SICK-STEGMANN GmbH
Dürrheimer Str. 36
78166 Donaueschingen
Telefon 0771/807-0
Telefax 0771/807-100

Comissioning Instructions

Hiperface-Devicenet-Adapter

to DeviceNet specification release 2.0 vol. 1 and 3

Feb. 2007

The data in this documentation describes the product. The properties, however, are not guaranteed.

SICK-STEGMANN GmbH
Dürrheimer Straße 36
D-78166 Donaueschingen

Phone: (49) 771 / 807 – 0
Fax: (49) 771 / 807 – 100
Web: http://www.sick.com
e-mail Info@sick.de

Version: Feb. 2007

We reserve the right ot make technical changes to the documentation and the products at any time.

. Content

. Schedule of Revisions

The following information lists the revisions made by the company since the first edition.

New information (N)

New properties and additional information on existing properties.

Revised information (R)

Revisions to the previous edition, which require a different procedure during commissioning.

Inf.	Revisions	Chapter	Rev	Date
	First edition of he document		1.00	Feb. 2007

1 Introduction

1.1 Using these Commissioning Instructions

These instructions give an overview of the Hiperface-Devicenet-Adapter and describe configuration, installation, operation and maintenance of the devices in the Devicenet network.

These instructions have been written for trained persons who are responsible for installation, assembly and operation of the Hiperface-Devicenet-Adapter in industrial environments.

The commissioning staff should understand Devicenet network operations and know how slave devices fuction in a network and communicate with a Bus Master. There should be basic understanding of and experience in electrical terminology, programming procedures, networks, the devices and software required as well as knowledge of applicable safety regulations.

1.2 Documentaion

The complete documentation comprises of:

- Comissioning Instructions for the Hiperface-Devicenet-Adapter (*)
- Electronic Data Sheet (EDS-File), Hiperface-Devicenet-Adapter Bitmap
- Assembly instructions for the Hiperface-Devicenet-Adapter
- Data sheet of the Hiperface-Devicenet-Adapter
- Assembly instructions for the particular encoder
- Data sheet of the particular encoder
- Hiperface® - interface description

(*) This documentation

1.3 Definition of the symbols used in the document

	This symbol identifies text wihich must be given special attention, to ensure correct use of devices and prevent dangers resulting from this. Non-compliance may lead to bodily harm and/or damage or destruction of the system.

	This symbl identifies text with important information on the correct use of the encoders. Non-compliance may lead to Hiperface-Devicenet-Adapter malfunctions.

	This symbol identifies text with useful details or explanations for the better understanding of this documentation.

1.4 Conformity

1.4.1 Devicenet

- Devicenet interface to ISO 11898 (CAN high speed) and CAN specification 2.0B, electrically isolated according to Devicenet protocol specification Release 2.0 – Volume 1 and 3.
- Profile for encoders (encoder device type 22_h) based on the "Position Sensor Object (class code 23_h)" – see device profile to Devicenet spezification Vol. 1, Chapter 6 and 5.23.

1.4.2 European Union Directive Compliance

Declaration of Conformity

1.4.3 United States Standards - Listed

UL - Authorization

Listed accessory which is to use with listee's listed SICK-STEGMANN GmbH encoders. For use in NFPA 79 applications only. Interconnection cables and accessories are available from Sick-Stegmann.

1.4.4 Hiperface

- Sick-Stegmann Hiperface® interface

2 Safety Precautions

2.1 Responsibilities of the Commissioning Staff

Due to the various possible applications of the Hiperface-Devicenet-Adapter described in this documentation, those responsible for the application and use of the devices must personally ensure that all necessary measures are taken so that the application and use of devices meets the performance and safety requirements. The applicable laws, regulations, guidelines and standards shall be taken into account.

The Hiperface-Devicenet-Adapter described in this documentation is intended for use under industrial conditions. Non-compliance with applicable safety regulations, process-related and other instructions can lead to bodily harm and/or damage to the devices or the system. Qualified users or commissioning staff for these devices are personally responsible for the selection of suitable products for the intended application. SICK-STEGMANN GmbH accepts no liability and no responsibility for direct or indirect consequential damage due to improper handling or incorrect selection of procucts. Poper handling assumes that the instructions in this documentation are followed.

Qualified users or commissioning staff are those familiar with the safety requirements and their application to installation, operation and maintenance of the devices.

It is advisable for persons operating or maintaining electrical or mechanical devices to have basic knowledge of First Aid.

2.2 Validity and Application

Hiperface-Devicenet-Adapter are complex interface converters for Hiperface Encoders. They are manufactured according to the known industrial regulations and meet the ISO 9001 quality requirements.

An Hiperface-Devicenet-Adapter is a device to be fitted and cannot be independently operated according to its intended function. Consequently, an Hiperface-Devicenet-Adapter is not equipped with direct safety features. The designer must provice measures for the safety of persons and systems, in accordance with statutory requriements.

The Hiperface-Devicenet-Adapter must be oprated according to its type-specific purpose and only within a Devicenet network.

The Devicenet specifications and guidelines for the cnstruction of a Devicenet network must be complied with.

Mechanical or electrical modifications of the devices are not permitted.

☞	**The safety precautions and instructions for installation and operation in this document, are binding.**

2.3 Authorised Users

Installation and maintenance of the Hiperface-Devicenet-Adapter and the corresponding encoder must be performed by trained and qualified staff (as described under point **Fehler! Verweisquelle konnte nicht gefunden werden.**) with knowledge of electronics, perecision engineering and controller programming. The appropriate standards of technical safety requirements must be met.

2.4 Safety Guidelines and Personal Protection

The safety guidelines must be observed by all persons charged with installation, operation or maintenance of the devices:

- The system and safety documentaion must always be available and observed.
- During installation and maintenance, non-qualified staff must not be near the system.
- The system must be installed in accordance with applicable safety requirements and instructions.
- The installation must meet the accident prevention regulations of the professional/trade associations in the country/countries concerned.
- Non-compliance with the relevant safety and accident prevention regulations can lead to bodily harm or system damage.

