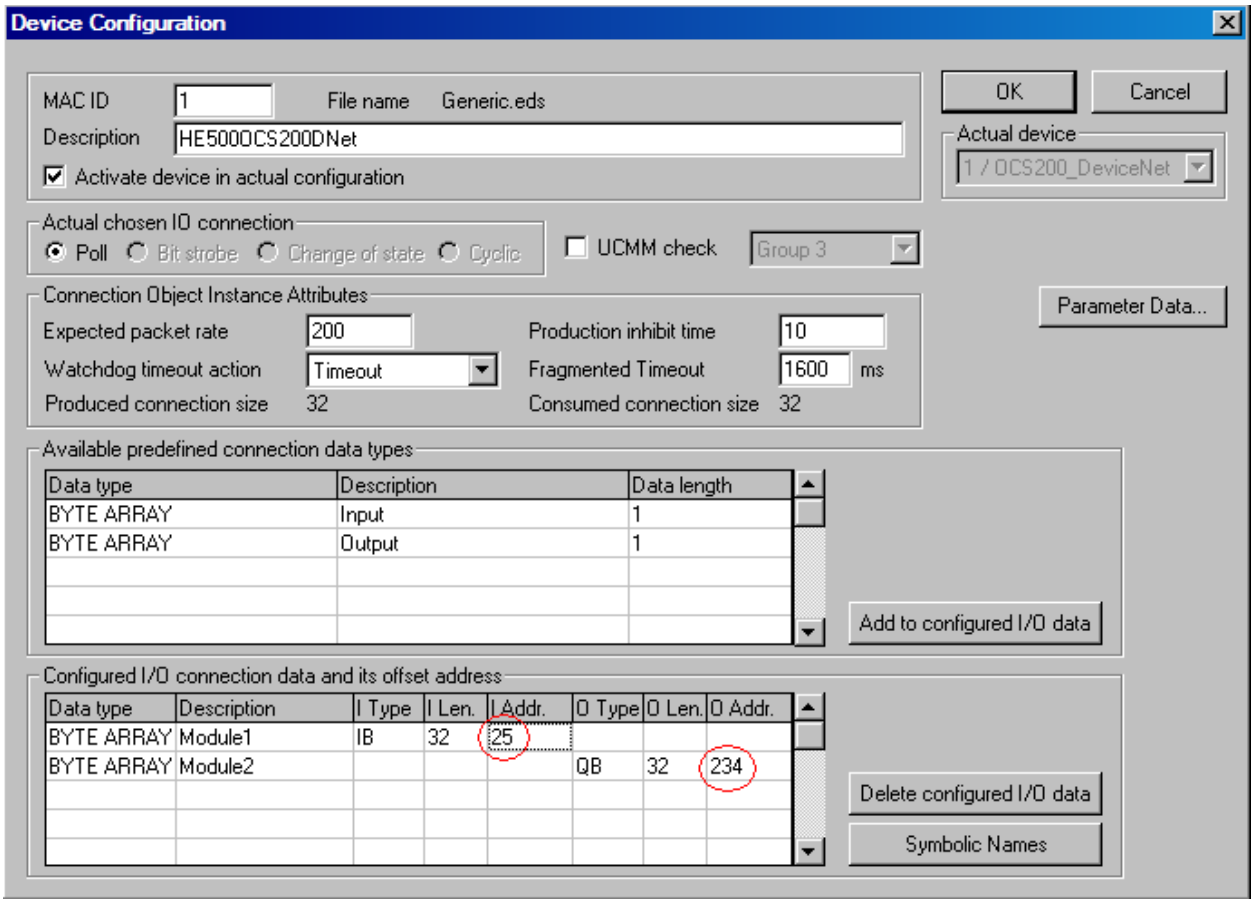


Figure 5: Settings > Device Configuration

In the column **I Addr** and **O Addr,** the addresses where to locate the data in the process image must be assigned. Remember that these addresses correspond to the application on the HOST side.
If a device is deactivated in the actual configuration the device is shown like this:

Activating or deactivating a device in a configuration can be very useful for devices that don't exist in the real physical network. The I/O offset addresses will be reserved or simply a symbolic 'missing device' waiting to be inserted as long the device is not connected.

CHAPTER 5: SETTINGS

5.1 Device Assignment

The Device Assignment setting determines how the System Configurator communicates with the device. This is set in the device arrangement via the menu **Settings > Device Assignment**. The following possibilities are available:

CIF Device Driver	CIF Serial Driver	CIF TCP/IP Driver

CIF Device Driver:

- Not supported do not choose this driver.

CIF Serial Driver:

- CIF Serial Driver: The HSycon Configurator communicates with the SmartStack device over a serial connection. In this case a COM interface of the PC must be connected via a cable (straight) with the diagnostic interface of the SmartStack device. The cable is standard Horner Programming cable.

CIF TCP/IP Driver:

- Not supported. Do not choose this driver.

5.2 COM Serial Driver

The serial driver supports COM1 to COM 4, in order to communicate via the diagnostic interface with the device. The Device is selected via **Settings > Device Assignment.**

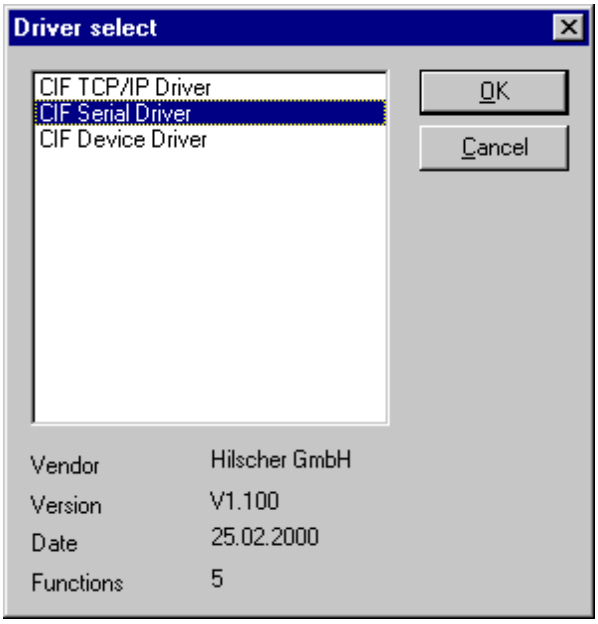


Figure 6: Driver selection > CIF Serial Driver

Choose the **CIF Serial Driver** and then **OK**, in order to select the CIF Serial Driver.

The connection must first be established using the switching surface **Connect COM1, Connect COM2, 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 interface and polls the Firmware of the device. A display of the Firmware will indicate when a device is connected. In the other case, a Timeout error (-51) appears, which will state that no device is connected.

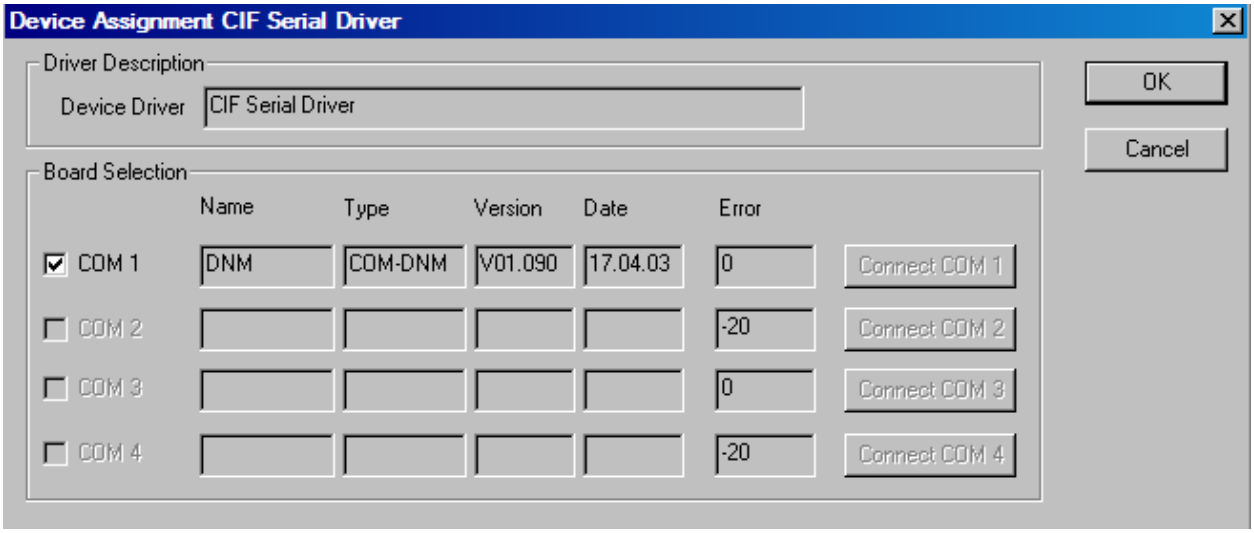


Figure 7: CIF Serial Driver > Device Assignment

The error number –20 indicates that this COM interface is not available or free.

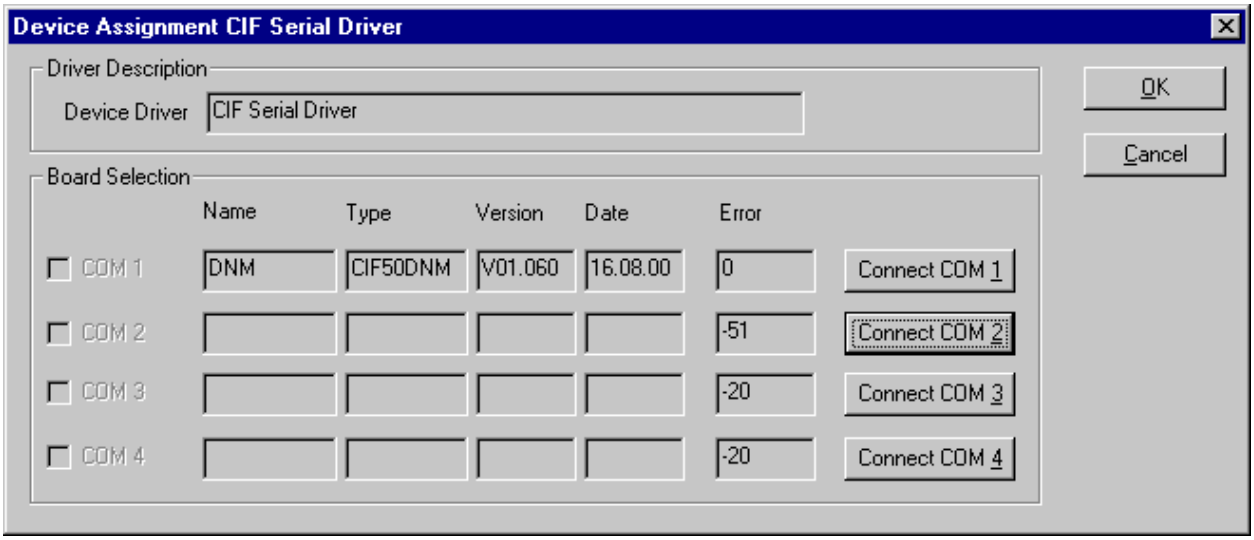


Figure 8: CIF Serial Driver > Device Assignment

5.3 Bus Parameter

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

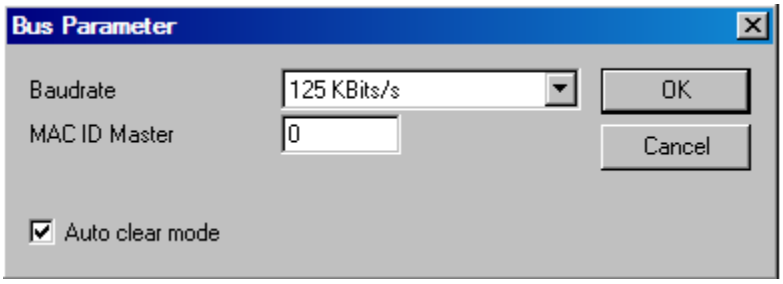


Figure 9: Settings > Bus Parameters

Mainly, this concerns the determination of the **Baudrate**. The DeviceNet board supports the baudrates 125kbit/s, 250kbit/s and 500kbit/s. Normally DeviceNet components use the autobaud detection to get the baudrate automatically. Furthermore, the **MAC ID** for the Master can be assigned in the Bus Parameter window.
The **Auto Clear mode** feature defines the behaviour of the Master if the communication breaks down or is interrupted to a Node. If the flag **Auto clear mode ON** is activated, then the Master will also stop the communication to all further Nodes which were still responding and active. If the flag Auto clear mode is not activated, then a lost communication contact to one Node has no influence on the communication channel of the still present ones. For all the error effected Nodes the Master remains in the state to try the reestablishment of the communication again.

5.4 DeviceNet Master

5.4.1. DeviceNet Master Settings

To enter the DeviceNet Master settings set the focus on the Master (left mouse click) and select the menu **Settings > Master Settings** or set the focus on the Master (left mouse click) and use the right mouse button at the DeviceNet Master device or simply doubleclick on the DeviceNet Master device.
The DeviceNet Master settings contain parameters which define the behaviour of the device on its user interface which is not a part of the DeviceNet configuration directly. This menu is only valid for Horner devices, and is downloaded with the DeviceNet configuration.

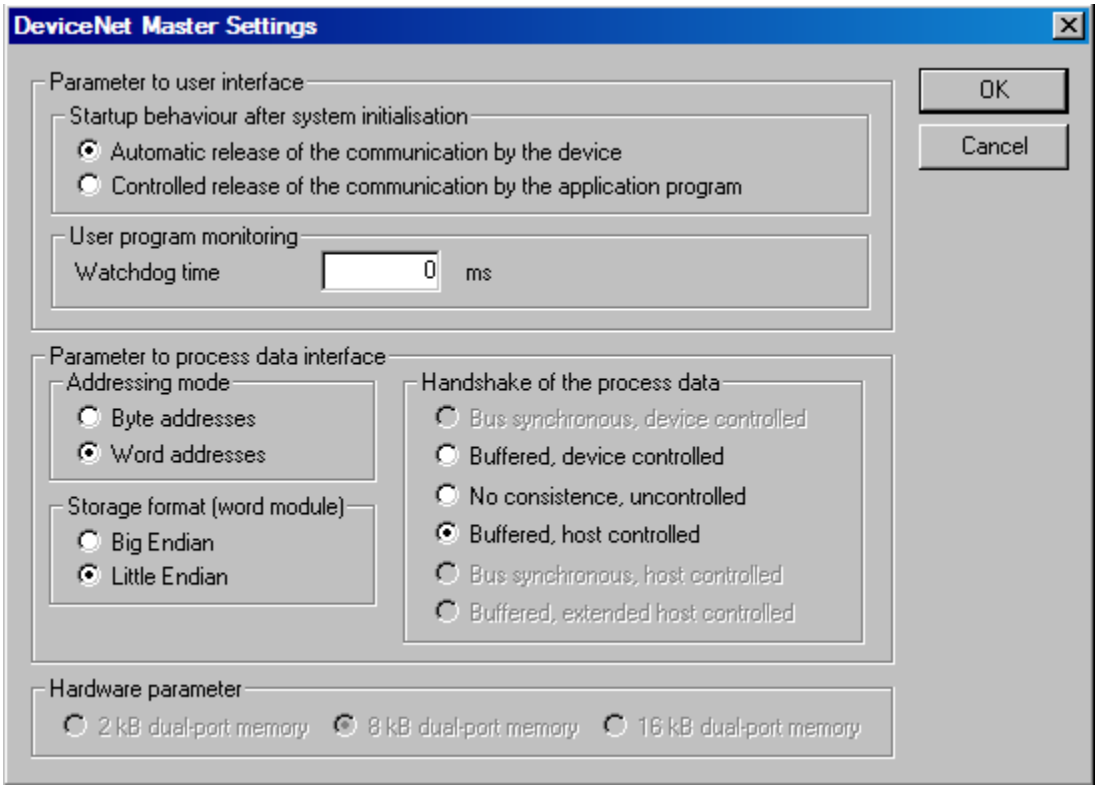


Figure 10: Settings > Master Settings

- **Startup behaviour after system initialization**

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

  Note: The HE800DNM650 must be set to Automatic release of the communication by the device

- **User program monitoring**

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

  Note: This is not a special DeviceNet function. Watchdog Time must be set to 0 for HE800DNM650

- **Addressing mode**

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

- **Storage format (word module)**

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

- **Handshake of the process data**

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

  NOTE: For HE800DNM650, this must be set to Buffered, host controlled

- **Hardware parameter**

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

5.4.2. Auto Addressing

With this Check Box it is possible to force the System Configurator to allocate the process data addresses in physical order itself or to enable the manual address configuration. Please see also Process Data Configuration on page 27.

