Open Architecture Control Software

SoftPLC Driver
Using the Hilscher CIF30 or CIF50 Cards
(eg: Profibus, DeviceNet, Interbus, etc.)

Because of the variety of uses of the information described in this manual, the users of, and those responsible for applying this information must satisfy themselves as to the acceptability of each application and use of the information. In no event will SoftPLC Corporation be responsible or liable for its use, nor for any infringements of patents or other rights of third parties which may result from its use.

SoftPLC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE.

SoftPLC Corporation reserves the right to change product specifications at any time without notice.

First Printing: September, 1996
Latest Printing:August, 2009

SoftPLC Corporation
25603 Red Brangus Dr.
Spicewood, Texas 78669 USA
Telephone: 512/264-8394 or 1-800-SoftPLC
Fax: 512/264-8399
WWW: http://www.softplc.com
Email: info@softplc.com

TABLE OF CONTENTS

1. INTRODUCTION

SoftPLC Corporation's SoftPLC product operates as a programmable controller and supports loadable drivers called TLM's. TLM's are used to add I/O device support and to extend the instruction set of SoftPLC.

Hilscher GmbH is a German company that makes a series of intelligent I/O scanner cards, which are sold through distributors. SoftPLC Corporation is a distributor of some of these cards. The SoftPLC driver for the Hilscher model CIF30 products, called HILSCHER.TLM, currently supports these ISA bus and PC/104 bus, Profibus and Interbus cards from Hilscher:

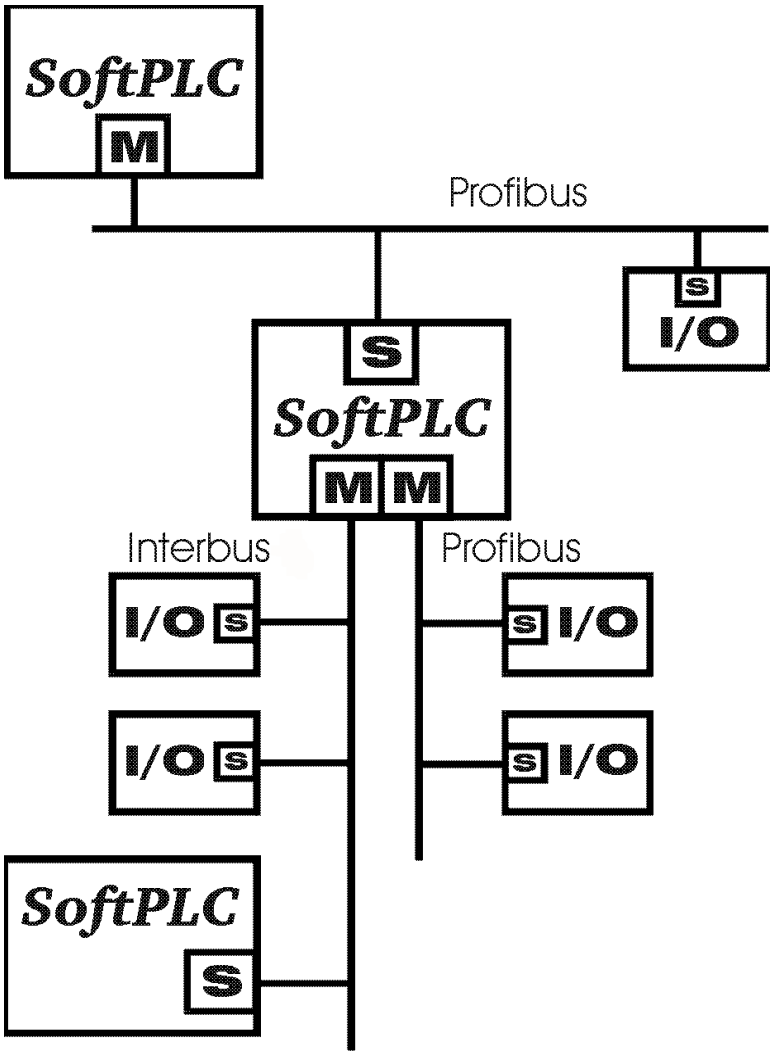
CIF30-DPM CIF104-DPM Profibus Master
CIF30-IBM CIF104-IBM Interbus Master
CIF30-DPS CIF104-DPS Profibus Slave
CIF50-PB (SoftPLC version 4.x only)