2.5 Hiperface-Devicenet-Adapter Safety Precautions for Installation, Operation and Maintenance

- The Hiperface-Devicenet-Adapter must only be fitted or maintained when free from voltage. Conseqently, for any electrical or electronic devices, machine and system parts linked to installation operation or maintanance, the voltage must previously have been switched off. Also check that they are voltage-free.
- Cabeling, earthing, screening and overvoltage protection are of particular importance.
- Check whether switching off devices, machines or system parts causes dangers.
- If necessary, place warning signs to prevent unauthorised persons from entering the danger area.
- The correct functioning of safety features shall be checked (e.g. emergency stopp).

- EMC-compliant earthing and screening must be prformed with particular care, to ensure satisfactory operation of the devices.
- The safe assembly of all components must be checked prior to switching back on.

- During commisioning of the encoders via a configuration tool, there must be no connection to a running network.

☞	• Hiperface-Profibus-Adapters must not be exposed to direct UV radiation, over long periods. Resistance to other environmental effects may be found in the data sheet. • The rating label of the Hiperface-Profibus-Adapters has a sealing function and must not be damaged in the area of the viewing window (e.g. through the effect[s] of pointed objects). • Knocks on shaft an collet of encoders must be avoided. • Programs which frequently use Explicit Messages, in order to write parameters to the EEPROM (non-volatile memory) of the encoders, should be avoided. This could significantly reduce the lifetime of the product.

3 System overview

Hiperface stands for High Performance Interface and is a standard interface for motor feedback systems (encoders) from SICK-STEGMANN GmbH . In order to also make these encoders avaiable for applications in factory automation a link to a PLC with Devicenet interface can be implemented via a Hiperface-Devicenet-Adapter.

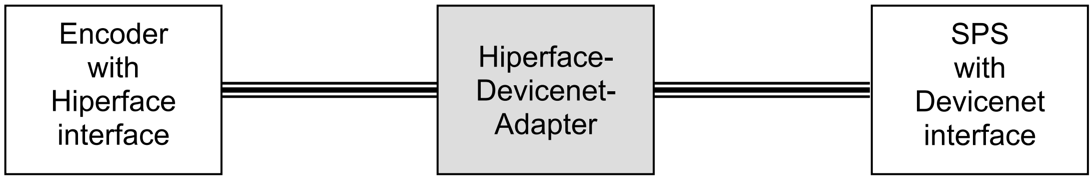


Figure 3–1: System overview – Hiperface on Devicenet.

4 Designations and abbreviations

4.1 General

CAN	Controller Area Network	
CiA	CAN in Automation.	
CIP	Common Industrial Protocol	
ODVA	Open DeviceNet Vendor Association.	

4.2 Data Type Specifications

BOOL	Boolean	1 Bit
BYTE	Bit String	1 byte (8 bits)
WORD	Bit String	2 bytes (16 bits)
USINT	Unsigned Short Integer	Int (1 bytes) - (0...255)
UINT	Unsigned Integer	Int (2 bytes) - (0...65,535)
UDINT	Unsigned Double Integer	Int (4 bytes) - (0...+2^{32}.-1)
ENGUNIT	Engineering Unit	2 bytes Ex: subunit of micrometres [µm] represented in the base unit 0xhhhh*10^{-6}m
SINT	Signed Short Integer	Int (1 bytes) - (-128...+127)
INT	Signed Integer	Int (2 bytes) - (-32,768...+32,767)
DINT	Signed Double Integer	Int (4 bytes) - (-2^{31}...+2^{31} - 1)
LSB	Least Significant Bit / Byte	Example: $81,938_D$ == LSB
MSB	Most Significant Bit / Byte	MSB
Little Endian	In a sequence of data packets requested/sent in the Devicenet protocol, the LSB is transmitted first. Also see LSB above: { 12.40.01.00 }	

4.3 Devicenet - spezifisch

ID	Identifier
EDS	Electronic Data Sheet – An electronic data sheet is a template provided by the manufacturer, which specifies how information is represented and which appropriate entries can be made.
PM-SC	Predefined Master / Slave Connection Set
Node	A node is a hardware unit with a unique address in the network (also called device).
MAC ID	Address of a Devicenet node
Explicit Messaging	Explicit messages are used for general data exchange between two devices via the Devicenet network. This type of communication is always a point-to-point connection in a Client/Server system, where the query from a client must always be acknowledged by a server (request/response).
IO (I/O)	Input and output data
Input Data	This data is produced by a Devicenet device (slave device) and collected by a master device and made available, for reading, to the processor of the PLC.
Output Data	This data is produced by a PLC processor and written to the memory of the master device. The master device sends this data to other Devicenet devices (slave devices).

4.4 Hiperface-Devicenet-Adapter -Specific or Encoder-Specific

CPR	Counts (steps) per Revolution (resp. span for linear devices). -- *Customer specified* --
CMR	Counts (steps) over the total Measuring Range. -- *Customer specified* --
R	Ratio (R) = [CMR] / [CPR]
Scaling Para- meters	Skalierungsparameter: [CMR], [CPR]

PRS	Physical Resolution Span: max. number of steps per revolution/span supported by the encoder – *specified by manufacturer.* --
PnumRev	Physical Number of Revolutions: Max. number of revolutions supported by the encoder – *specified by manufacturer. --*
PMR	Physical Measuring Range (PRS x PnumRev): Total number of steps across all revolutions or across the entire span – *specified by manufacturer* --
PM_Bit	According to 'PMR', as bit mapping with basis (2^{**PM_Bit}). Corresponds to (26) for rotary encoders, to (24) for linear ones.
PmaxVal	Maximum position value is 'PMR' minus one (1).

ScF	Scaling factor = [CPR] / [PRS]
Pos_Scal	Scaled position value after conversion is performed by the scaling parameters, offset and preset values
Pos_Phy	Physical (numerical) position value prior to conversion
CPS	Encoder speed: "Counts (steps) per sec." – hence (cps, CpS)
RPS	Encoder speed: "Revolutions per sec." – hence (rps, RpS)
RPM	Encoder speed: "Revolutions per min." – hence (rpm, RpM)
mm/sec	Encoder speed: millimetres per second

5 Devicenet – General

5.1 Introduction

Devicenet is a field bus based on the CAN medium (to ISO 11898 – High Speed) mainly used in automation technology. Devicenet was developed by Allen Bradley and later handed to ODVA (Open Devicenet Vendor Association) as an open standard. ODVA administers the Devicenet specification and drives its development forward. Devicenet is widely used in Asia and the USA.