5.4.2. Addressing Mode

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

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

Table 7: 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 address in Byte mode	IEC address in Word mode	Offset address in the Dual-port memory	Data in the Process image	Output to an I/O module
QB 0	QB 0	0	0000 0000	
QB 1		1	0000 0000	
QB 2	QB1	2	1110 0010	Output of QB2 / QB1 to a single Byte module: 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	QB2	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	QW3	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 0 1 1 1 1 1 1 1

Table 8: Example for data in the process data 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
	-	

Figure 11: 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
	-	

Figure 12: Image of the method of addressing for output

5.5 Device (Slave)

5.5.1. Device Configuration

To call up the menu Device Configuration, set the focus at the device (left mouse click) and select the menu **Settings > Device Configuration**
or
make a doubleclick on the device icon will open the **Device Configuration** window. The section Device Configuration on page 25 describes the configuration.

5.6 Project Information

When a project is created, the project information can be typed into the **Settings > Project Information** menu. This entry can then be read when this menu is called up.

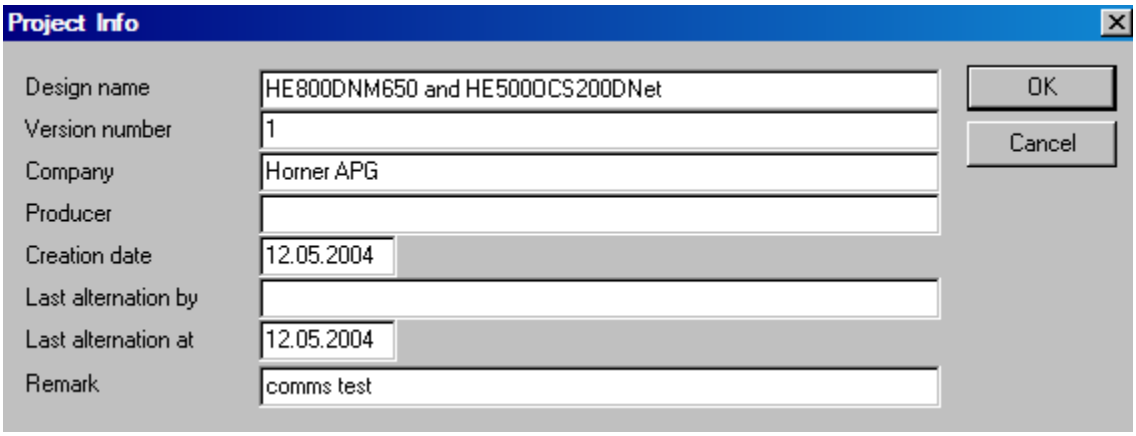


Figure 13: Settings > Project Information

Click the **OK** button to save the Project Information.

5.7 Path

In the menu **Settings > Path** the path directory of the EDS files is shown (EDS File directory). The default value is:
C:\Program Files\HornerAPG\HSyCon\Fieldbus\DEVNet\EDS
The path for Project directory defines the path where the project specific files are stored.

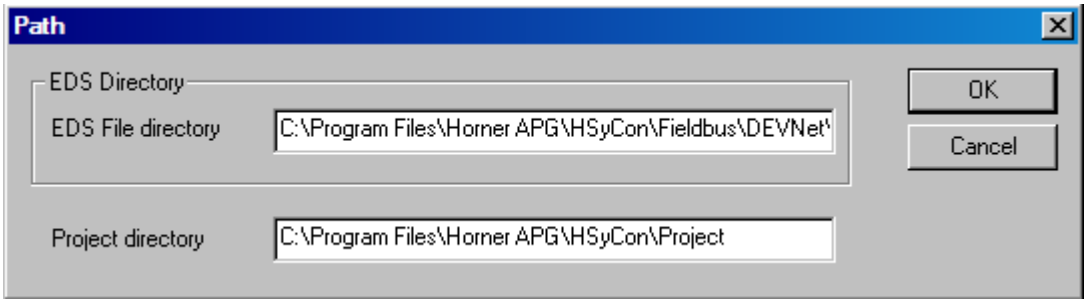


Figure 14: Settings > Path

Click the **OK** button to read in all EDS files.

5.8 Language

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

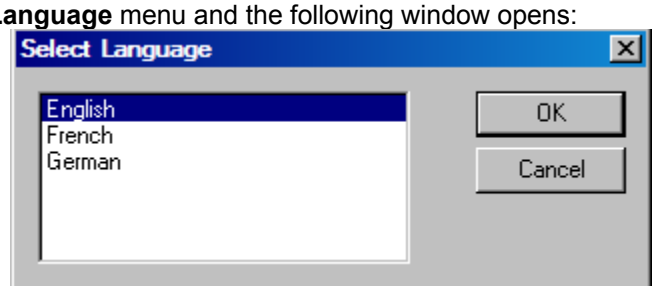


Figure 15: Settings > Language

The language of the System Configurator can be set. 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!

5.9 Start Options

After activating the **Settings > Start Options** menu point in the network mode (the network mode can be changed by choosing **Window > Logical Network View**) , the following dialog will appear.
Here it is possible to set the various starting options or modes. Some are of importance only for the OPC-Server operation.
The important ones are given below.

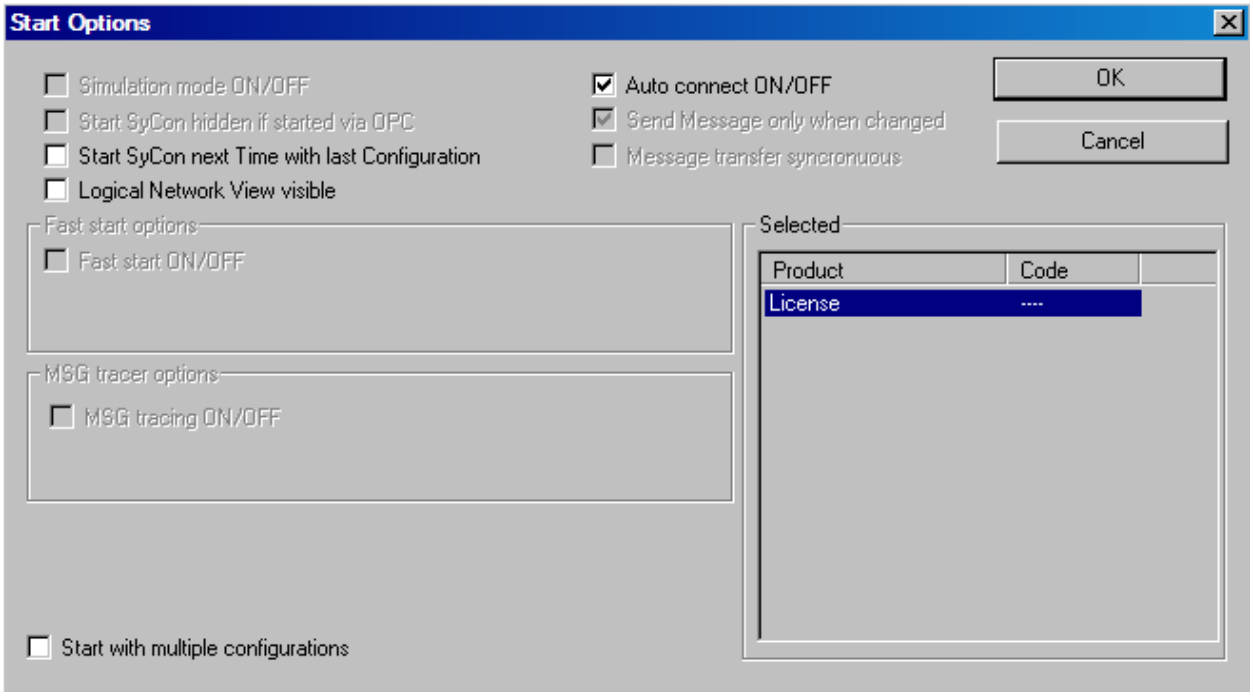


Figure 16: Settings > Start Options

- **Simulation mode ON/OFF**

  Only valid for the OPC Server.

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

  Only valid for the OPC Server.

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

  When this is selected the last saved configuration in the HSyCon is automatically loaded when the HSyCon is started again.

- **Logic Network View visible**

  When this is selected possibility of diverting to the network mode without having to install the HSyCon with OPC. It is also possible to use the Watch List from the network mode.

- **Fast start ON/OFF**

  Only valid for the OPC Server.

- **TAG tracing ON/OFF**

  Only valid for the OPC Server.

- **OPC tracing ON/OFF**

  Only valid for the OPC Server.

- **Start with multiple configurations**

  If this option is selected HSyCon can be started with up to four configurations simultaneously. The paths are shown in the window and they are changeable there.

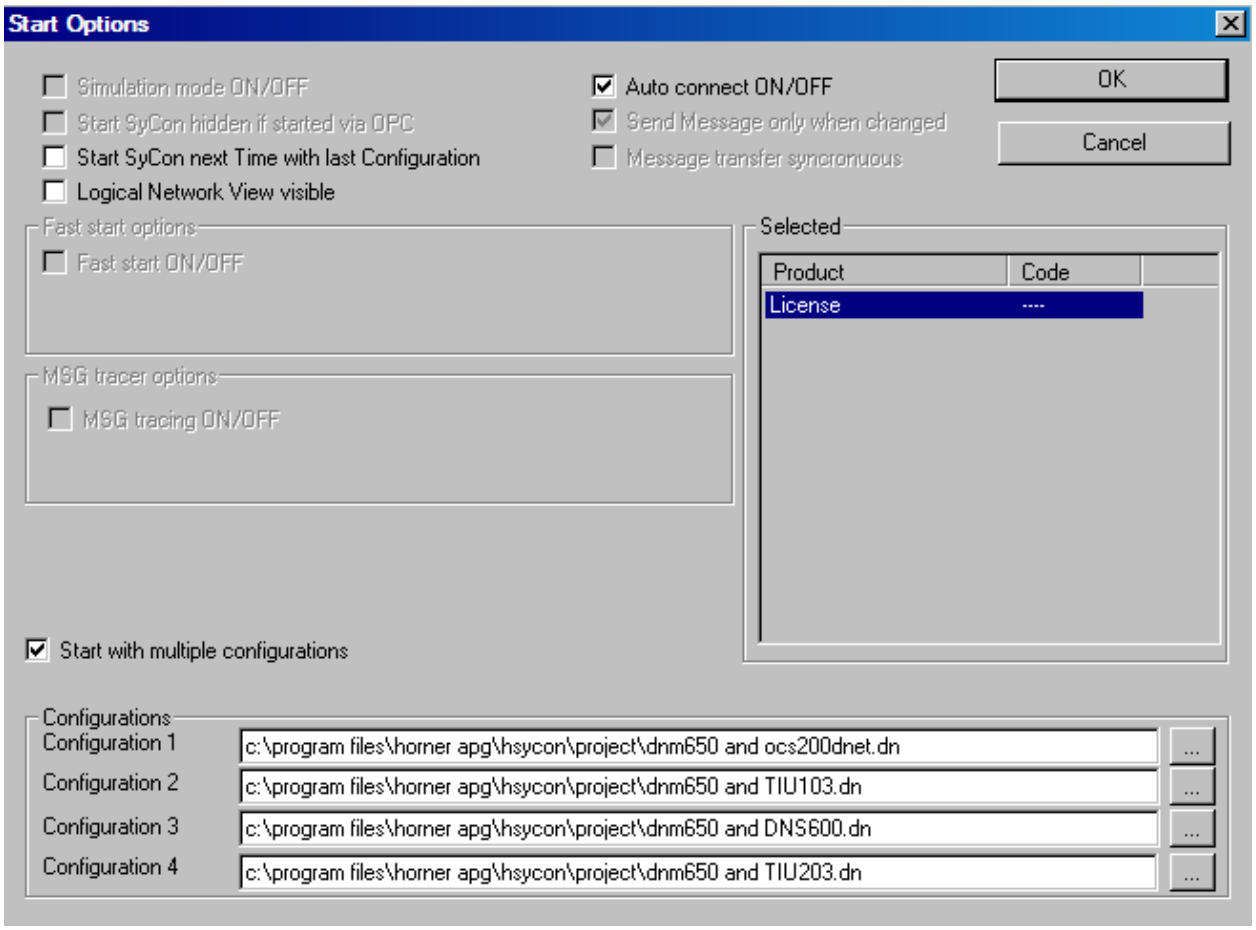


Figure 17: Settings > Start Options

CHAPTER 6: ONLINE FUNCTIONS

6.1. Introduction

In this section all the functions that directly affect the Horner DeviceNet Modules are described, e.g. HE800DNS600 and HE800DNM650.

Note: This will permit interruption of Devicenet communication and outputs can be switched ON or OFF.

6.2. Online to the Module

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

6.3. Downloading the Configuration

In order to release the configuration and network access, a transfer (Download) to the DNM/DNS devices must be carried out on the **Online > Download** menu. A warning will appear that the communication on the DeviceNet will be interrupted. This warning must be confirmed.

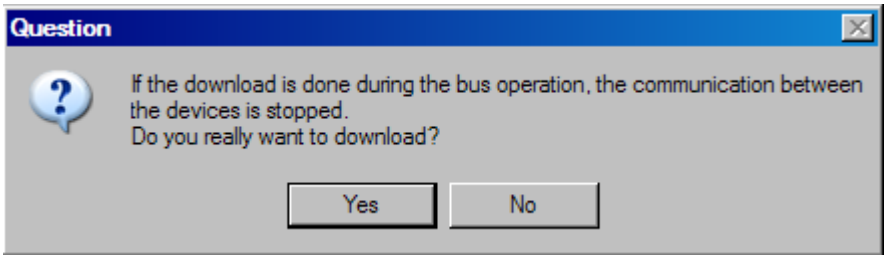


Figure 18: Security question before Download

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

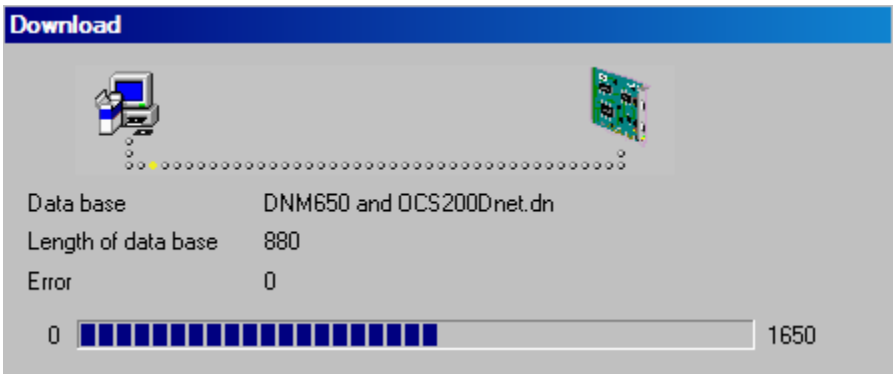


Figure 19: Online > Download

Before the download is executed, the configuration is tested by the Configurator. The most common cause of error is overlapping of addresses in the process data image. This can be checked by calling up the address table with the **View > Address Table** menu point.
If the issue of addresses in the process data image should be carried out automatically, then the **Auto Addressing** button in the **Master Configuration** window must be activated.
The configuration is transferred into the selected device and stored there in FLASH memory in a zero voltage manner so that the configuration is available when the voltage supply is switched off and on again.
After the download, the device carries out an internal restart and begins with the communication if in **DeviceNet Master Settings** the **Automatic Release of Communication by the Device** menu point has been set.