HILSCHER.TLM supports up to 8 cards simultaneously. The HILSCHER.TLM does not use hardware interrupts and uses only a 2K memory footprint for the CIF30 and CIF104 cards. The CIF50 card uses an 8K memory footprint. With the small memory footprint and no IRQ usage, there will rarely be resource availability problems which would preclude actually using 8 cards simultaneously. The most likely resource limit would be a lack of sufficient ISA or PCI bus slots.

Cards may be configured either as masters or slaves. As a master, a card is responsible for sending outputs and for receiving inputs. As a slave, a card is responsible for receiving outputs from a master, and responding with inputs to the master. Any particular host running SoftPLC may be configured to be multiple masters and/or multiple slaves on multiple Profibus or Interbus networks. There is tremendous flexibility with just this one HILSCHER.TLM and the proper types of Hilscher cards.

Using a master/slave scheme between SoftPLC nodes is a very effective way to transfer a few hundred words between various SoftPLC nodes in an extremely fast fashion, normally 2-20 milliseconds per transfer depending on distances, baudrates, scan times, and number of slaves per bus segment. Figure 1 shows examples of how this technology may be used.

Figure 1 - Example Configurations



M = Master
S = Slave

2. OVERVIEW

To set up the driver for use with SoftPLC, you edit a configuration text file through the TOPDOC NexGen PLC Configuration Menu. The configuration file is named HILSCHER.LST. The memory address of each Hilscher card that you intend to support is listed in this file, along with definitions of blocks of data to transfer from each card (inputs), as well as to each card (outputs). Each card may support a number of data block definitions, either as inputs or outputs.

All Hilscher input data for any particular card is packed into a block of memory on that card, called the **input area**. All Hilscher output data is packed into a 2nd block of memory on that card called the **output area**. There are only two blocks like this, one for inputs and one for outputs. You determine the "packing" or "mapping" when you run the Hilscher configuration program, SYCON.EXE for master cards.

Each type of I/O module, also called a **slave**, will get assigned one or more bytes of memory space in either the input area or output area (or both) of the card to which it will be attached by cable. Sometimes, input modules will use memory only in the input area, and output modules will use memory in the output area. However, it is possible for a slave to get mapped into both the input area and output area if it is intelligent or more feature rich. A slave may also be another Hilscher card resident in a SoftPLC node.

The configuration text file HILSCHER.LST informs HILSCHER.TLM how to copy words of data from the input area into SoftPLC's datatable, and how to copy words of data from SoftPLC's datatable to the output area on each card. This mapping mechanism helps shield the ladder logic programmer from changes in field I/O modules, so that as modules are added or removed, minimal or no changes in ladder logic datatable addresses need be made. This is possible because the transfer of each input area or output area is accomplished **with one or more** block definitions. The fact that you may use several block definitions means you can allow for omissions, additions, or deletions in the layout of either the input area or output area, again with the goal of reducing the need to change ladder logic datatable addresses as changes are made to the field I/O module sets.

One of the nice features about the Hilscher cards is they have a diagnostics serial RS-232 port in addition to the I/O bus port. There is a diagnostics program which can be run on a laptop or separate computer to monitor I/O communications while SoftPLC is controlling the dual port memory of each Hilscher card. The diagnostics program talks to the Hilscher card via the diagnostics serial port at a fixed baudrate. This is a powerful feature, making it possible to troubleshoot intermittent communications problems without having to shut down SoftPLC.