Devicenet essentially uses, at the lower layers 1-4 of the OSI layer model, the CAN specifications with some limitations and additions. For example, Devicenet does not allow all baud rates known to CAN. At the upper layers 5-7, Devicenet uses the CIP defined by ODVA. Thus, Devicenet is the implementation of CIP on the basis of CAN. CIP is also used for the field bus Ethernet IP.

Device profiles ensure the interoperability of the devices in the network and the interchangeability of the devices.

5.2 Topology and Specifications

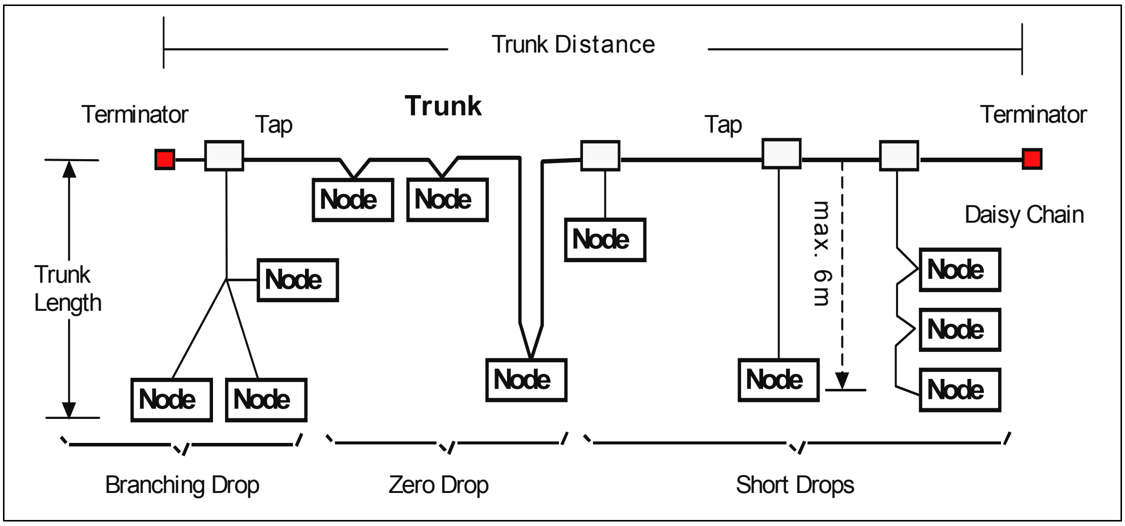


Figure 5–1: Topologie of Devicenet

☞	**Only cables according to Devicenet specification may be used for network cabling**.		
	Core assignments:		
	CAN_L	Blue	
	CAN_H	White	
	V--	Red	
	V+	Black	
	Drain	coloruless braid	

- Line structure topology with trunk line and drop line configuration
- Devicenet allows taps as drop lines only.
- Up to 64 nodes are supported for each network.
- Terminators = 120 ohms at the two – physically – most distant end points of the trunk line of a network

- Screened and twisted pair cables are the physical medium for signal transmission and power supply.
- The device's power supply in the network depends on the cable type [thick cable: 8 A, thin cable: 3 A].
- The M12 5-way plug types ('A' coded) are meant for the Hiperface Devicenet Adapter.
- The bus coupler and encoder device are protected against incorrect wiring.
- Devicenet is based on the standard CAN protocol and thus uses an 11-bit message identifier.
- Max. cable lengths for the trunk line and the drops lines – see table below:

Data rate	Trunk length		Drop length	
	Thin cable	Thick cable	Single drop line	Accumulated thin cable
125 KBaud	100 m	500 m	6 m	156 m
250 KBaud	100 m	250 m	6 m	78 m
500 KBaud	100 m	125 m	6 m	39 m

The line length between the two most distant points in the cabling system must not exceed the length allowed for the baud rate selected. The cable length between two points thus includes both the trunk line and the drop lines. The drop line length is measured by taking the distance from the tap of the trunk line to the respective transceiver of the corresponding node in the drop line.

5.3 Device Profiles

Devicenet describes all data and functions of a device using an object model. In practice, there usually is a combination of Devicenet standard objects (to ensure the basic interoperability with the Devicenet network) and application-specific objects which are defined in a device profile. For the Hiperface Devicenet Adapter, this is the device profile for encoders. The Devicenet specification contains the library of the objects and the device profiles.

Devicenet specification vol. 1, chapter 6. – Device profile for encoders (Encoder Device Type 22h)

5.4 Further Information Sources

ODVA für DeviceNet
20423 State Road 7
Boca Raton, FL 33498, USA
Tel.: (1) 954-340-5412
email: odva@powerinternet.com
Web: http://www.odva.org

CAN in Automation (CiA) e.V.
Am Weichselgarten 26
D-91058 Erlangen
(49) 9131 - 6 90 86-0
http://www.can-cia.de

The information from ODVA, the Devicenet users' organisation, is usually available in English only.

6 Devicenet Object Model

6.1 Terminology

Devicenet describes all data and functions of a device through an object model. Thus, the functions of a device are mapped using a packet of objects.