6.4. Firmware Download

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

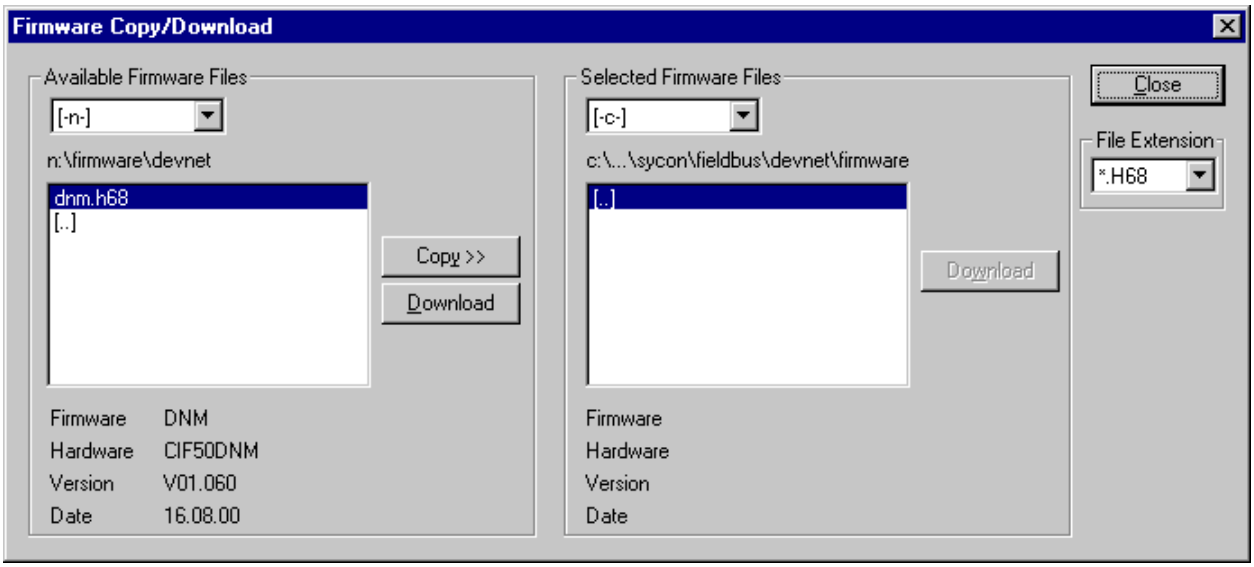


Figure 20: Online > Firmware Download

6.5. Firmware / Reset

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

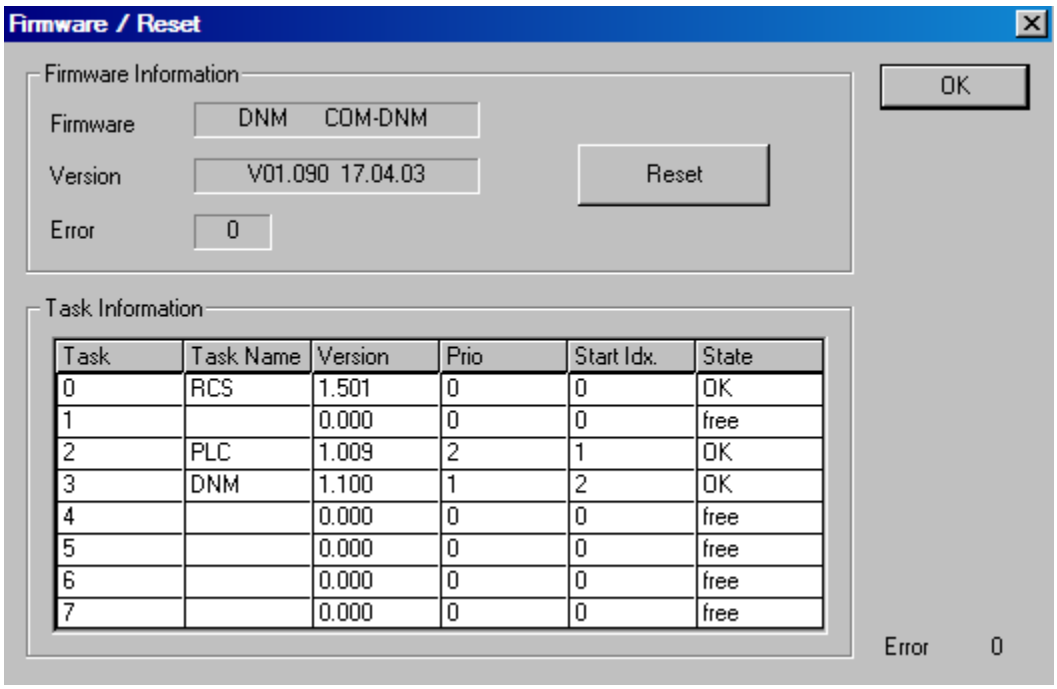


Figure 21: Online > Firmware / Reset

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

6.6. Device Info

First the desired device must be chosen with a left mouse click on the symbol of the device. Then select the **Online > Device Info** menu in order to obtain further information on the selected device.
The manufacturer date, the device number and the serial number of the device is retrieved and shown.

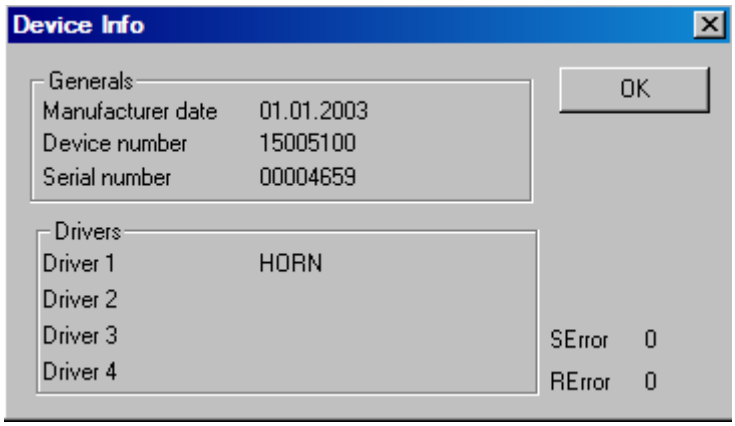


Figure 22: Online > Device Info

6.7. Automatic Network Scan

After the Master device is configured, it is possible to scan the DeviceNet network for other devices (automatic network scan). This allows a very fast configuration, and detailed parameters for these devices can be changed later.
To start an automatic network scan, please proceed as followed:

1. Create a new project: Select the menu **File > New** and **DeviceNet**.
2. Select a Master: The Master is selected with the menu **Insert > Master**
3. Click on **Settings > Bus parameters** and select the baudrate and the MAC ID from the Master (explained in section Bus Parameter on page 34)
4. Select **Online > Download** to load these settings into the DeviceNet Master.
5. Save: With the menu **File > Save** the so far set settings are saved.
6. Click on the Master and choose **Online > Automatic Network Scan**

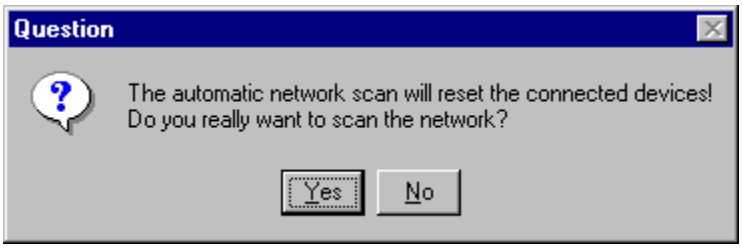


Figure 23: Online > Automatic Network Scan (Security question)

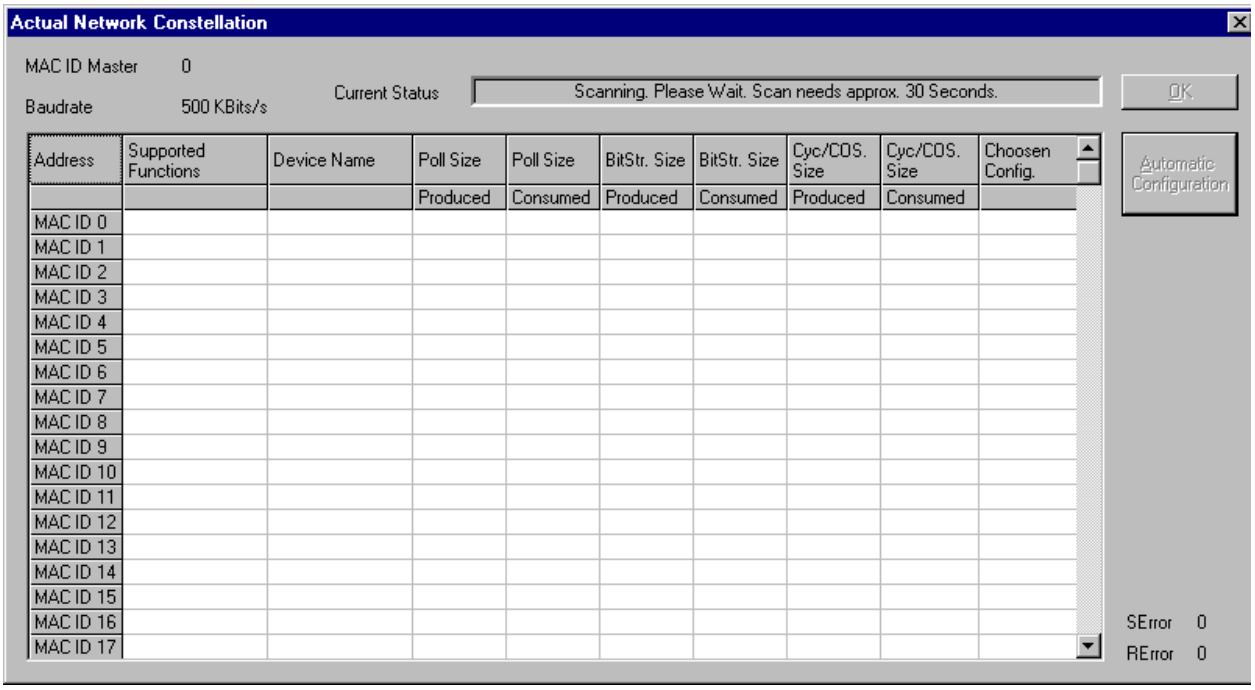


Figure 24: Online > Automatic Network Scan (during scan)

The network scan will take approx. 30 seconds. The network scan is still in progress and could not be interrupted until the status, shown in the field “Current Status”, is “Ready!”. When the scan is done, the devices found can be seen at the corresponding MAC ID address in the table.
The master reads the following objects from the DeviceNet device (Slave) during the scan:

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

Table 9: Readed Classes.Instance.Attribute during the network scan

Example:

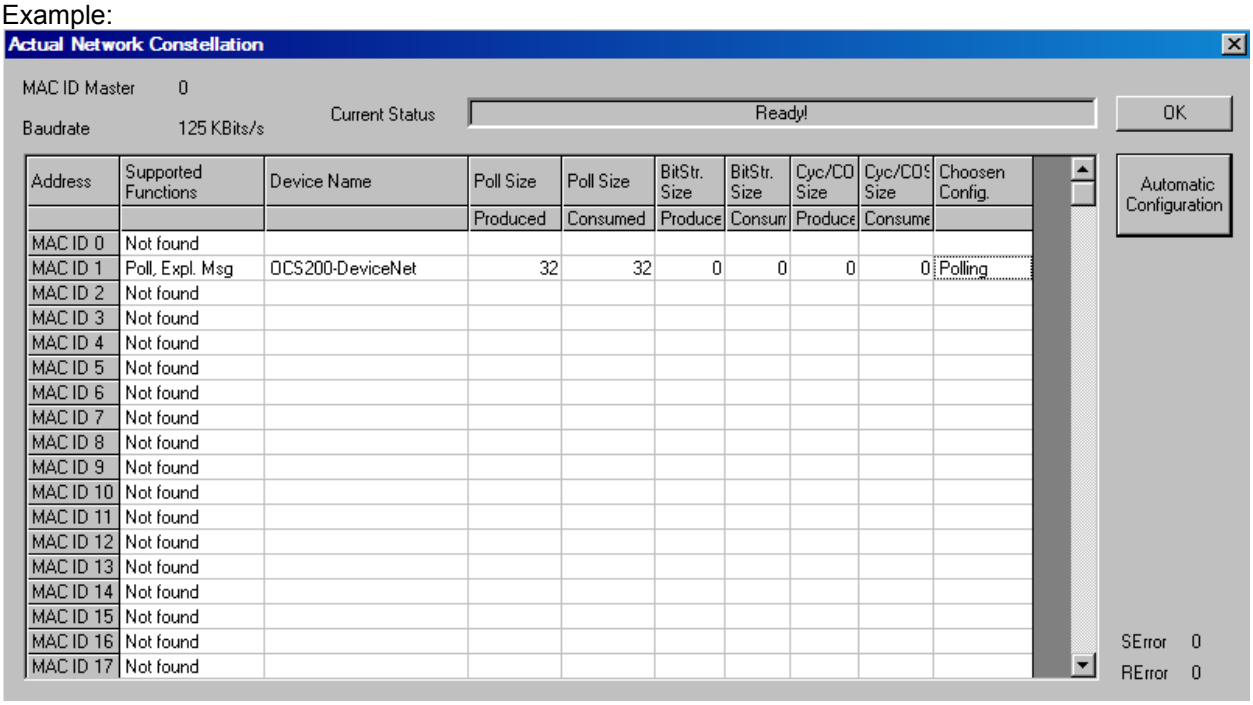


Figure 25: Online > Automatic Network Scan (after scan)

In our example in the figure above the automatic network scan detected a Horner device at MAC ID 1. Here is an explanation of the different columns:

Variable	Meaning
Supported functions	Functions supported by the device, could be polled, bit strobe or cyclic/change of state (see explanation in Actual chosen IO Connection page 27)
Device Name	Name of the device, result from network scan
Poll Size Produced	Number of data for poll connection (input)
Poll Size Consumed	Number of data for poll connection (output)
BitStr. Size Produced	Number of data for bit strobe connection (input)
BitStr. Size Consumed	Number of data for bit strobe connection (output)
Cyc/COS. Size Produced	Number of data for cyclic/COS connection (input)
Cyc/COS. Size Consumed	Number of data for cyclic/COS connection (output)
Chosen Config	Configuration chosen by the user, could be Change of State, Cyclic, Polling, Bit strobed or explicit only and depends on the functions supported by the device. Click on the cell to change the configuration.

Table 10: Meaning of the columns in the automatic network scan

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

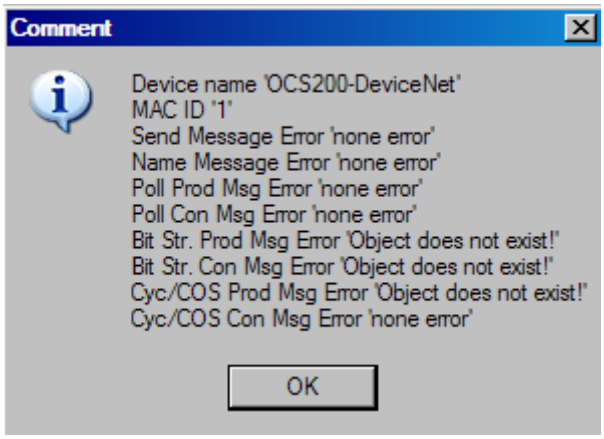


Figure 26: Information on a device in the automatic scan window

If this configuration is the required one, click on **Automatic Configuration** and select **Yes** when prompted. Afterwards the Automatic Configuration Window can be closed by clicking on **Close**. If the devices that were found in the configuration are not required, just click **Close**.
To manually insert the devices, please go on with section Insert Device (Slave) on page 24.