3. HARDWARE INSTALLATION

In the case of Profibus ISA or PC104, the master card is physically different than the slave card, so you must buy the particular type of card you intend to use in advance. They are not field convertible. When used with HILSCHER.TLM, each ISA or PC104 Hilscher card utilizes 2K of memory and no hardware interrupts. You set the jumpers on each card according to the physical memory address that you use, and remove all IRQ jumpers. (The PCI card sets its own memory address and does not use jumpers.) Here is a partial address table for the ISA/PC104 Profibus master card. There are many more possibilities not shown in the range D0000 to DF800. These jumpers are found on pad J4.

Table 1 - Popular Card Jumper Settings

C = closed = installed = 0 (zero)O = open = not installed = 1 (one)

ADDRESS	A19	A18	A17	A16	A15	A14	A13	A12	A11
CA000	O	O	C	C	O	C	O	C	C
CA800	O	O	C	C	O	C	O	C	O
CB000	O	O	C	C	O	C	O	O	C
CB800	O	O	C	C	O	C	O	O	O
CC000	O	O	C	C	O	O	C	C	C
CC800	O	O	C	C	O	O	C	C	O
CD000	O	O	C	C	O	O	C	O	C
CD800	O	O	C	C	O	O	C	O	O
CE000	O	O	C	C	O	O	O	C	C
CE800	O	O	C	C	O	O	O	C	O
CF000	O	O	C	C	O	O	O	O	C
CF800	O	O	C	C	O	O	O	O	O
D0000	O	O	C	O	C	C	C	C	C
D8000	O	O	C	O	O	C	C	C	C

Remove the IRQ jumpers so each Hilscher card does not use interrupts. Turn off power before installing the cards in your computer.

4. SOFTWARE INSTALLATION

When working with the Hilscher cards you must use both a SoftPLC computer as well as a separate diagnostics computer running Windows 9x or NT/2000. This latter computer is not needed full time, and will be referred to here as “your laptop”. You will also need a null modem cable which will run from the laptop to the Hilscher diagnostics port, but only when doing diagnostics and configuration. SYCON.EXE is a Win32 program, and you won’t have Windows installed on your SoftPLC machine.

SoftPLC Computer: All SoftPLC installations automatically install HILSCHER.TLM and HILSCHER.LST in the SoftPLC CPU. Before proceeding, you should ensure that SoftPLC will successfully start up and that TOPDOC NexGen can communicate to SoftPLC over TCP/IP. For detailed instructions, refer to the SoftPLC Reference Guide.

Laptop Computer: Use the Synergetic CD to install SYCON, the FIELDBUS support program for the bus type you will be using, and the DEVICE DRIVER.

You need *.GSD files for any Profibus slave devices that you intend on communicating with. GSD files are part of the Profibus standard and should be available from your I/O manufacturer. You should copy those files onto your laptop into the ???\SYCON\fieldbus\profibus\ directory.

For Interbus, you may either 1) use existing *.ESD files, 2) make your own *.ESD files, or 3) use the autoconfiguration menu option and skip the ESD file issue altogether.

5. SOFTWARE CONFIGURATION

5.1 Configuring the Hilscher Card(s)

First run the System Configurator (SYCON.EXE) from your laptop. Attach a null modem cable from your laptop's COM1 or COM2 port to the Hilscher card's diagnostics port. This cable as a minimum should have send data, receive data, and return (3 wires minimum) with receive and send data crossing. Normally send data and receive data are pins 2 and 3. See the Hilscher Device Manual if you need more detail on this cable. Basically, the diagnostics port on a Hilscher card has the same pin configuration as a 9 pin RS-232-C connector on a desktop PC. Handshake lines are not used.

In System Configurator's menu system select **Setup | Driver**, and choose "Serial driver" and "Com1" or "Com2" according to which port you have your RS-232-C cable attached on your laptop.