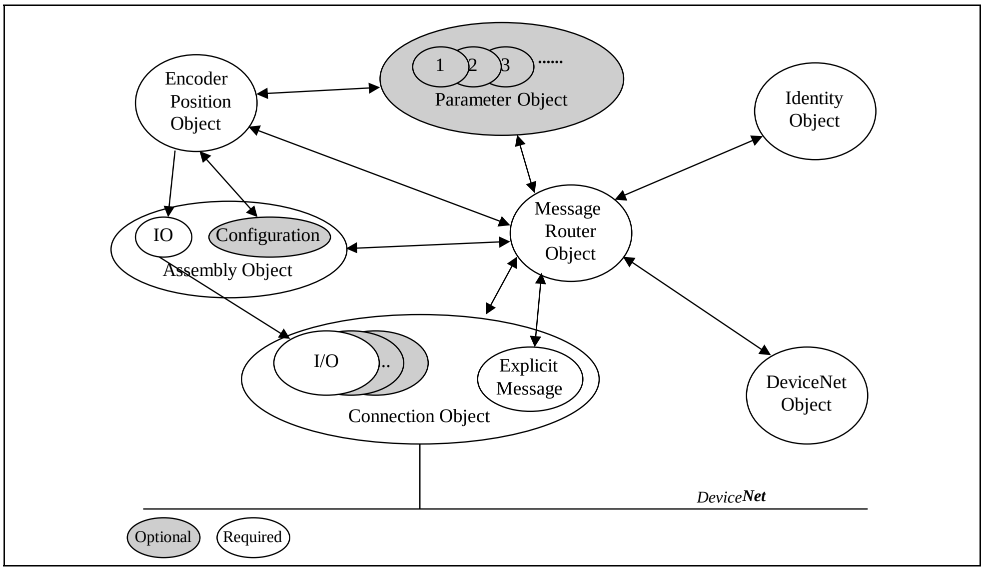


Figure 6–1: Devicenet Object Model

Objektbestandteile:

Designa-tion	Explanation
Object	An object itself is an abstract representation of a component within a device. A distinction is made between communication objects (connection objects), system objects (general functions) and application-specific objects (special device functions – e.g. the Position Sensor Object)
Instance	Characteristic feature of an object
Services	Services are used to access certain attributes or instances of an object (e.g. reading/writing).
Properties/Attributes	Properties represent data made available by an object for a device, via Devicenet.

6.2 Object Addressing

6.2.1 Object Addressing in General

The data, and the services of an object performed on it, are addressed via a hierarchical addressing concept with the components shown in the table:

Component	Description
MAC ID	Device address in the network (each address must only occur once)
Class ID	Addresses each class, which can be accessed from within the network, of an object

Component	Description
Instance ID	Addresses the individual instances of an object and thus identifies the instances of the same class.
Attribute ID	Addresses the individual properties (attributes) of each class and/or an instance.

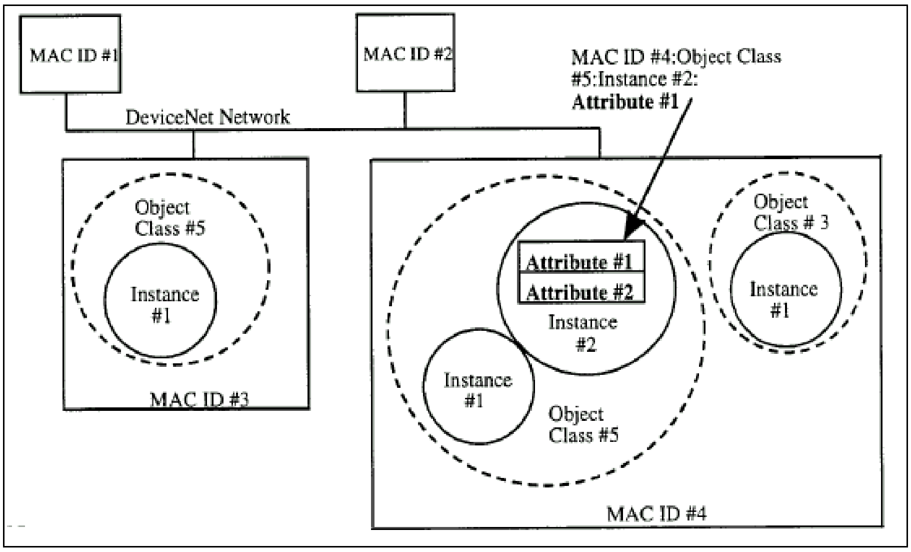


Figure 6–2: Object Addressing Schematic

6.3 Vom Hiperface-Devicenet-Adapter unterstützte Objekte:

Designation	Class ID	Explanation	
Identity object	01h	Contains information of the node in the network	see 6.3.1
Message router object	02h	Processes all messages and forwards these to the corresponding objects	see 6.3.2
Devicenet object	03h	Administers the configuration and the status of the physical connection (MAC ID, baud rate etc.)	see 6.3.3
Assembly object	04h	Assembles the attributes (data) of different objects in one object. Is used for I/O messages.	see 6.3.4
Connection object	05h	Manages the internal resources allocated to the I/Os and explicit message links.	see 6.3.5
Acknowledge handler object	2Bh	Controls receipt of the "Acknowledge Messages" for message-generating objects, e.g. COS (Change of State)	see 6.3.6
Position sensor object	23h	Administers the device-specific data from the Hiperface Devicenet Adapter, such as position and direction of rotation	see 6.3.7

6.3.1 Identity Object

Class ID	Class Attributes	Instance Attributes	Number of Instance	Services
01_{hex}	Yes (1, 2)	Yes	1	Yes

6.3.1.1 Class Attributes Relating to the Identity Object

Attr. ID	Attribute Name	Access Rule	Type	Description
1	Revision	Get	UINT	Revision of the Identity Object (= 01)
2	Max Instance	Get	UINT	Highest instance number of an object generated in this class (= 01)

6.3.1.2 Instance Attributes Relating to the Identity Object

Attr. ID	Attribute Name	Access Rule	Type	Description	
1	Vendor ID	Get	UINT	511	SICK-STEGMANN
2	Device Type	Get	UINT	**22** $_{hex}$	
3	Product Code	Get	UINT	65	
4	Revision	Get	Struct of USINT	1,00	major / minor revision
5	Status	Get	WORD	Status of the complete device acc. to DN Spec. Chapter 6	
6	Serial Number	Get	UDINT	32-bit deposited in the EEPROM of the Hiperface Devicenet Adapter (data format see section 10.2)	
7	Product Name	Get	SHORT_ STRING	Hiperface Comm Adapter DN	
8	State	Get	USINT	Status of the State Machine 0: Nonexistent 1: Self Testing 2: Stand By 3: Operational 4: Major Recoverable Fault 5: Major Unrecoverable Fault	
9	Configur. Consistency Value (CCV)	Get	UINT	Counter value – is incremented after each write operation into the EERPROM	
10	Heartbeat Interval	Set	USINT	Time interval {1…255} in sec. for sending the heartbeat message. If the heartbeat time set ≠ 0, the heartbeat protocol is used instead of the cyclic query of the nodes by a superior instance. The adapter then is the heartbeat producer and signals its communication status by sending the heartbeat message.	