6.8. Start/Stop Communication

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

6.9. Diagnostic Functions

The following table shows Diagnostic Functions and their using for

- Horner DeviceNet Master devices
- Horner DeviceNet Slaves

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

Table 11: Overview Diagnostic Functions

6.10. Live List

Select the menu **Online > Live List** for an overview of all devices physically present in the actual network constellation. Present devices are drawn in black, all other non present devices are drawn in grey. The live list works online. If a station is connected or disconnected, the result can be seen as soon as HSyCon collects the latest live list from the Master board. Remember that all devices on DeviceNet have to proceed the autobaud detection phase first to get wholly run. This can take several milliseconds.

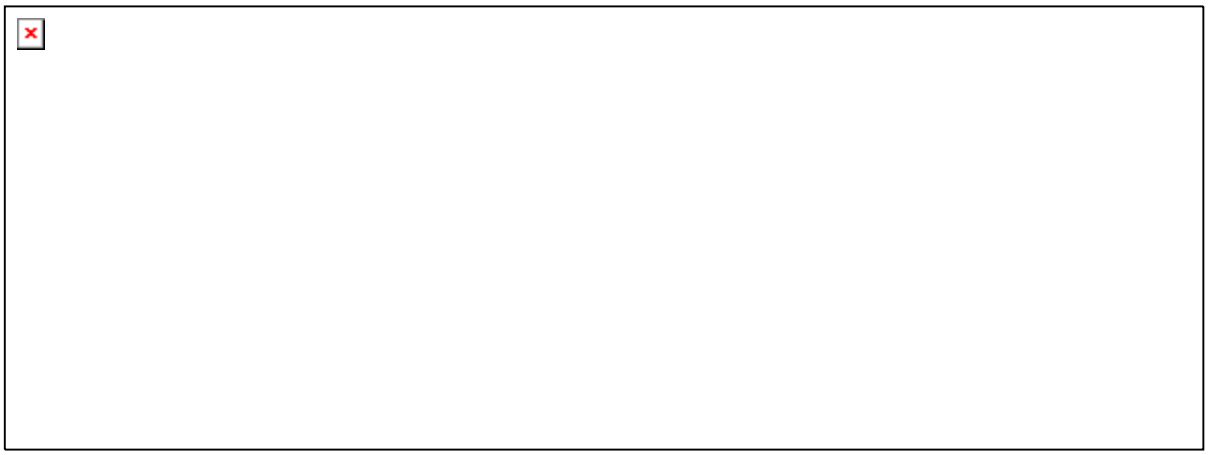


Figure 27: Online > Live List

6.11. Change MAC-ID

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

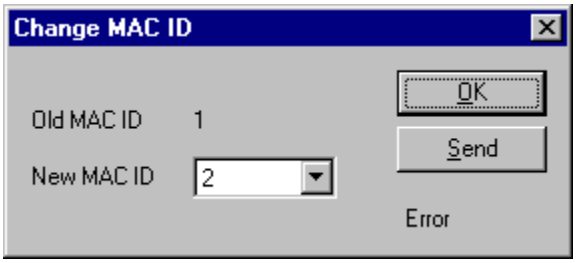


Figure 28: Online > Live List > Change MAC-ID

Select the new MAC-ID in the **New MAC-ID** field and click on the **Send** button to send it to the DeviceNet device (slave).
Note: The DeviceNet device (slave) has to support this function if this function is required.

6.12. Debug Mode

Select the Master device by clicking on it. Then select the menu item **Online > Start Debug Mode**. The System Configurator cyclically interrogates the status of the network communication on the CIF, COM or PKV and the individual condition of the devices.
To end the Debug Mode select the menu **Online > Stop Debug Mode**.

6.13. The Debug Window

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

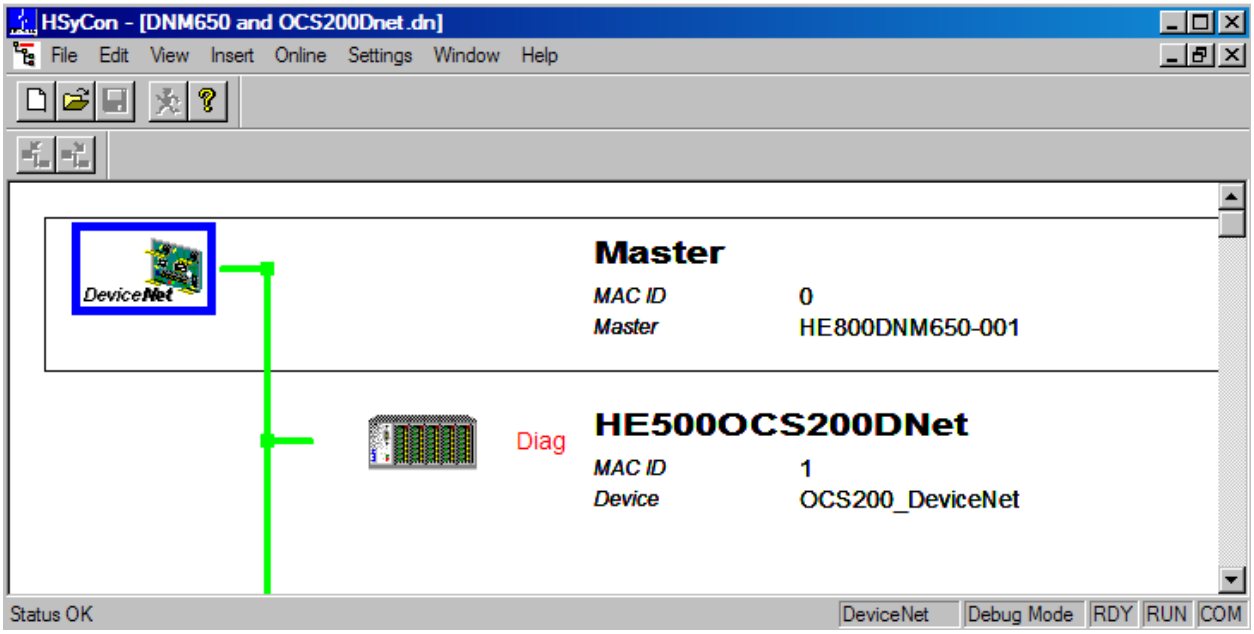


Figure 29: The Debug window

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

6.14. Device Diagnostic

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

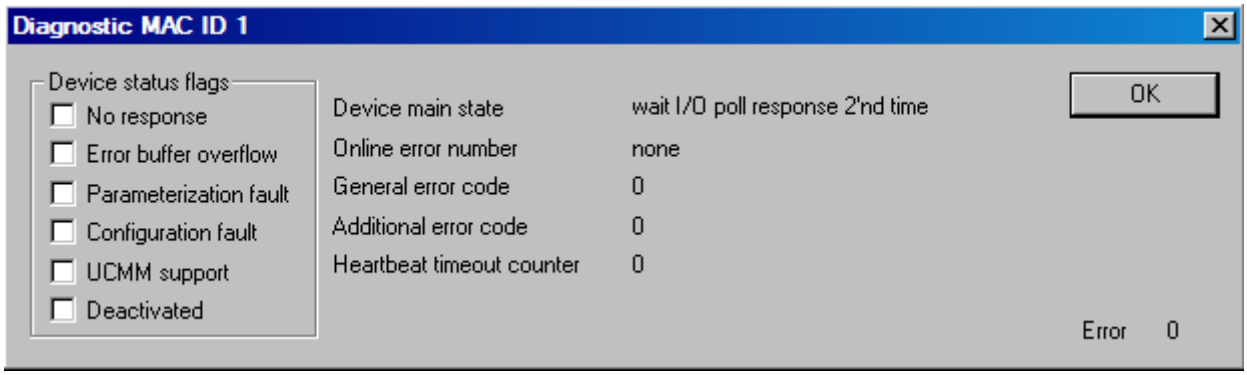


Figure 30: Online > Device Diagnostic

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

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

Table 12: Meaning of the bits in the Device Diagnostic

6.15. Global State Field

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

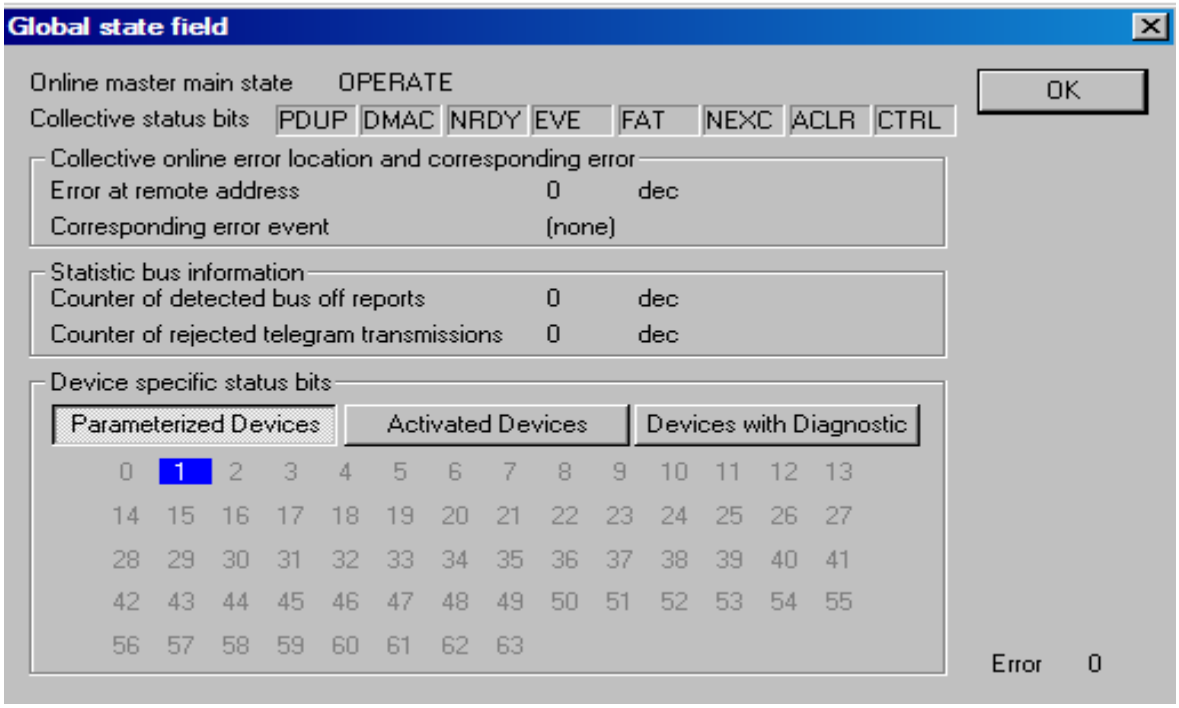


Figure 31: Online > Global State Field

The first row displays the main state of the Master. It can have the state **OPERATE** or **STOP**.
The next row display individual bus errors. A pending error is displayed with a red field. The meaning of the individual abbreviations is described in the following.

Status Bits	Meaning
PDUP	The device is involved in the duplicate MAC-ID check procedure, to check if other devices with the same address are connected to the network. The duplicate MAC-ID check will be finished, if at least one DeviceNet device could have been found connected to the network.
DMAC	The DEVICE has stopped the duplicate MAC-ID check procedure and found an other device having the same MAC-ID address. Change the DEVICE address to avoid this failure.
NRDY	HOST-NOT-READY-NOTIFICATION indicates if the host program has set its state to operative or not. If this bit is set the host program is not ready to communicate.
EVE	EVENT-ERROR The used 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 severe 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 The device stopped the communication to all Nodes and reached the auto-clear end state.
CTRL	CONTROL-ERROR Master parameterization error detected.

Table 13: Meaning of collecting status bits in the Global State Field

Further Information:
Collective online error location and corresponding error gives the address of the faulty station and the error in words.
Statistical bus information gives the number of detected bus errors and the rejected telegrams.
Device specific status bits: **Parameterized Devices, Activated Devices** and **Devices with Diagnostic** are shown when this button is clicked. The activated addresses are white numbers.
This application updates online the status in the global state field.
Diagnostics can be seen by double clicking at a highlighted station address of a device.

6.16. Device Diagnostic

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

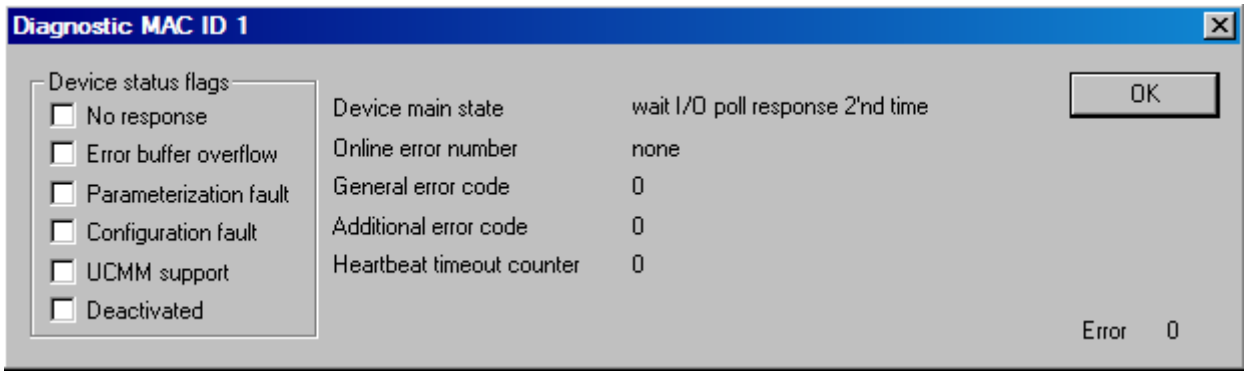


Figure 32: Online > Device Diagnostic

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

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

Table 14: Meaning of the bits in the Device Diagnostic

6.17. Extended Device Diagnostic

The Extended Device Diagnostic helps to find bus and configuration errors when the HSyCon menu functions are of no further help.
First the desired device must be chosen with a left mouse click on the symbol of the device. Then select the menu **Online > Extended Device Diagnostic**.
This menu opens a list of diagnostic structures. These contain online counters, statuses and parameters:

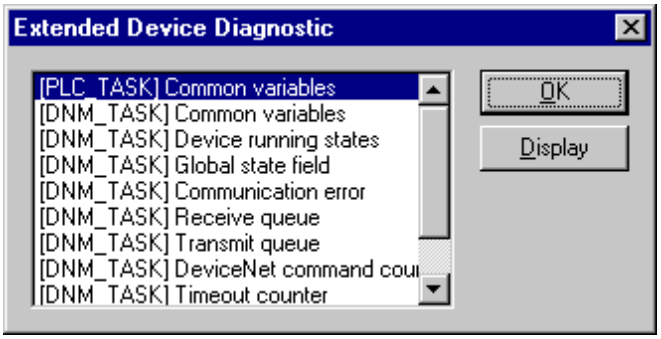


Figure 33: Online > Extended Device Diagnostic (Master)

6.18. Extended Device Diagnostic Master

PLC_TASK
DNM_TASK

Task / Task State	Page
PLC_TASK Common Variables	*88*
DNM_TASK Common Variables	*88*
DNM_TASK Device Running States	*90*
DNM_Task Communication Error	*90*
DNM_Task Receive Queue	*91*
DNM_Task Transmit Queue	*91*
DNM_Task DeviceNet Command Counters	*92*
DNM_Task Timeout Counter	*93*
DNM_Task Init Counter	*93*

Table 15: Extended Device Diagnostic Master

6.19. Extended Device Diagnostic Device (Slave)

PLC_TASK
DNS_TASK

Table 16: Extended Device Diagnostic Device (Slave)

6.20. User Data Transfer

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

- Horner DeviceNet Master devices
- Horner DeviceNet Slaves

User data transfer function	Usage	Usable with Horner DeviceNet Master devices	Usable with Horner DeviceNet Master 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	Yes
Get Device Attribute and *6.25**Set Device Attribute***	Get and Set attribute	Yes	No, only for Horner Master devices

Table 17: Overview User Data Transfer

Further test functions with user data transfer are available for

- Explicit Messaging for DeviceNet Master in section ***Message Monitor for testing explicit*** messaging of DeviceNet
- on page *62*.