Consult the Hilscher manual on SYCON on how to define your slaves for any particular Hilscher card. Each Hilscher card will need its own configuration.

Profibus

You need *.GSD files for any Profibus slave devices that you intend on communicating with. GSD files are part of the Profibus standard and should be available from your I/O manufacturer.

Interbus

Whereas *.GSD files are part of the Profibus standard and are mandatory, there is no similar standard device specific definition file for Interbus. However, Hilscher has introduced their own file format for this purpose and have given it an extension of *.ESD. Some ESD files come with the SYCON program for some Phoenix devices, however you are unlikely to find all that you would need. There is also a way to generate your own ESD files within SYCON. And lastly, you can skip this step altogether and simply have the card find your configuration from a connected and powered up Interbus network using the Autoconfiguration menu option.

Downloading

After adding each slave to a configuration, you must download the configuration to flash memory on a Hilscher card, and do this process separately for each card. Of course you will have to move the cable between your various cards. There is an online debugger which lets you test the communications **before or after** you have started SoftPLC.

Figure 2 - Important SYCON Settings for Profibus master cards.

Setup | Bus Parameters

Baudrate: <depends on distance and type of I/O modules>

Bp Flag: Auto Clear Mode Off

WatchDog Control/TTR: 1

Setup | Global Parameters

Release Application: No

Watch dog time: <your longest expected SoftPLC program scan * 2>

PLC parameter: no consistence, uncontrolled

Storage format: Intel format

The WatchDog timeout value is in milliseconds and is a very important value. HILSCHER.TLM programs each Hilscher card to use the WatchDog timeout value that you enter here in SYCON for it. When in either run mode or test mode, SoftPLC must communicate with each Hilscher card within its watchdog time interval. In the unlikely event SoftPLC misses this commitment, then that Hilscher card will signal all its slaves that they are to go to their failed position, usually off in the case of outputs. In program mode the WatchDog timer is not in effect.

Once you have a stable master configuration and have downloaded it and tested communications with the SYCON debugger, then you should generate a hardcopy report of your **address list**. The address list is your documentation on how the input area and output area are mapped out. You may also view the address list from SYCON from the **View | Address** List menu.

The columns labeled I-Addr and O-Addr are most important, when **sorted by Address**. They give either the word offset or the byte offset of each slave module, in each of the two areas, input or output.

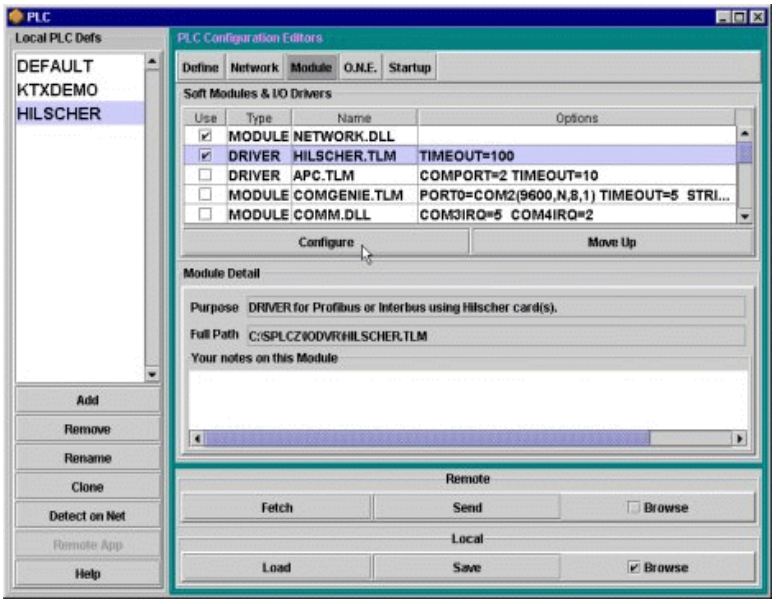
IB = input byte
IW = input word
QB = output byte
QW = output word

It is tricky, but a word offset needs to be multiplied by 2 to get a byte offset. Make sure you are working in byte offsets.