6.3.1.3 General Services Relating to the Identity Object

Service Code	Service Name	Description
05hex	Reset	Reset service of the device Value 0: standard reset Value 1: reset to “as delivered”
0Ehex	Get_Attribute_Single	Delivers the content of the specified attribute
10hex	Set_Attribute_Single	Changes the content of the specified attribute

Identifier (bit 10...0)			Data (8 bytes)				**--------->(Request)**			
Gr	MAC ID	MSG ID	Head	Service	Class	Inst	Data 0	Data 1	Data 2	Data 3
10	111111	100	01	05	01	01	nn			

MSG ID	[04]: Master's Explicit Request Message
Head	Frag-Bit = 0, XID-Bit = 0, Source MAC-ID (Master)
Service	"Reset"
Class	Identity Object
Instance	Instance ID of Class
Data 0	Type of Reset (Value: 0, 1). -- USINT {0x00 }, {0x01 }.

6.3.2 Message Router Object

Class Code	Class Attributes	Instance Attributes	Number of Instance	Services
02hex	No	Not supported	1	Not supported

6.3.3 Devicenet Object

Class Code	Class Attributes	Instance Attributes	Number of Instance	Services
03hex	Yes (1)	Yes	1	Yes

6.3.3.1 Class Attributes Relating to the Devicenet Object

Attr. ID	Attribute Name	Access Rule	Type	Description
1	Revision	Get	UINT	Revision of the Devicenet object (= 02)

6.3.3.2 Instance Attributes Relating to the Devicenet Object

Attr. ID	Attribute Name	Access Rule	Type	Description
1	MAC-ID	Get / Set	USINT	Node address [Node ID] (0 – 63) (only if DIP-Switch S2-DIP1 = ON)
2	Baud Rate	Get / Set	USINT	Data rate (0 – 2) --- (only if DIP-Switch S2-DIP2…5 in Pos. EEPROM)
3	BOI	Get	BOOL	Bus-OFF Interrupt (Default = 0)
4	Bus-OFF-Ctr.	Get / Set	USINT	Quantity how often CAN has changed to the Bus-OFF state
5	Allocation Infor-mation	Get	Struct of: USINT	Allocation choice Byte and MAC-ID of the master
6	MAC-ID Switch	Get	BOOL	Node address switches have been

Attr. ID	Attribute Name	Access Rule	Type	Description
	changed			changed since the last re-start/reset
7	Baud Rate Switch changed	Get	BOOL	Baud rate switches have been changed since the last restart/reset
8	MAC-ID Switch Value	Get	USINT	Current value of the node address switches
9	Baud Rate Switch Value	Get	USINT	Current value of the Baud rate switches

6.3.3.3 General Services Relating to the Devicenet Object

Service code	Service name	Description
$0E_{hex}$	Get_Attribute_Single	Delivers the content of the specified attribute
10_{hex}	Set_Attribute_Single	Changes the content of the specified attribute

6.3.3.4 Class-Specific Services Relating to the Devicenet Object

Service code	Service name	Description
$4B_{hex}$	Allocate_Master / Slave Connection_Set	Requests the use of the specified connection(s)
$4C_{hex}$	Release_Master / Slave Connection_Set	Indicates that the specified connections (PM-SC) are to be released

6.3.4 Assembly Object

Class Code	Class Attributes	Instance Attributes	Number of Instance	Services
04_{hex}	Yes (1)	Yes	4	Yes

6.3.4.1 Class Attributes Relating to the Assembly Object

Attr. ID	Attribute Name	Access Rule	Type	Description
1	Revision	Get	UINT	Revision of the Assembly Object (= 02)

6.3.4.2 Instance Attributes Relating to the Assembly Object

Attr. ID	Attribute Name	Access Rule	Type	Description
3	Data	Get	Array of BYTE	Depending on the instance: [1]: Position value [2]: Position value + flags [3]: Position value + velocity value [100]: Position value + cam status
		Get / Set		[101]: Preset

6.3.4.3 General Services Relating to the Assembly Object

Service code	Service name	Description
$0E_{hex}$	Get_Attribute_Single	Delivers the content of the specified attribute
10_{hex}	Set_Attribute_Single	Changes the content of the specified attribute

6.3.4.4 Input and Output Assembly Data Components

The attributes from different objects are summarised in an Assembly Object.

Datenbaustein	Object / Class		Instance	Attribute	
	Name	Number	Number	Name	Name
Position value	Pos. Sensor Object	$0x23_{hex}$	1	Position value	10_{dec}
Warning flag	Pos. Sensor Object	$0x23_{hex}$	1	Warning flag	49_{dec}
Alarm flag	Pos. Sensor Object	$0x23_{hex}$	1	Alarm flag	46_{dec}
Speed value	Pos. Sensor Object	$0x23_{hex}$	1	Velocity value	24_{dec}
Cam status	Pos. Sensor Object	$0x23_{hex}$	1	CAM status register	35_{dec}

6.3.4.4.1 Input Assembly Data Format

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	0	Position value							
	1								
	2								
	3								

Definition of the position value (4 bytes) with a maximum resolution of 'nn' bits. -- (2^{**nn}).

Byte_3 (MSB)	Byte_2	Byte_1	Byte_0 (LSB)
31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 09 08	07 06 05 04 03 02 01 00
	Number of revolutions (1...4096)		CPR (1...4096) - [steps per rev.]
	[CMR]		

Byte_3 (MSB)	Byte_2	Byte_1	Byte_0(LSB)
31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 09 08	07 06 05 04 03 02 01 00
	Number of revolutions (1…4096)	CPR (1…32768) - [steps per rev.]	
	[CMR]		

Byte_3 (MSB)	Byte_2	Byte_1	Byte_0 (LSB)
31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 09 08	07 06 05 04 03 02 01 00
	Number of revolutions (1...4096)	CPR (1...262144)-[steps per rev.]	
	[CMR]		

- For single-turn encoders, the “Number of revolutions” column is dropped.
- CMR is definitive for linear encoders.