6.21. I/O Monitor

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

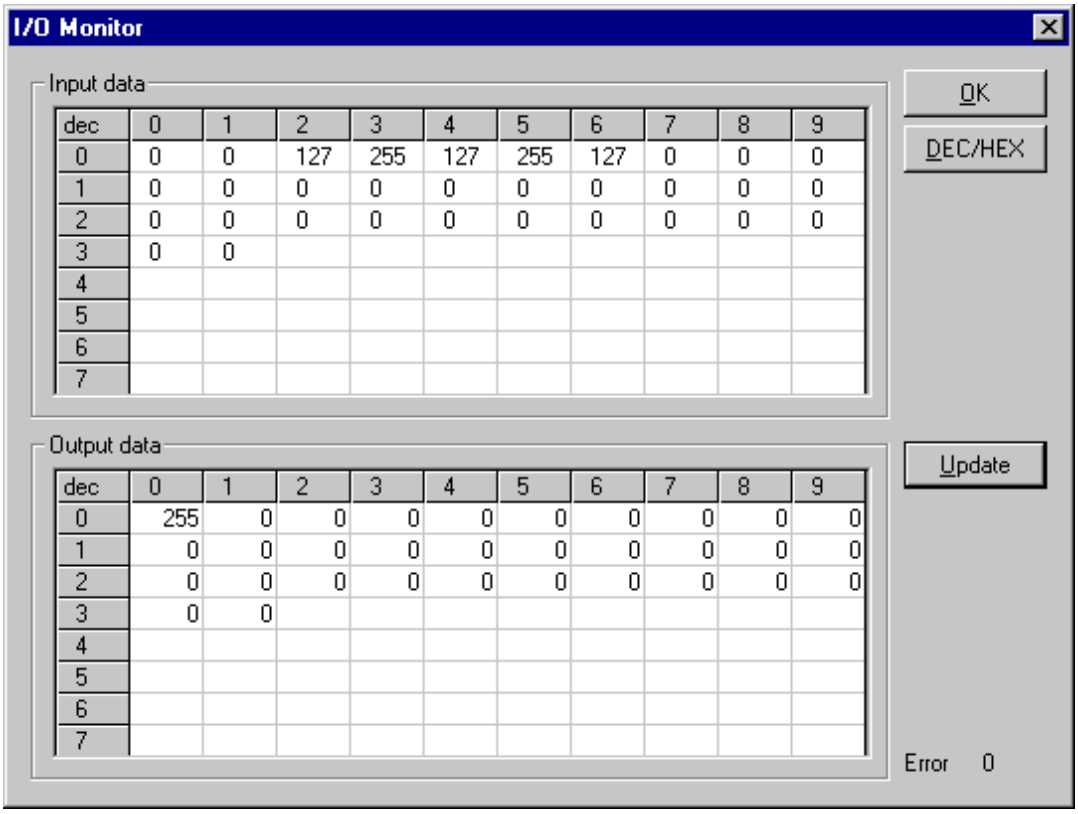


Figure 34: Online > IO Monitor

DEC/HEX converts the display of the input data. The output data are always in the decimal form.
Enter the output value and then press **Update**.
Always the first 32 input and output Bytes of the process description are shown, also when these Bytes have not been occupied by the configuration.
The display is always in a byte manner.
The I/O Watch Monitor described in the next section allows a more comfortable display.

6.22. 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 (Kombimaster) bzw. 1.140 (DP-Master)
InterBus Master	2.040
CANopen Master	1.040
DeviceNet Master	1.058

Table 18: Firmware for I/O Watch functions

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

- The project/configuration already exists, containing a DeviceNet Master and a DeviceNet device (or Slave) as described in CHAPTER 3: Getting Started – CSCAPE Configuration starting on page *15*.
- The Configuration has been downloaded into the DeviceNet Master using **Online > Download**
- Running bus system
  1. Open the existing project using **File > Open**.
  2. Open the Windows dropdown menu and select **Window > Logical Network View** to change the window. A window with three sections opens

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

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

     Project > Master > Device > Connection Type > Date Type

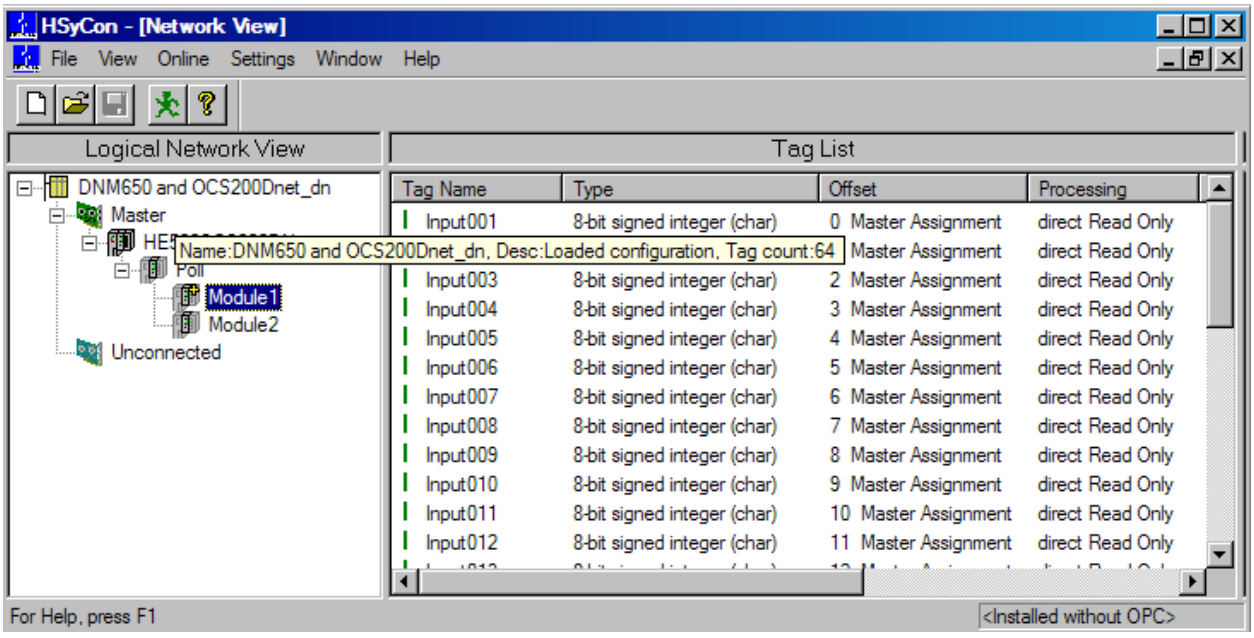
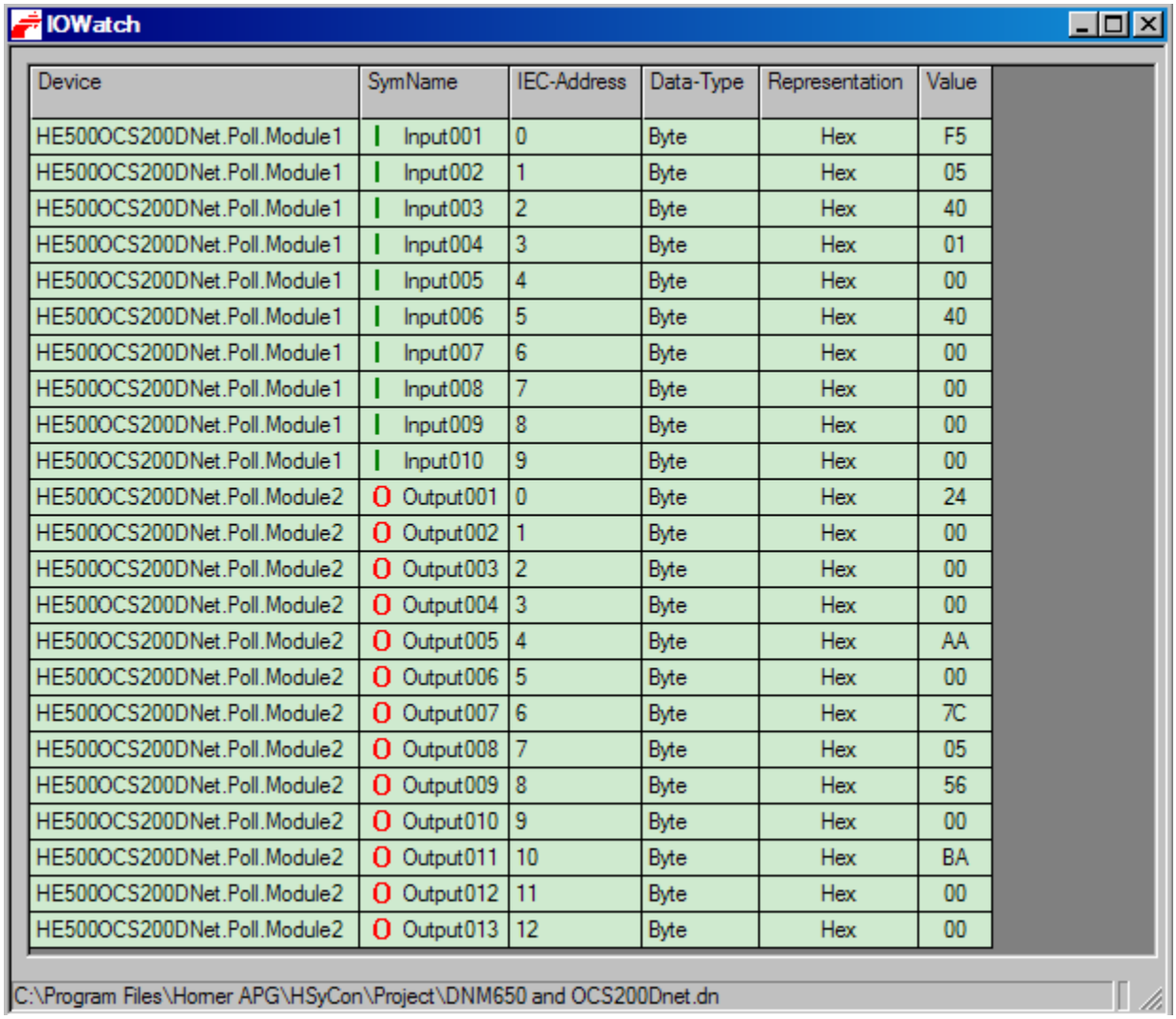


Figure 35: Logical Network View and I/O Watch

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



Device	SymName	IEC-Address	Data-Type	Representation	Value
HE500OCS200DNet.Poll.Module1	Input001	0	Byte	Hex	F5
HE500OCS200DNet.Poll.Module1	Input002	1	Byte	Hex	05
HE500OCS200DNet.Poll.Module1	Input003	2	Byte	Hex	40
HE500OCS200DNet.Poll.Module1	Input004	3	Byte	Hex	01
HE500OCS200DNet.Poll.Module1	Input005	4	Byte	Hex	00
HE500OCS200DNet.Poll.Module1	Input006	5	Byte	Hex	40
HE500OCS200DNet.Poll.Module1	Input007	6	Byte	Hex	00
HE500OCS200DNet.Poll.Module1	Input008	7	Byte	Hex	00
HE500OCS200DNet.Poll.Module1	Input009	8	Byte	Hex	00
HE500OCS200DNet.Poll.Module1	Input010	9	Byte	Hex	00
HE500OCS200DNet.Poll.Module2	Output001	0	Byte	Hex	24
HE500OCS200DNet.Poll.Module2	Output002	1	Byte	Hex	00
HE500OCS200DNet.Poll.Module2	Output003	2	Byte	Hex	00
HE500OCS200DNet.Poll.Module2	Output004	3	Byte	Hex	00
HE500OCS200DNet.Poll.Module2	Output005	4	Byte	Hex	AA
HE500OCS200DNet.Poll.Module2	Output006	5	Byte	Hex	00
HE500OCS200DNet.Poll.Module2	Output007	6	Byte	Hex	7C
HE500OCS200DNet.Poll.Module2	Output008	7	Byte	Hex	05
HE500OCS200DNet.Poll.Module2	Output009	8	Byte	Hex	56
HE500OCS200DNet.Poll.Module2	Output010	9	Byte	Hex	00
HE500OCS200DNet.Poll.Module2	Output011	10	Byte	Hex	BA
HE500OCS200DNet.Poll.Module2	Output012	11	Byte	Hex	00
HE500OCS200DNet.Poll.Module2	Output013	12	Byte	Hex	00

Figure 36: I/O Watch Window

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

6.23. DeviceNet Services

6.24. Get Device Attribute

This menu selection enables the user to get/receive attribute related information from a Slave device. The user should be familiar with the supported Class, Instance, and Attribute entrees for the Slave device. These entries should be available within the suppliers data sheet for the Slave product. The return value will be represented in Hexadecimal. Clicking the ASCII button will change this value to ASCII text. The Hexadecimal code can be resorted by clicking now the Hex button. Clicking the Get button will receive the Value from the device.

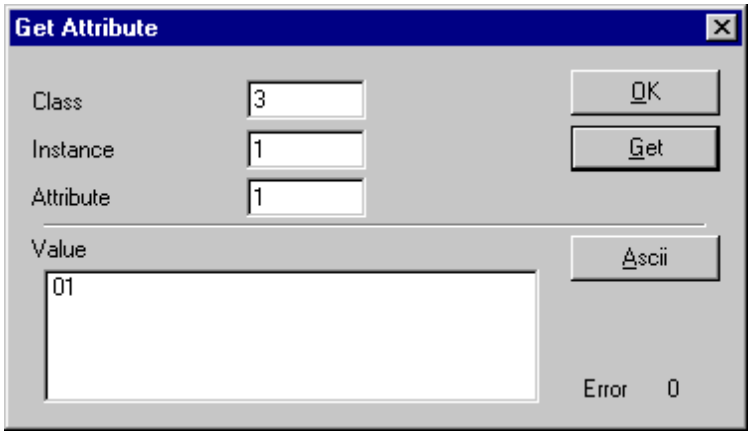


Figure 37: Get Attribute Window

6.25. Set Device Attribute

This menu selection enables the user to set a attribute related to a Slave device. The user should be familiar with the supported Class, Instance, and Attribute entrees for the Slave device. These entrees should be available within the suppliers data sheet for the Slave product. The Value will be represented in Hexadecimal. Clicking the Set button will send the information to the Slave device.

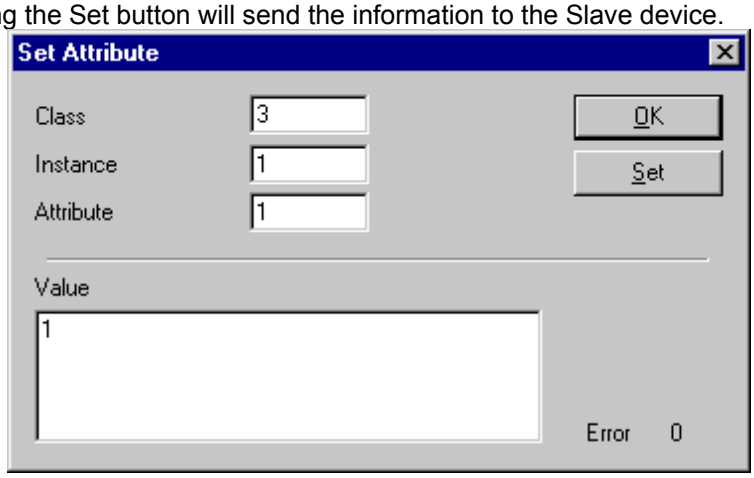


Figure 38: Set Attribute Window

6.26. Change MAC-ID

To change a MAC-ID of a DeviceNet device (slave), open the Live List. This is described in section Change MAC-ID on page 49.