5.2 Configuring the SoftPLC Driver

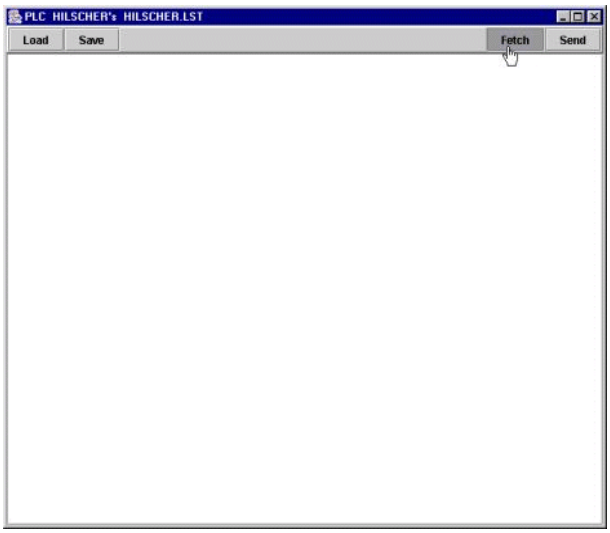
Use TOPDOC NexGen's Module Editor to load and configure the HILSCHER.TLM for use with SoftPLC. If you want to specify a TIMEOUT value, enter it into the options field as shown in Figure 3 (Default is 100).

Figure 3 - Loading the HILSCHER.TLM Driver



Next, click Configure to edit the configuration file HILSCHER.LST. If you do not already have one stored locally for use and are connected to the SoftPLC, click Fetch to retrieve a default file for editing, otherwise click Load. Once you have made the appropriate changes, click Save to store the file on your laptop, and Send to save the changes on the SoftPLC.

Figure 4 - Retrieving the Template Configuration File



After you have closed the text editor and selected and configured any other TLMs necessary, click Save to store the list of TLMs being used, and Send to store it on the SoftPLC.

Figure 5 shows the template HILSCHER.LST file installed with SoftPLC. Note that byte offsets are always used, not word offsets.

Figure 5 - Sample HILSCHER.LST File

```
;This file contains 3 sections: [CARDS], [INPUTS] and [OUTPUTS].
;The terms apply from the perspective of SoftPLC.  Inputs are data
; going into SoftPLC's datatable, and outputs are data copied
; from SoftPLC.  Anything after a semicolon is a comment.

; add one row for each card.  First entry is card "0", 2nd is card "1", etc.
; ISA/PC104: use 20 bit address, e.g. D8000.  (5 digit hex number.)
; PCI: use PCI
[CARDS]
CA000       ;0
;PCI        ;1 PCI cards only supported in version 4 softplc


;Each row in the [INPUTS] and [OUTPUTS] section represents a block of words.
;One or more blocks can be configured for each card.

;SplcWord column: contains the PLC memory address which will either
; receive (if input) or source (if output) the data.  If SplcWord is a
; word corresponding to either the I datatable section or the O datatable
; section, then I/O forcing will be possible.  Otherwise the data will be
; transfered without I/O forcing to a non-I/O image table area.  This
; latter capability can be useful to avoid having analog data eat up the
; valuable I/O image table areas, or as a means of mapping a card which
; is running as a slave.  Example legal values are: N30:0, I12, O33.
; Note that the I and O datatable sections take octal word addresses, so
; I18 would be illegal because of the '8' digit.

;NumWords column: contains the number of 16 bit words to transfer
; starting at the SplcWord to or from the ByteOffset column.  A multiple
; of full words (2 bytes each) must be transferred. Transferring an odd
; number of bytes is not possible.  Use fewer, bigger blocks where
; possible, as opposed to many, smaller blocks, for better speed.

;CardNumber column: tells which card to interface with for this block.

;InputByteOffset or OutputByteOffset columns: gives which !byte! offset
; within the associated card's dual port memory process data image.
; There are two block of process data image on each card, one for
; inputs and one for outputs.  Byte offsets start at 0 in each.  A byte
; offset is 2 times a word offset.

:SwapBytes: put a Y here if you want all the 16 bit words associated
; with this block to be swapped MSB for LSB.

[INPUTS]
;SplcWord,  NumWords,   CardNumber, InputByteOffset     SwapBytes
I:000,      10,         0,          0                  Y
;N10:0       1           1           0                  N


[OUTPUTS]
;SplcWord,  NumWords,   CardNumber, OutputByteOffset    SwapBytes
O:000,      10,         0,          2
;O:001,      1,          1,          0

;END OF FILE
```