Example: 4,096 x 262,144 >> output values 0 … 1,073,741,823 -- view as UDINT value in the "Little Endian" data format.

Byte_3 (MSB)	Byte_2	Byte_1	Byte_0 (LSB)	
3 F	F F	F F	F F	$1,073,741,823_D$ == $[3F,FF,FF,FF]_{hex}$

For the subsequent instances (if required), the triggering to the position value can be prevented in the COS/Cyclic operating mode (i.e. the position value is supplied, but a position change does not trigger a new message). This is controlled by attribute 20 (Cos Delta) in the Position Sensor Object (6.3.7.2). For this, the value for COS Delta shall be set to the maximum value (thus the value is greater than the set measuring range). Hence, for example, triggering is only possible to the velocity value for instance 3.

<table>
<tr><th>Instance</th><th>Byte</th><th>Bit 7</th><th>Bit 6</th><th>Bit 5</th><th>Bit 4</th><th>Bit 3</th><th>Bit 2</th><th>Bit 1</th><th>Bit 0</th></tr>
<tr><td rowspan="6">2</td><td>0</td><td colspan="8" rowspan="4">Position value</td></tr>
<tr><td>1</td></tr>
<tr><td>2</td></tr>
<tr><td>3</td></tr>
<tr><td rowspan="2">4</td><td colspan="8">Flag</td></tr>
<tr><td></td><td></td><td></td><td colspan="3">Reserved</td><td>Warn</td><td>Alarm</td></tr>
</table>

Triggering the flag through a status change 0 <--> 1 of a warning or an alarm.

<table>
<tr><th>Instance</th><th>Byte</th><th>Bit 7</th><th>Bit 6</th><th>Bit 5</th><th>Bit 4</th><th>Bit 3</th><th>Bit 2</th><th>Bit 1</th><th>Bit 0</th></tr>
<tr><td rowspan="8">3</td><td>0</td><td colspan="8" rowspan="4">Position value</td></tr>
<tr><td>1</td></tr>
<tr><td>2</td></tr>
<tr><td>3</td></tr>
<tr><td>4</td><td colspan="8" rowspan="4">Velocity value</td></tr>
<tr><td>5</td></tr>
<tr><td>6</td></tr>
<tr><td>7</td></tr>
</table>

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
100	0	Position value							
	1								
	2								
	3								
	4	CAM Status Channel 1							
		Cam 8	Cam 7	Cam 6	Cam 5	Cam 4	Cam 3	Cam 2	Cam 1
	5	CAM Status Channel 2							
		Cam 8	Cam 7	Cam 6	Cam 5	Cam 4	Cam 3	Cam 2	Cam 1
	6	CAM Status Channel 3							
		Cam 8	Cam 7	Cam 6	Cam 5	Cam 4	Cam 3	Cam 2	Cam 1
	7	CAM Status Channel 4							
		Cam 8	Cam 7	Cam 6	Cam 5	Cam 4	Cam 3	Cam 2	Cam 1

For further explanations regarding the electronic cam switching mechanism see 6.3.8.3

6.3.4.4.2 Ausgangs- (Output) Assembly Datenformat

This output data assembly had been generated in order to be able to execute a transmission of the preset value in a similarly comfortable way as in case of devices with Profibus interface.

If during the transmission, additionally to the preset value to be transmitted, the MSB is set to '1' , this new value will be taken over as the new preset value and an internal offset will be generated.

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
101	0	Preset value							
	1								
	2								
	3								

Display after preset, such as preset value = 0 and the set counting sequence CW

Direction of rotation	(reverse) <<				Preset	>> (forward)			
Display	-4	-3	-2	-1	0	1	2	3	4

6.3.5 Connection Object

Class Code	Class Attributes	Instance Attributes	Number of Instance	Services
05_{hex}	Yes (1)	Yes	4	Yes

The following "Connection Objects" (Inst. IDs) are implemented in the Hiperface-Devicenet-Adapter:

Inst. ID	Instances
1	Group 2 Explicit Message Connection
2	Poll IO Connection
3	Bit Strobed Connection
4	Change of State / Cyclic Connection

6.3.5.1 Class Attributes Relating to the Connection Object

Attr. ID	Attribute Name	Access Rule	Type	Description
1	Revision	Get	UINT	Rev. des Connection Objektes (= 01)

6.3.5.2 Instance Attributes Relating to the Connection Object

Attr. ID	Attribute Name	Access Rule	Type	Description
1	State	Get	USINT	Status of the connection
2	Instance_Type	Get	USINT	I/O or Explicit Message
3	Transportclass_trigger	Get	BYTE	Behaviour of the connection
4	Produced_connection_id	Get	UINT	Connection Identifier for the producer
5	Consumed_connection_id	Get	UINT	Connection Identifier for the consumer
6	Initial_comm_characteristics	Get	BYTE	The 'Message Groups' assigned to this connection
7	Produced_connection_size	Get	UINT	Number of bytes which can be transmitted via this connection
8	Consumed_connection_size	Get	UINT	Number of bytes which can be received via this connection
9	Expected_packet_rate	Get / Set	UINT	Timing allocated to this connection
12	Watchdog_timeout_action	Get	USINT	Defines how to proceed in case of standstill / watchdog timeouts
13	Produced_connection_path length	Get	UINT	Number of bytes in the attribute 'Produced_connection_path'
14	Produced_connection_path	Get / Set	ARRAY of EPATH	Defines the 'Application Object' whose data is produced through this connection
15	Consumed_connection_path length	Get	UINT	Number of bytes in the attribute 'Consumed_connection_path'
16	Consumed_connection_path	Get / Set	ARRAY of EPATH	Defines the 'Application Object' whose data is consumed through this connection
17	Production_inhibit_time	Get / Set	UINT	Smallest time interval until a Change of State connection is allowed to produce data again