6.27. Message Monitor

The Message Monitor permits access to the Mailbox of the CIF. The usage of the Message Monitor assumes advanced knowledge from the user.
First the Horner device must be chosen with a left mouse click on the symbol of the Horner device. Then call up the **Online > Message Monitor** menu.

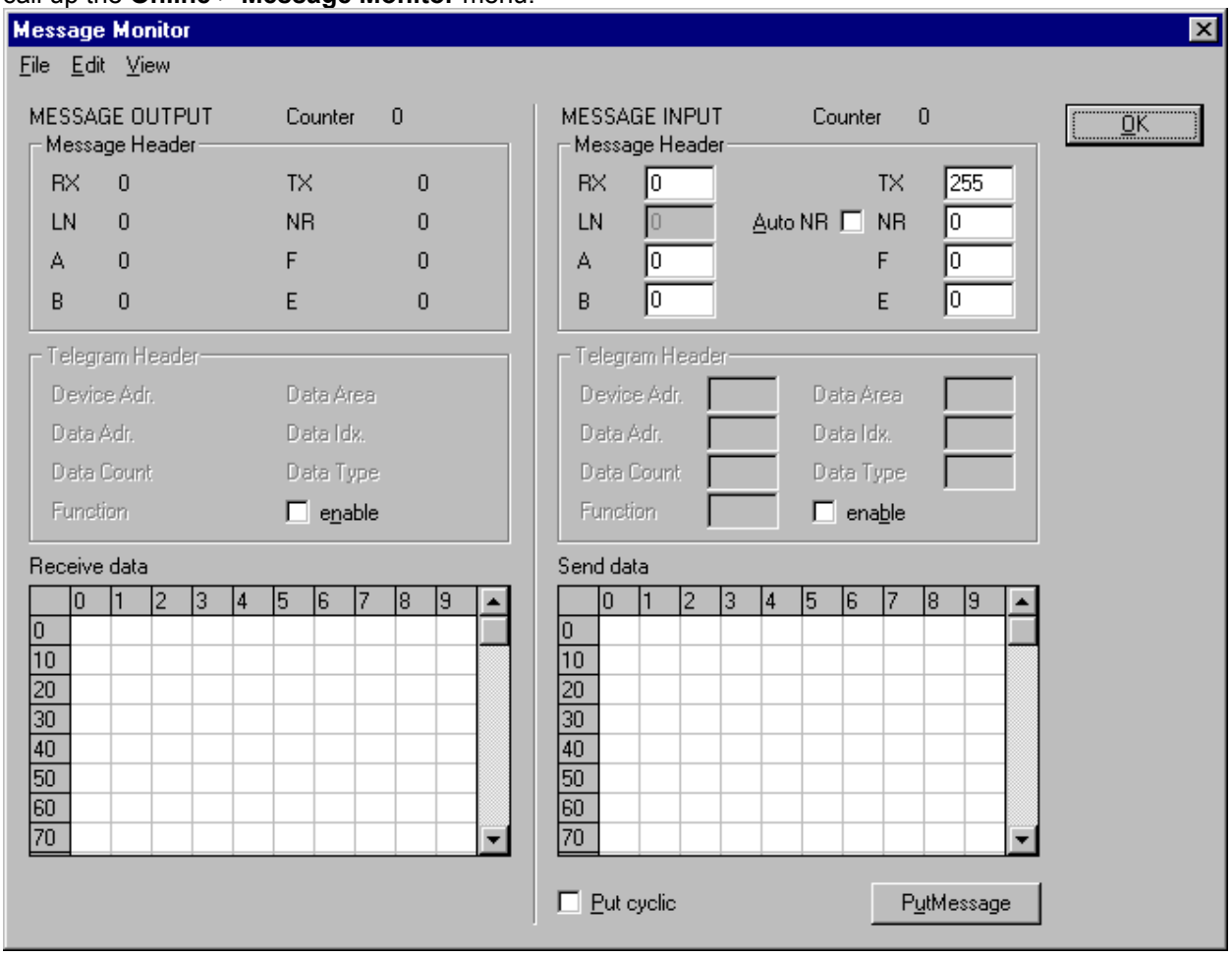


Figure 39: Online > Message Monitor

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 HSyCon.
Edit > Create answer: creates an answer Message
Edit > Reset counter: resets the Message counter
View > Review the received data: all received data is shown
View > Review the send data: all the send data is shown
View > Number of receipt errors: the number of the receipt errors are shown
View > Decimal/Hexadecimal: Switch the display format

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

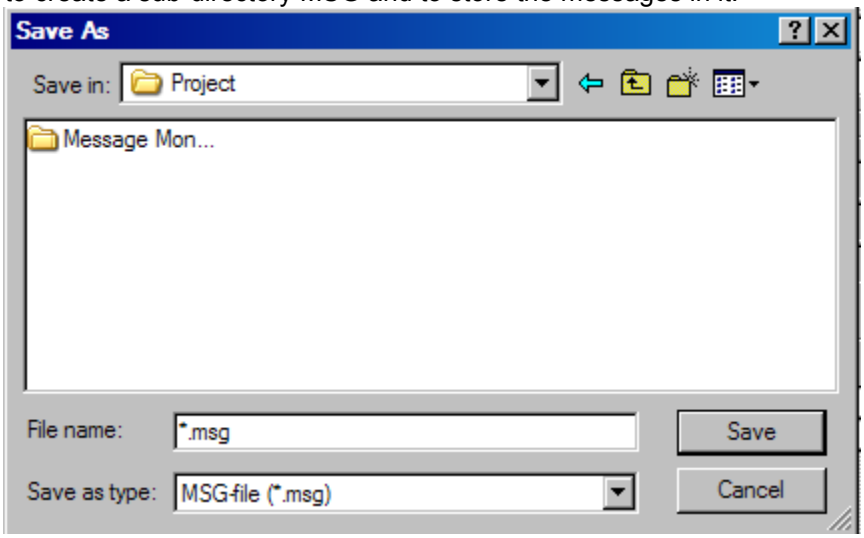


Figure 40: Save a Message

6.28. Message Monitor for testing explicit messaging of DeviceNet

In the following, the Message Monitor for reading and writing data via DeviceNet at the Master is described.
The following must be entered in the Message Monitor in order to read or write data via DeviceNet from a Device:

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

Table 19: Message Monitor – Example DeviceNet

CHAPTER 7: FILE, PRINT, EXPORT, EDIT AND VIEW

7.1. File

7.1.1. Open

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

7.1.2. Save and Save As

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

7.1.3. Close

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

7.2. Print

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

Figure 41: File > Print

The base setting prints information on one sheet only for one device.
Topology the topology of the Bus system.
Bus parameters prints the Bus parameters of the Bus system.
Address table prints the address table of the Master.
Device table prints the device table.

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

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

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

7.3. Export Functions

7.3.1. DBM Export

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

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 placed in the same directory as the configuration, but with the ending *.csv.
The CSV file can be read with a table program like for example Excel.
The CSV Export saves only the text and the values of the configured Slaves. The meaning of the individual values can be shown in the table.
Here is the description of the text and values:

Value	Meaning
Stationaddress	The Stationaddress is the clear device address of the Slave on the bus.
RecordType	The RecordType defines the arrangement of the structure and is always defaultly 2.
IdentNumber	This number is the clear device number of the Slave (if available).
VendorNumber	The VendorNumber is the clear number of the vendor (if available).
VendorName	Here the name of the vendor is shown.
Device	Name of the device.
Description	This is the description of the device, which is set by the user.
MasterAddress	This is the number of the Master Address, where the devices are connected to (if available).
Settings	The Settings of the I/O Data is shown here.
Reserved	reserved
ModulCount	The ModulCount gives the number of the actual modules. Depending on this there are the DataType, DataSize, DataPosition and the Address from 0 to 59.
DataType_0	The DataType, which is used in the configuration. The code for this can be found in section 7.3.2.1. DataType Code.
DataSize_0	Number of bytes, which were used by the module.
DataPosition_0	The byte DataPosition, which is used in the configuration. The code for this can be found in section 7.3.2.2. DataPosition Code.
Address_0	Offset Address in the Dual-port memory

Table 20: CSV Export - Meaning of the values

7.3.2.1. DataType Code

D7	D6	D5	D4	D3	D2	D1	D0
SubFlag	Data Direction			Data Format			
				according EN standard 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 emty space 1 input 2 output						
0 start of a module 1 submodule							

Table 21: CSV Export > DataType Code

7.3.2.2. DataPosition Code

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

Table 22: CSV Export - DataPosition Code

Example of a CSV file which was exported in Excel:

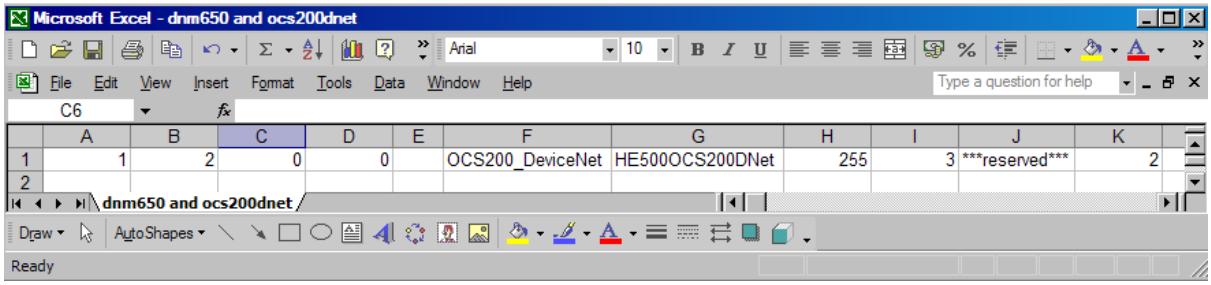


Table 23: Example of a CSV file in Excel format

The DeviceNet Salve device has the **Station address** 1 (A1).
The **RecordType** is default 2 (B1).
The **IdentNumber** of the Slave is 0 (C1).
No **VendorNumber** is available, so this is 0 (D1).
The **VendorName** is empty (Horner) (E1).
The **Device** has the description OCS200_DeviceNet (F1).
The **Description** of the Device is HE500OCS200DNet (G1).
The **Masteraddress** is 255 (H1).
There are **Settings** of the I/O data, so this is 3, (I1).
This field is **Reserved**, because of "reserved" is displayed (J1).
The number of the actual modules is 2, so in the field **ModulCount** a 2 is shown (K1).
The **DataType** of module 1 is 2 (K1).
The **DataSize** of the first module is 32 (L1).

The **DataPosition** of the first module is 26 (M1).
In the following field the DataType, DataSize and DataPosition of the following modules are shown.

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

7.4. Edit

With **Edit > Cut** and **Edit > Copy** the Slave device with its settings and configuration (not the description of the device) is placed on the Clipboard and with **Edit > Paste** it can be inserted.
The difference between **Cut** and **Copy** is:
With the menu option **Edit > Cut** the Slave device can be moved from one point in the configuration to another. With the menu option **Edit > Copy** an existing Slave device is duplicated.
Selecting **Edit > Cut** gives the following security question:

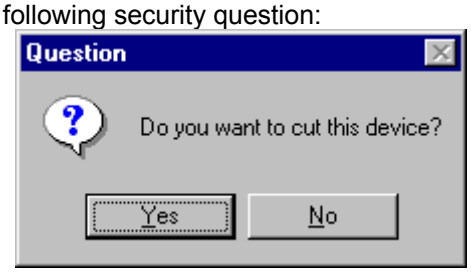


Figure 42: Security question cut device

If **Yes** is selected, the Slave device is cut and stays in the clipboard.
When **Edit > Paste** is selected, the device can be inserted again at the position required.
The mouse pointer changes into the "insert device mouse pointer". Click the position where the device should be inserted. A window opens, where the cut/copied device can be selected.

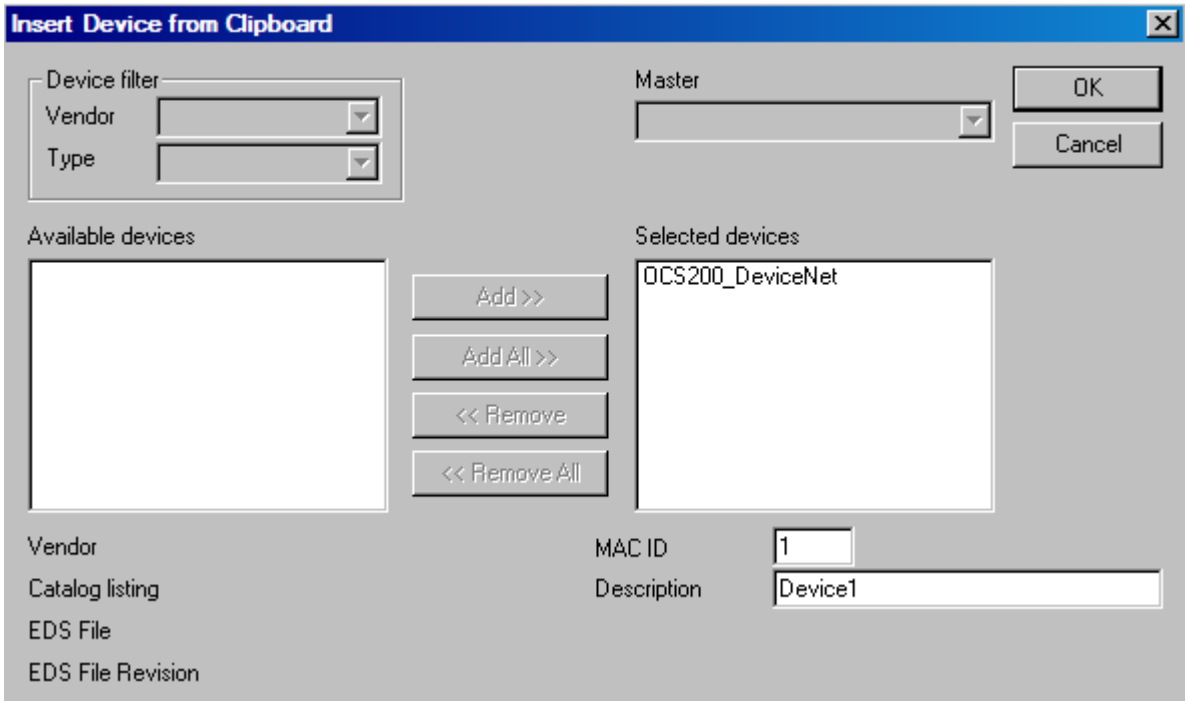


Figure 43: Edit > Paste cut/copied device

Click **OK** button to insert the device.

7.4.1. Delete

To delete a Slave device, select the device by clicking on it. Then select the menu **Edit > Delete**. Before HSyCon deletes the Slave a security question appears.

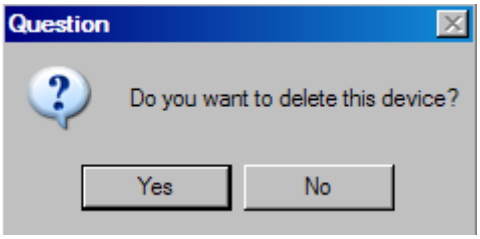


Figure 44: Security question delete device

Note: When a device is deleted, the settings and the configuration of this device are deleted.

7.4.1. Replace

With the menu **Edit > Replace** the Slave device can be replaced. Look in section Replace Slave at page 25.

7.5. View of the Configuration

7.5.1. Device Table

The **View > Device Table** menu shows the list of all devices that have been inserted.