Note that the example HILSCHER.LST file uses a SplcWord of N10:0. This is not an I/O address. Moreover, it is memory that may not even exist unless you create it with TOPDOC for each ladder file that you want this SoftPLC to execute. If memory is referenced but does not exist, HILSCHER.TLM will simply ignore the block definition without any indication of error. It is the ladder programmer's responsibility to create this non-I/O datatable memory.

6. TLI's

The HIL_DIAGNOSTICS TIL can be used to grab the "task state 2" 64 byte memory block from a card. These 64 bytes are copied unchanged from a card into a 32 word integer datatable block. See the documentation on this format in Hilscher's dpm_pie.pdf (on their CD) in its chapter 3.

CardNum:	a zero based index identifying which card you want to read
DiagBlk:	an integer block of 32 words (=64 bytes). You should see the Hilscher documention on the format of this block. There are 2 bytes per word, with least significant byte first then most significant byte in each word.

7. OTHER TIPS

- Using the diagnostics port of a Hilscher card increases the processing load on the card and can slow down the I/O scan somewhat, particularly when the I/O bus is running at the fastest baudrates. Use the port only to troubleshoot, in normal operation your laptop won't be connected.
- Improper cabling is a frequent cause of error.
- Many Profibus modules autobaud at power up, so if you change the master's baudrate, plan on cycling power to all the slaves.
- All Hilscher cards are given a "cold reset" command by HILSCHER.TLM at the time SoftPLC starts up. You may do this manually on-demand at any time you like from TOPDOC, simply by doing an operating mode change from REM/PROG mode to REM/PROG mode. Yes that's right, from REM/PROG mode to REM/PROG mode. While this is really no mode change at all, SoftPLC notifies HILSCHER.TLM of the mode change, who treats it as an opportunity to "cold reset" each Hilscher card. Remember that in REM/PROG mode outputs are off.

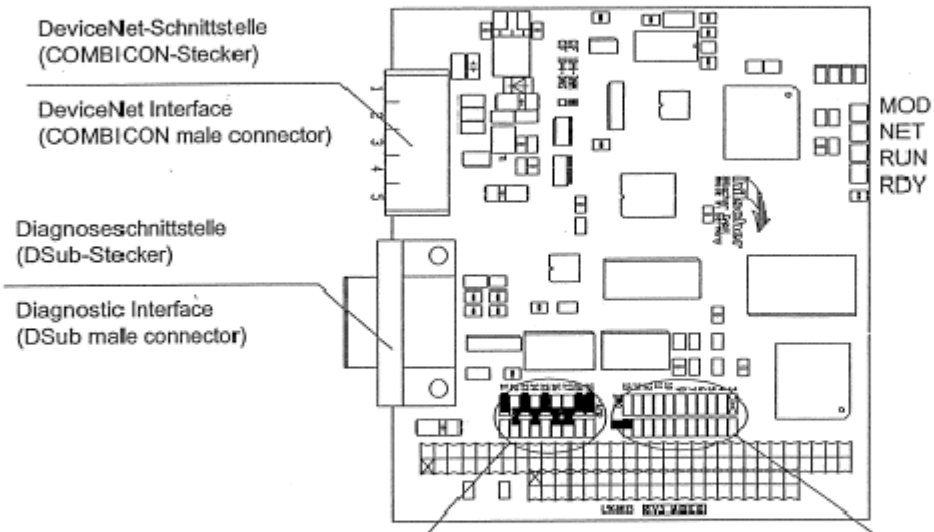
- You'll get slightly better performance by making your master station the lowest station address and then telling SYCON-DP that the **Setup | Bus Parameter | Highest Station Address|** is equal to your master address, normally 1 (one). Highest Station Address is mis-named, it actually refers to the highest station address of any master, and normally there is only one master, namely the current card.