6.3.5.3 General Services Relating to the Connection Object

Service code	Service name	Beschreibung
05_{hex}	Reset	Resets the 'Standstill Watchdog Timer' of the Connection Object
$0E_{hex}$	Get_Attribute_Single	Delivers the content of the specified attribute
10_{hex}	Set_Attribute_Single	Changes the content of the specified attribute

6.3.6 Acknowledge Handler Object

Class Code	Class Attributes	Instance Attributes	Number of Instance	Services
$2B_{hex}$	Yes (1, 2)	Yes	1	Yes

6.3.6.1 Klassen Attributes Relating to the Acknowledge Handler Object

Attr. ID	Attribute Name	Access Rule	Type	Description
1	Revision	Get	USINT	Revision of the Acknowledge Handler Object (= 01)
2	Max. Instance	Get	USINT	Highest instance number of an object produced in this class (= 01)

6.3.6.2 Instance Attributes Relating to the Acknowledge Handler Object

Attr. ID	Attribute Name	Access Rule	Type	Description
1	Acknowledge Timer	Get / Set	UINT	Waiting time for an 'Acknowledge', before sending again
2	Retry Limit	Get / Set	USINT	Number of 'Acknowledge Time Outs', before the producing application is notified
3	COS Producing connection Instance	Get	UINT	Connection instance containing the path from the producing 'I/O Application Object', which is notified by the Acknowledge Handler Event

6.3.6.3 General Services Relating to the Acknowledge Handler Object

Service code	Service name	Description
$0E_{hex}$	Get_Attribute_Single	Delivers the content of the specified attribute
10_{hex}	Set_Attribute_Single	Changes the content of the specified attribute

6.3.7 Position Sensor Object

Class Code	Class Attributes	Instance Attributes	Number of Instance	Services
23_{hex}	Yes (1, 2)	Yes	1	Yes

6.3.7.1 Klassen Attribute zum Position Sensor Object

Attr. ID	Attribute Name	Access Rule	Type	Description
1	Revision	Get	USINT	Revision of the Position Sensor Object (= 02)
2	Max. Instance	Get	USINT	Highest instance number of an object produced in this class (= 01)

6.3.7.2 Instance Attributes Relating to the Position Sensor Object

Factory default values are shown in '**bold**'. The abbreviations used have already been explained in chapter 4.

⚠	If basic values such as the scaling parameters or the measuring ranges are changed, this is not followed by a plausibility check for the set limits of subordinate values e.g. of the CAMs, the velocity, the acceleration, COS … ! When changing the output formats, the limits for monitoring and setting the resolution must also be redefined. These are not automatically converted and adjusted to the new formats!

Attr. ID	Attribute Name	Access Rule	Volatility	Type	Description
1	Num. of attributes	Get	NV	USINT	Number of attributes supported
2	Attribute list	Get	NV	Array of USINT	List of the attributes supported in the Hiperface Devicenet Adapter
3-9					Not supported
10	Position value	Get	V	DINT	Current position value of the encoder (Pos_Scal)
11	Position sensor type	Get	NV	UINT	Encoder type: 01: absolute singleturn **02: absolute multiturn** 08: absolute linear 65535: no encoder detected
12	Direction counting toggle	Set	NV	BOOL	Counting direction looking at the shaft end of the encoder: **0: CW (clockwise or forward for linear)** 1: CCW (counter-clockwise or backward for linear)
13	Commissioning diagnostic control	Set	NV	BOOL	**Default 1 = ON** Other setting without effect
14	Scaling function control	Set	NV	BOOL	**Default 1 = ON** The 'physical_resolution_span' (attr. 42) is converted to a numerical value (see also 6.3.8.1) – for setting 0 = OFF the scaling function is switched off
15	Position format	Set	NV	ENG UNIT	Format of the 'Position Attribute' **0x1001 = Counts (default rotary)** 0x2203 = Millimetres 0x2204 = µmetres **0x2205 = Nanometres (default linear)** 0x2207 = Inch 0x2208 = Feet A modification leads automatically to an adjusted default value of Att. 16 (if the basic resolution of the encoder completely supports this measurement range).

Attr. ID	Attribute Name	Access Rule	Volatility	Type	Description
16	Measuring units per span - CPR	Set	NV	UDINT	Number of the selected steps per revolution [value range < = **'physical_resolution_span'** (attr. 42)] (see also 6.3.8.1)
17	Total measuring range in measuring units - CMR	Set	NV	UDINT	Number of measuring steps over the entire measurement path [value range < = **PMR 'physical_resolution_span' x 'number of spans'** (attr. 42, 43)] --- (see also 6.3.8.1)
18	Position measuring increment	Set	NV	UDINT	**Default = 1** Defines the smallest value change of the output position value
19	Preset Value	Set	NV	DINT	For the current mechanical position of the encoder, the preset value is set as the new current position value (via an internal 'Offset Value', which is calculated automatically) – if the scaling function (att. 16, 17) is performed after defining the preset value, this may lead to values which no longer correspond (see also 6.3.4.4.2)
20	COS / delta	Set	NV	UDINT	**Default = 5** This position value must change by this amount before an output value is read out, regardless of the set 'Production Inhibit Time' (see 7.2.3.3)
21	Position state register	Get	NV	BYTE	State of the software limit switch: Bit 0: 1 = outside of range Bit 1: 1 = range overflow Bit 2: 1 = dropped below range
22	Position low limit	Set	NV	DINT	**Default = 0x00 00 00 00** Lowest switching point of the limit switch
23	Position high limit	Set	NV	DINT	**Default = 0x7F FF FF FF** Highest switching point of the limit switch [If the value of attr. 10 is greater than the 'position high limit', a range overflow will occur]
24	Velocity value	Get	V	DINT	Current speed of the encoder (updated every 50 ms) Value (sign) / Explanation: 0 – Velocity is constant + – Velocity is increasing - – Velocity is decreasing (slow down) A change in the direction of rotation or counting direction also reverses the sign (change of direction). [The formats are defined in attr. 25 and 26 and refer to the amount (disregarding the sign)]