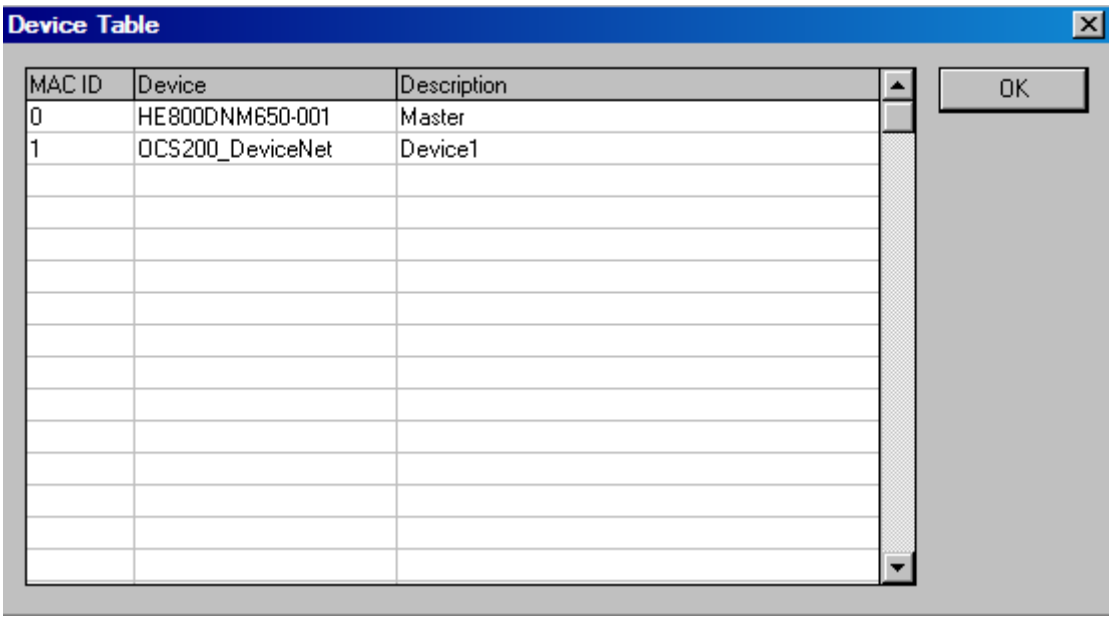


Figure 45: View > Device Table

7.5.2. Address Table

A list of all addresses used in the process depiction is displayed in the **View > Address Table** menu. For this purpose the current Master for which the table is to be displayed must be chosen.
Note: Addresses refer to the Master.

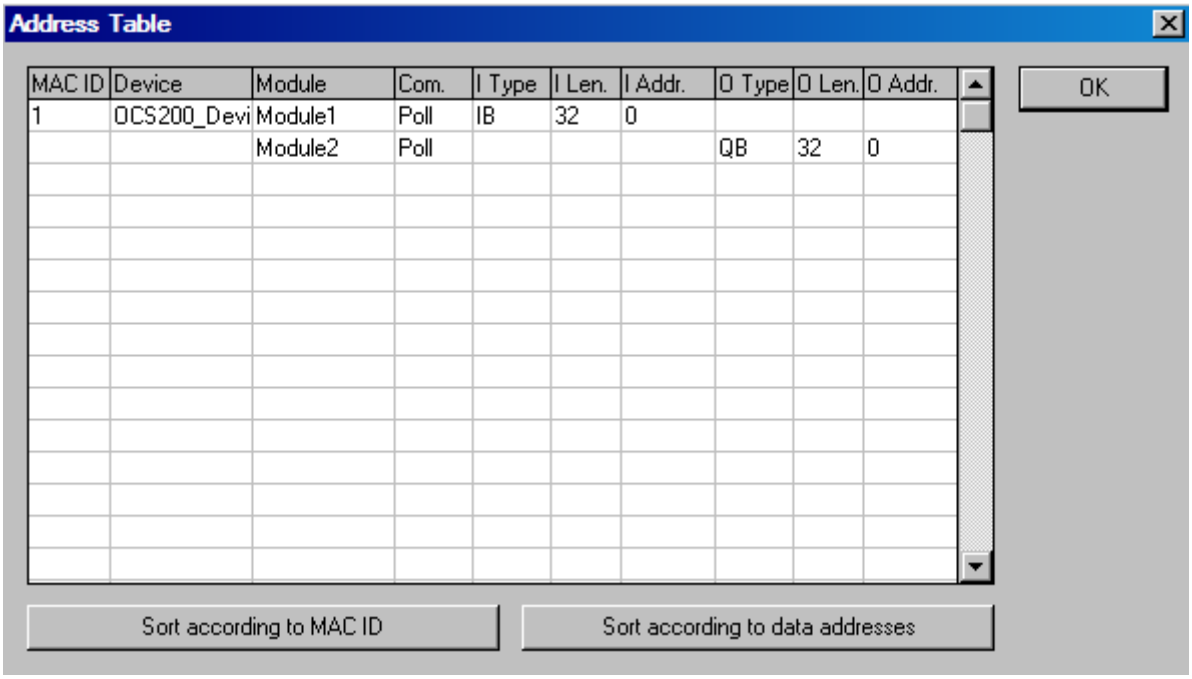


Figure 46: View > Address Table

7.6. View Menu HSyCon

7.6.1. Logical Network View

In the menu **View > Logical Network View** the user can activate or deactivate the network view by selecting its (with hook) or by not selecting it (without hook).
The network view is used for example for the Start Options.

7.6.2. 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).

CHAPTER 8: ERROR NUMBERS

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

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

Error Number	Description
-1	Driver: Board not initialized The communication board is not initialized by the driver. No or wrong configuration found for the given board, check the driver configuration. Driver function used without calling DevOpenDriver() first.
-2	Driver: Error in internal ‘Init state‘
-3	Driver: Error in internal ‘Read state‘
-4	Driver: Command on this channel is active
-5	Driver: Unknown parameter in function occurred
-6	Driver: Version is incompatible The device driver version does not correspond to the driver DLL version. From version V1.200 the internal command structure between DLL and driver has changed. Make sure to use the same version of the device driver and the driver DLL.
-10	Device: Dual port memory RAM not accessible (board not found) Dual-ported RAM (DPM) not accessible / no hardware found. This error occurs, when the driver is not able to read or write to the Dual-port memory. Check the BIOS setting of the PC Memory address conflict with other PC components. Try another memory address, check the driver configuration for this board, 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 initialisation error. No data base is loaded into the device or a wrong parameter can causes that a task can't initialise.
-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 24: 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 an interrupt is used, 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 polling mode is used, 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 an interrupt is used, 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 polling mode is used, 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 25: 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 initialisation error. No data base is loaded into the device or a wrong parameter can causes that a task can't initialise. 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 an interrupt is used, 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 polling mode is used, 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 an interrupt is used, 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 polling mode is used, 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 26: 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 initialised (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 27:CIF Device Driver Error Numbers (-30..-49)

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

Table 28: CIF Device Driver Error Numbers (1000)

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

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

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

Table 29: 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 30: CIF Serial Driver Error Numbers (-20..-47)

8.3. 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 Horner 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 Horner device. An example of this communication is the download of a configuration.

Error Number	Description
4	Task does not exist
5	Task is not initialised
6	The MCL is locked
7	The MCL rejects a send command because of an error
20	The user will download a database into the device that is not valid for this device type.
21	Data base segment not configured or not existed
22	Number for message wrong during download
23	Received number of data during download does not match to that in the command message
24	Sequence identifier wrong during download
25	Checksum after download and checksum in command message do not match
26	Write/Read access of data base segment
27	Download/Upload or erase of configured data base type is not allowed
28	The state of the data base segment indicated an error. Upload not possible
29	The access to the data base segment needs the bootstraploader. The bootstraploader is not present
30	Trace buffer overflow
31	Entry into trace buffer too long
37	No or wrong licence. The OEM licence of the System Configurator allows only communication to devices that have the same licence 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 31: 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	Licence code invalid
92	Licence code does already exist
93	All memory locations for licence codes already in use

Table 32: RCS error numbers (answer message) (40..93)

8.4. 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	Nullpointer 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 33: Database Access Error Numbers (100..130)

8.5. 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 DeviveObject 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 34: Online Data Manager Error numbers (1000..1018)

8.6. 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 35: Error Numbers of the Message Handler of the Online Data Manager (2010..2027)

8.7. 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 36: Error Numbers of the Driver Functions of the Online Data Manager (2501..2512)

8.8. 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 37: Subfunction Error Numbers of the Driver Functions of the Online Data Manager (8001..8035)

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

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

Error Number	Description
4000	File does not exist
4001	Success in 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 38: 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 initialisation
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 39: 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 40: 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 initialisation 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 41: Error numbers of converting functions (4082..4199)

8.10. 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 greather 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 42: Error Numbers of data base functions (5000 .. 5012)

CHAPTER 9: EXPLICIT MESSAGING USING DNM650

9.1 General

The DNM650 supports both POLLED and EXPLICIT connections. Explicit Messaging requires a great deal of overhead in both the DNM650 and the OCS. Multiple PLC scans may be required to access the required data between the OCS and the DNM650. As a result, Explicit Messages should be reserved for access to infrequently needed data, such as configuration or tuning parameters only.

The sequence below presents a general description of the process that must be executed to service an Explicit request.

1. The ladder code builds an explicit request message in a group of %R registers.
2. The ladder code plugs the start of and length of the transmit and receive buffers into four %AQ registers. The transmit and receive buffers must both be located in the %R register space of the OCS.
3. The ladder code then sets the “Send Explicit Message” command bit.
4. The DNM650 periodically checks the “Send Explicit Message” command bit.
5. If the “Send Explicit Message” command bit is set, processing of the explicit request begins. This command bit must remain on until success or error status is returned.
6. The four %AQ registers are examined and checked for validity.
7. The DNM650 requests the PLC to send the transmit buffer.
8. The DNM650 then checks the MACID contained in the transmit buffer. Several checks take place, MACID out-of-range, referenced node not configured and referenced node not On-line.
9. Then the message is formatted and sent to the referenced node.
10. When the response message is received, a check is made to see if the allocated receive buffer is large enough to accept the response message.
11. The message is then sent to the block of %R registers designated as the receive buffer.
12. The Explicit Transaction Complete bit is then set.
13. If there were any errors detected in any of the previous steps, the process is aborted and the appropriate error status bit along with the Explicit Transaction Complete bit are set.

9.2 Building Explicit Messages

In the following example, let’s assume that we want to read the polled consumption size from the node at MACID 3, we also want to locate the transmit buffer at %R101, the receive buffer at %R51 and we will also allocate 20 bytes to the receive buffer.

BYTE NUMBER	REGISTER NUMBER	DESCRIPTION	TRANSMIT BUFFER DATA
0	R101 LSB	MACID (Node Address)	03
1	R101 MSB	Service Code (Get Attribute Single)	14
2	R102 LSB	Class ID (Connection Class)	05
3	R102 MSB		00
4	R103 LSB	Instance ID (Polled Connection)	02
5	R103 MSB		00
6	R103 LSB	Attribute # (Consumption Size)	07

REGISTER	DESCRIPTION	VALUE
%AQ1	Start of Transmit Buffer	101
%AQ2	Number of Bytes to Transmit	7
%AQ3	Start of Receive Buffer	51
%AQ4	Receive Buffer Allocation Size	20

In the following example we want to write the polled expected packet rate to the node at MACID 3, we also want to locate the transmit buffer at %R101, the receive buffer at %R51, we will also allocate 20 bytes to the receive buffer.

BYTE NUMBER	REGISTER NUMBER	DESCRIPTION	TRANSMIT BUFFER DATA
0	R101 LSB	MACID (Node Address)	03
1	R101 MSB	Service Code (Set Attribute Single)	16
2	R102 LSB	Class ID (Connection Class)	05
3	R102 MSB		00
4	R103 LSB	Instance ID (Polled Connection)	02
5	R103 MSB		00
6	R104 LSB	Attribute (Expected Packet Rate)	07
7	R104 MSB	Data Value Low Byte	E8(Hex)
8	R105 LSB	Data Value High Byte	03

REGISTER	DESCRIPTION	VALUE
%AQ1	Start of Transmit Buffer	101
%AQ2	Number of Bytes to Transmit	9
%AQ3	Start of Receive Buffer	51
%AQ4	Receive Buffer Allocation Size	20

9.3 How to Interpret Explicit Response Messages

The normal, expected response from an Explicit Message is the Acknowledge Message:

BYTE NUMBER	DESCRIPTION	EXAMPLE
0	Number of bytes received	Xx
1		Xx
2	MACID	01
3	*Service Code	0x90(Hex)
4	Optional Data	Xx
-	Optional Data	Xx
-	Optional Data	Xx
n	Optional Data	Xx
* Previously sent Service Code (0x10) + 0x80		

Note: The Most Significant Bit in the Service Code byte is used as a Response Bit indicating that the command was properly received. [Service Code 0x10 + Response Bit 0x80 = 0x90]. If any extra data needs to be returned, that data will be placed into subsequent bytes.

Note: The number of bytes received value is the number of bytes of the message placed in the receive buffer including the two bytes devoted to the number of bytes received.

9.4 Explicit Message Errors

In the case that an Explicit Message requests a function that cannot be performed by the referenced node, an Explicit Error Message will be returned.

An Explicit Error Message takes the following form:

Byte Number	Description	Example (values in HEX)
0	Number of bytes received	06
1		00
2	MACID	01
3	Service Code	94
4	General Error Code	xx
5	Additional Error Code	xx

Word Offset	Value (from example)
%R1	0006
%R2	0x9401
%R3	xxxx

An Error is indicated by a 0x94 in the service code byte. This indicates that the referenced node has detected an error and the two bytes following indicate the specifics to that error. The next byte indicates the General Error Code. The last byte contains an Additional Error Code to indicate additional information. See appendix C for a list of and definitions of General and Additional Error Codes.

Many DeviceNet manufactures have defined vendor specific error codes. In this case the General error code will be in the range of 0xD0 to 0xFF (Vendor-specific Object and Class errors). In these cases the Additional error code can be anything. Consult the documentation from the node manufacture for more details.