- Last time we looked (Hilscher keeps changing), configuring a **slave** Profibus card requires the use of COMPRO.EXE for the slave database. That configuration must be downloaded with COMPRO to the slave. Then make sure you copy the GSD file for the slave into the \SYCON\FIELDBUS\PROFIBUS\ directory and add it to the master's database, add the Hilscher slave device to the configuration, and download it to the master using SYCON. HILSCHER.TLM supports both masters and/or slaves without having to tell it the difference. Remember that [INPUTS] to a slave are actually coming from the master's [OUTPUTS] and vice versa.

8. DEVICENET SETTINGS

Typ / *Type*	Karte / *Card*	Dual-Port Memory	Funktion / *Function*
CIF 104-DNS	PC/104	8 KByte	DeviceNet-Slave
CIF 104-DNS-R	PC/104	8 KByte	DeviceNet-Slave *
CIF 104-DNM	PC/104	8 KByte	DeviceNet-Master
CIF 104-DNM-R	PC/104	8 KByte	DeviceNet-Master *

* Karte mit Stecker an der rechten Seite nicht abgebildet / *boards with connector on the right side are not shown*



Option
Pfostenverbinder
anstelle von
DSub-Stecker

Option
Square post
connector instead
of a DSub connector *

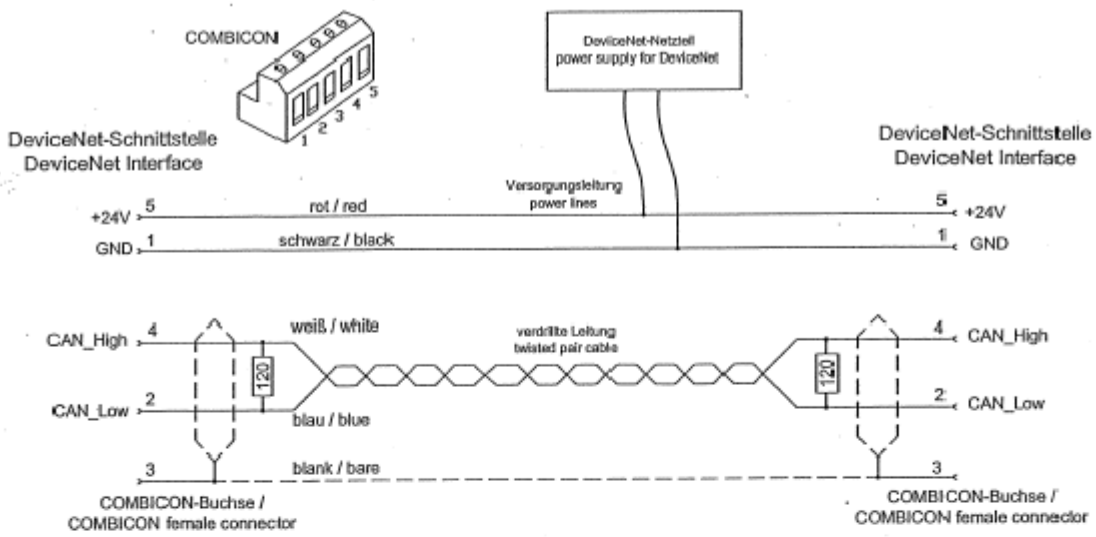
\P	10-polig gewinkelt *10 pin angled*
\S	10-polig gerade *10 pin straight*
\H	16-polig gewinkelt *16 pin angled*
\V	16-polig gerade *16 pin straight*

Adresse / *Address*	A13	A14	A15	A16	A17	A18	A19
CA000		**X**		**X**	**X**		
CC000	X			X	X		
CE000				X	X		
D0000	X	X	X		X		
D2000		X	X		X		
D4000	X		X		X		

X = *Jumper closed*

Interrupt	I15	I14	I12	...	I3
Kein / *no*					
15	X				
14		X			
12			X		
...					
3					X

Default configuration



Sie können bis zu 64 DeviceNet-Geräte über den Bus miteinander verbinden. Die maximale Länge des Buskabels ist abhängig von der verwendeten Baudrate und dem Kabeltyp. Bitte verwenden Sie nur speziell für DeviceNet zugelassenes Kabel.

Stromversorgungskabel* *Power supply cable**	dick *thick*	dünn *thin*	
Schleifenwiderstand *Loop resistance*	<11,8	<57,4	Ohm/km
Aderndurchmesser *Wire gauge*	2 x 1,4	2 x 0,7	mm

Baudrate *Baud rate*	Max. Länge bei Kabeltyp *Max. distance with cable type*	
	dick */ thick*	dünn */ thin*
125 kBit/s	500 m	100 m
250 kBit/s	250 m	100 m
500 kBit/s	100 m	100 m

Up to 64 DeviceNet devices can be linked together over the bus. The maximum length of the bus cable depends on the used baud rate and the used cable type.
Only special proved DeviceNet cable should be used.

Datenleitung* *Data line cable*	dick *thick*	dünn *thin*	
Wellenwiderstand *Impendance*	120	120	Ohm
Kapazitätsbelag *Capacity*	<39,4	<39,4	pF/m
Schleifenwiderstand *Loop resistance*	<22,6	<91,8	Ohm/km
Aderndurchmesser *Wire gauge*	2 x 1,1	2 x 0,5	mm

* Das DeviceNet-Buskabel besteht aus den Daten- und den Spannungsversorgungsleitungen.
The DeviceNet cable contains of the data line cables and the power supply cables.

Diagnoseschnittstelle

Nicht auf PCMCIA-Karten und PMC-Modulen

Potentialgebundene RS-232C-Schnittstelle zum Anschluss an die COM-Schnittstelle des PCs.

Diagnostic Interface

Not at PCMCIA cards and PMC modules

Non isolated RS-232C interface to connect with the COM port at the PC.

DSub-Stecker 9-polig *DSub male connector 9 pin*	Pfostenverbinder 10-polig *square post connector 10 pin*	16-polig *16 pin*	Signal *Signal*	Bedeutung *Meaning*	Eingang/Ausgang *Input/Output*
2	3	7	RXD	Empfangsdaten / *Receive Data*	Eingang / *Input*
3	5	9	TXD	Sendedaten / *Send Data*	Ausgang / *Output*
4	7	11	DTR	Datenendeinrichtung betriebsbereit / *Data Terminal Ready*	Ausgang / *Output*
5	9	13	GND	Betriebserde / *Signal Ground*	
(6) n.v. / *n.c.*	n.v. / *n.c.*	n.v. / *n.c.*	DSR	Betriebsbereitschaft / *Data Set Ready*	Eingang / *Input*
7	4	8	RTS	Sendeteil einschalten / *Ready to Send*	Ausgang / *Output*
8	6	10	CTS	Sendebereitschaft / *Clear to Send*	Eingang / *Input*

n.v. nicht verwendet / *n.c. not connected*