APPENDIX:

A. EXTENDED DEVICE DIAGNOSTIC MASTER

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

A.1. PLC_TASK Common Variables

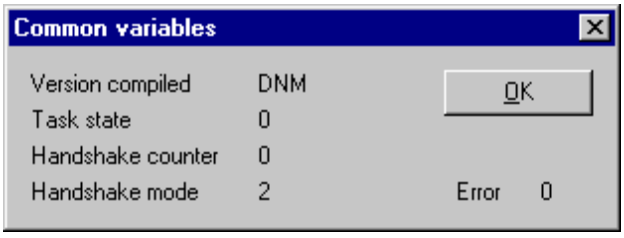


Figure 47: PLC_TASK Common Variables

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

Table 43: PLC_TASK Common Variables

A.2. DNM_TASK Common Variables

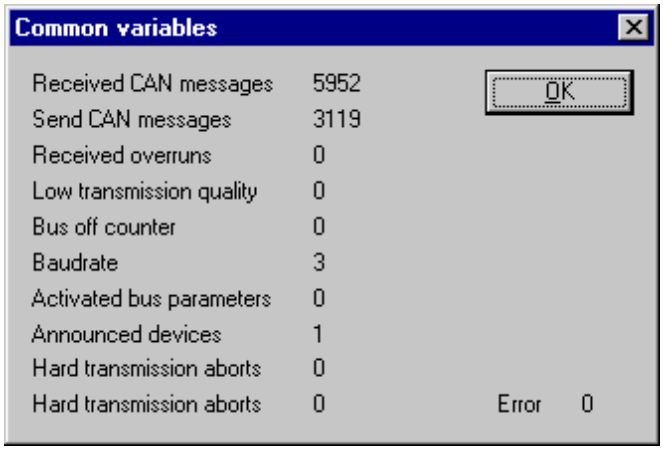


Figure 48: DNM_TASK Common Variables

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

Table 44: DNM_TASK Common Variables

A.3. DNM_TASK Device Running States

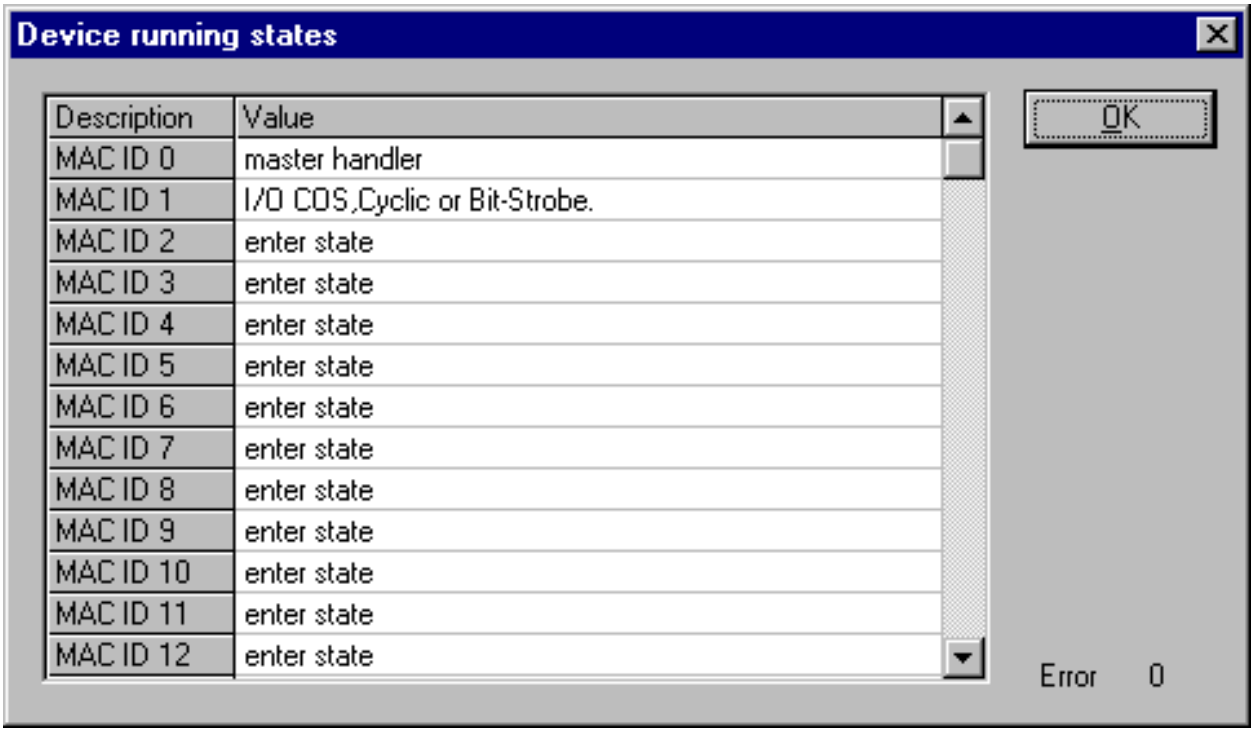


Figure 49: DNM_Task Device Running States

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

A.4. DNM_Task Global State Field

See section Global State Field at page 52.

A.5. DNM_Task Communication Error

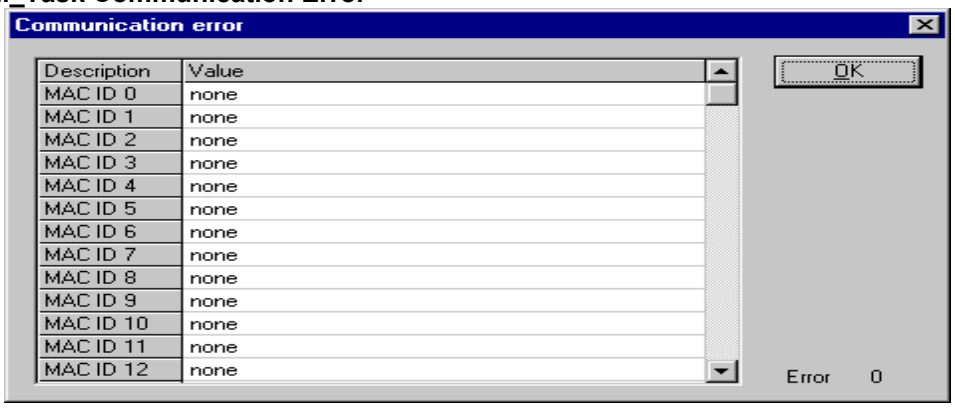


Figure 50: DNM_Task Communication Error

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

A.6. DNM_Task Receive Queue

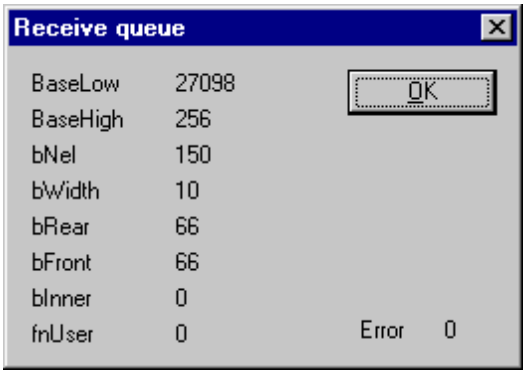


Figure 51: DNM_Task Receive Queue

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

Variable	Meaning
bRear	A pointer to where the next message will be dequeue from the queue body
bFront	A pointer to where the next message will be stored
bInner	The actual number of stored messages

Table 45: Receive Queue

A.7. DNM_Task Transmit Queue

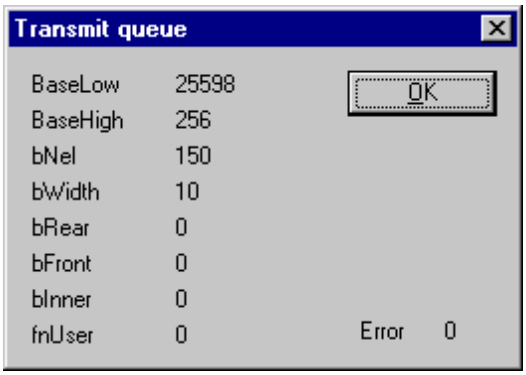


Figure 52: DNM_Task Transmit Queue

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

Variable	Meaning
BaseLow	not documentated
BaseHigh	not documentated
bNel	not documented
bWidth	not documented
bRear	A pointer to where the next message will be deckle from the queue body
bFront	A pointer to where the next message will be stored
bInner	The actual number of stored messages
fnUser	not documented

Table 46: DNM_Task Transmit Queue

A.8. DNM_Task DeviceNet Command Counters

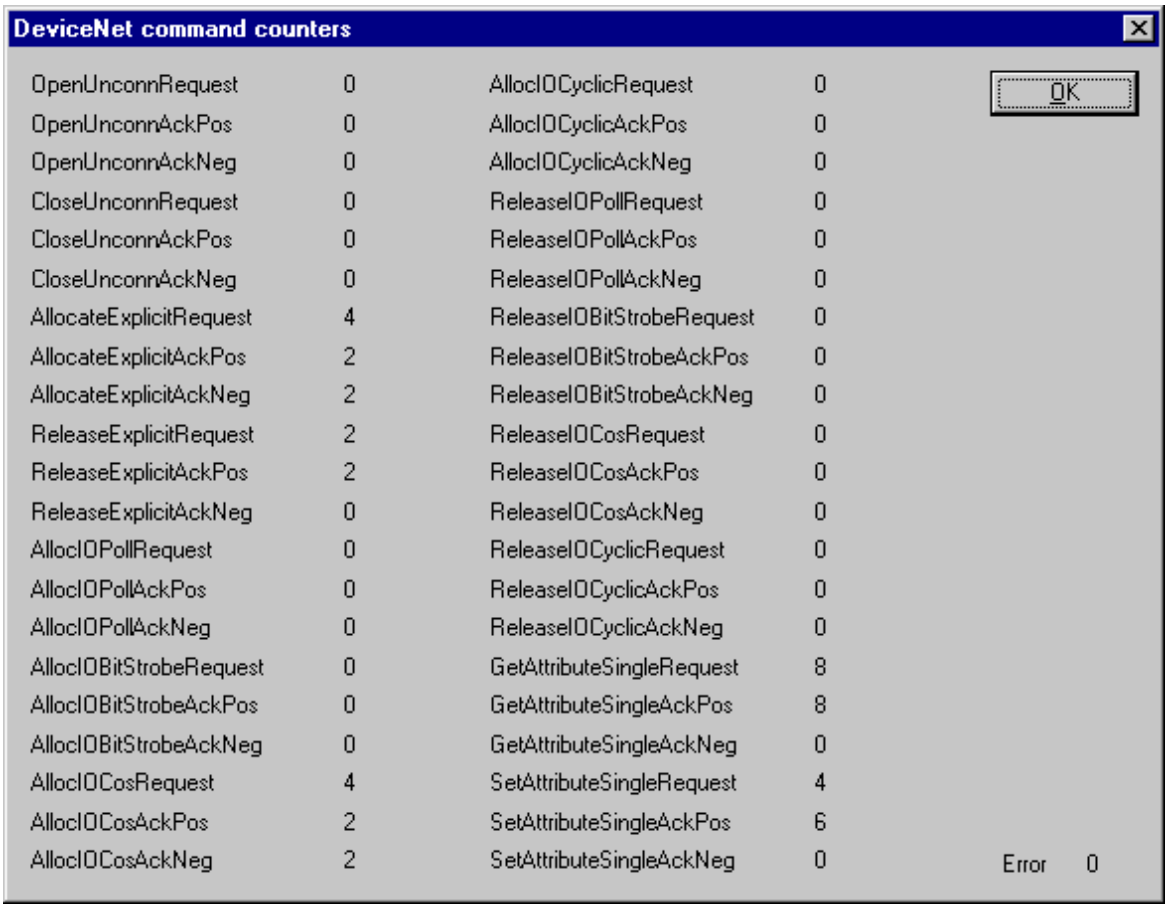


Figure 53: DNM_Task DeviceNet Command Counters

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

A.9. DNM_Task Timeout Counter

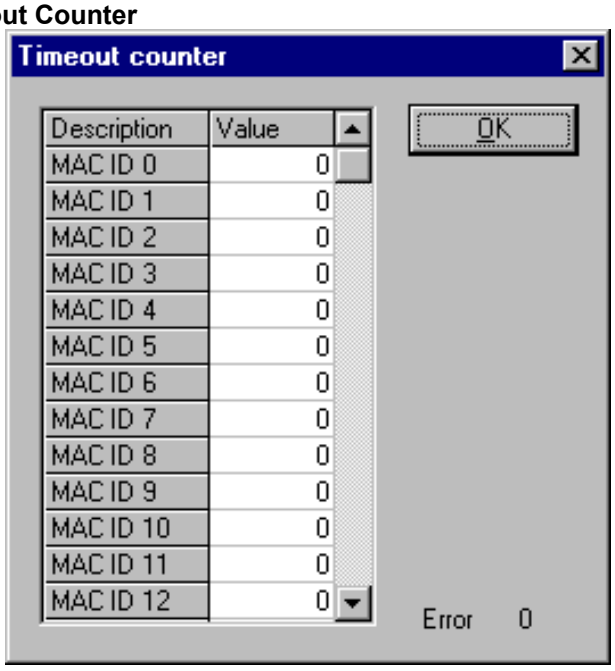


Figure 54: DNM_Task Timeout Counter

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

A.10. DNM_Task Init Counter

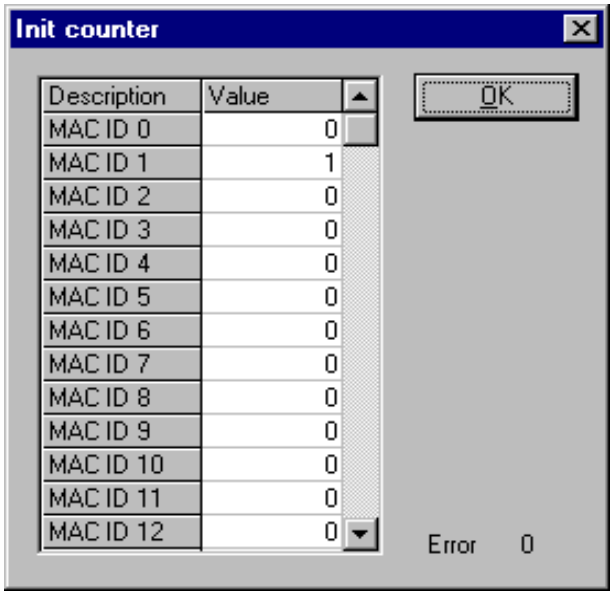


Figure 55: DNM_Task Init Counter

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

B. EXTENDED DEVICE DIAGNOSTIC DEVICE (SLAVE)

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

B.1. PLC_Task Common Variables (Device)

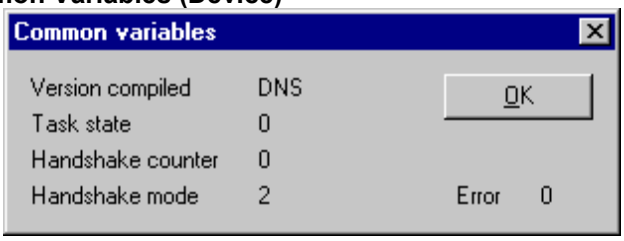


Figure 56: PLC_Task Common Variables (Slave)

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

Table 47: PLC_Task Common Variables (Slave)

B.2. DNS_Task Common Variables

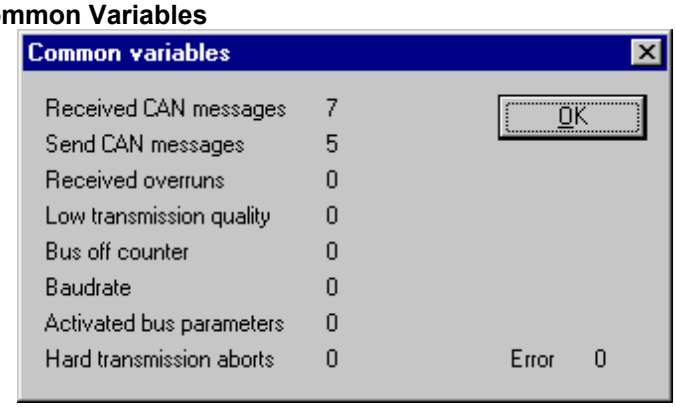


Figure 57: DNS_Task Common Variables

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

Table 48: DNS_Task Common Variables

B.3. DNS_TASK Receive Queue (Device)

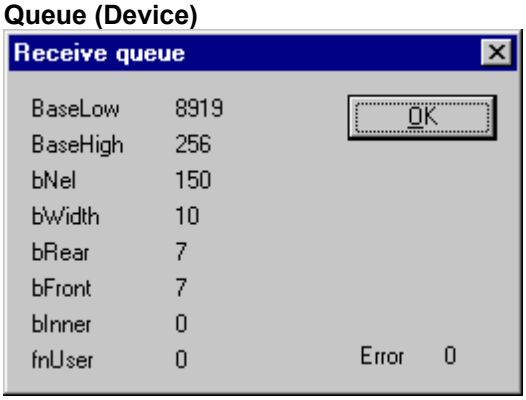


Figure 58: DNS_TASK Receive Queue (Device)

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

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

Table 49: DNS_TASK Receive Queue (Device)

B.4. DNS_TASK Transmit Queue (Device)

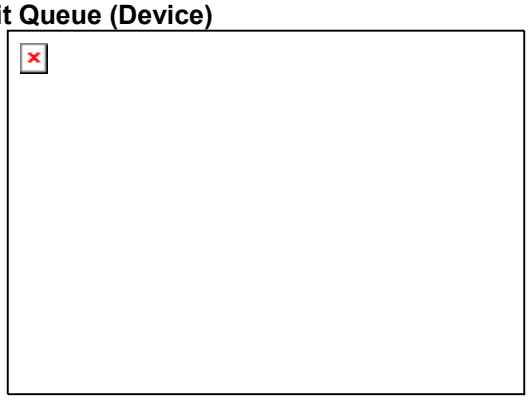


Figure 59: DNS_TASK Transmit Queue

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

Table 50: Transmit Queue (Device)

NOTES