Attr. ID	Attribute Name	Access Rule	Vola-tility	Type	Description
25	Velocity format	Set	NV	ENG UNIT	Format of the acceleration value: **0x1F04 = steps per second (default for rotative and linear encoder – no over flow alam function of the value rage)** 0x0801 = rev. per second 0x0802 = rev. per minute 0x2B00 = meter per second 0x2B06 = foot per socond 0x2B07 = inch per second
26	Velocity resolution	Set	NV	UDINT	**Default = 1** Defines the smallest value change of the velocity value
27	Minimum velocity set point	Set	NV	DINT	**Default = 0x00 00 00 00** Lower threshold of the velocity value – dropping below it will set the 'min. velocity flag' (warning attr. 47)
28	Maximum velocity set point	Set	NV	DINT	**Default = 0x7F FF FF FF** Upper threshold of the velocity value – exceeding it will set the 'max. velocity flag' (warning attr. 47)
29	Acceleration value	Get	V	DINT	Current acceleration of the encoder (update every 50 ms) [The formats are defined in the attr. 30 and 31]
30	Acceleration format	Set	NV	ENG UNIT	Format of the acceleration value: **0x0810 = Steps per second² (default for rotative and linear encoder)** 0x0811 = rev. per minute per second 0x0812 = rev. per second² 0x1500 = meter per second² 0x1501 = foot per second² 0x1502 = inch per second² Calculation of the difference of the previous speeds
31	Acceleration resolution	Set	NV	UDINT	**Default = 1** Defines the smallest change of the acceleration value
32	Minimum acceleration setpoint	Set	NV	DINT	**Default = 0x00 00 00 00** Lower threshold of the acceleration value – is not evaluated!
33	Maximum acceleration setpoint	Set	NV	DINT	**Default = 0x7F FF FF FF** Upper threshold of the acceleration value
	Further explanations regarding the electronic cam switching mechanism (attributes 34-40) see 6.3.8.3				
34	Number of CAM channels	Get	NV	USINT	N = 4 Number of independent CAM channels

Attr. ID	Attribute Name	Access Rule	Vola-tility	Type	Description
35	CAM channel state register	Get	V	ARRAY of BOOL	State/status of the independent CAM channels
36	CAM channel polarity register	Set	NV	ARRAY of BOOL	Determines the polarity for the CAM channel state register
37	CAM channel enable register	Set	NV	ARRAY of BOOL	Activates the independent CAM channels – array size corresponds with the number of CAM channels
38	CAM low limit	Set	NV	ARRAY of DINT	**Default = 0x00 00 00 00** Switching point for the lower CAM threshold – array size corresponds with the number of CAM channels
39	CAM high limit	Set	NV	ARRAY of DINT	**Default = 0x7F FF FF FF** Switching point for the upper CAM threshold – array size corresponds with the number of CAM channels
40	CAM hysteresis	Set	NV	ARRAY of UINT	**Default = 0** Definition of the hysteresis of the CAM switching points – When calculating the CAM switching points, this value is added to the CAM high limit and subtracted from the CAM low limit
41	Operating status	Get	V	BYTE	Operating status of the encoder: Bit 0: Direction (0 = inc; 1 = dec) Bit 1: Scaling (0 = off; 1 = on) Bit 2…4: Reserved for Devicenet Bit 5…7: Manufacturer-specific
42	Physical resolution span - PRS	Get	NV	UDINT	Max. number of steps per span, which the encoder supports – for rotary encoders, a span corresponds to one revolution (see also 6.3.8.1)
43	Number of spans	Get	NV	UINT	Max. poss. number of spans, which the encoder supports – for rotary encoders, this corresponds to the number of rev. which the encoder supports (see also 6.3.8.1)
44	Alarms	Get	V	WORD	Alarm messages (see 6.3.9.1)
45	Supported Alarms	Get	NV	WORD	Supported alarms: Bit 0 : Position error Bit 2…11 : Reserved for Devicenet Bit 12 : EEPROM checksum error Bit 13 : Encoder startup error
46	Alarm Flag	Get	V	BOOL	Indicates that an alarm from attr. 44 has occurred
47	Warnings	Get	V	WORD	Warning messages (see 6.3.9.2)
48	Supported warnings	Get	NV	WORD	Supported warnings: Bit 0: Speed/velocity outside limit values (in connection with bit 6 and bit 7) Bit 1: Sender current critical (if supported by the encoder)

Attr. ID	Attribute Name	Access Rule	Volatility	Type	Description
					Bit 2: CPU Watchdog Bit 6: Dropped below velocity Bit 7: Velocity exceeded Bit 9: Acceleration exceeded Bit 13: Encoder temperature (if sup ported by the encoder)
49	Warning Flag	Get	V	BOOL	Indicates that a warning from attr. 47 has occurred
50	Operating Time	Get	NV	UDINT	Operating time of the encoder since the last Power ON – unit: tenths of an hour (10/60 h)
51	Offset value	Get	NV	DINT	The device's internal offset, recalcu-lated when the 'Preset Value' is acti-vated

6.3.7.3 Manufacturer-specific (Mandatory) Attributes in Relation to the Position Sensor Object

Attr. ID	Attribute Name	Access Rule	Type	Description
100	Assembly Poll	Set	USINT	Assembly Instanz für den Polling Mode
101	Assembly COS	Set	USINT	Assembly Instanz für den COS / Cyclic Mode
102	Assembly STRB	Set	USINT	Assembly Instanz für den Bit Strobe Mode

6.3.7.4 Manufacturer-specific (Optional) Attributes in Relation to the Position Sensor Object

Factory default values are shown in '**bold**'.

Attr. ID	Attribute Name	Access Rule	Type	Description
108	Max. Velocity	Get	DINT	Max. encoder velocity, which occurred since the last power-on of the Hiperface Devicenet Adapter
109	Max. Acceleration	Get	DINT	Max. acceleration, which occurred since the last power-on of the Hiperface-Devicenet-Adapter
111	Sin/Cos analogue value monitoring	Set	BOOL	Monitoring of the encoder analogue signals 0: Monitoring off **1: Monitoring on**
112	Place of stored 'Preset' value	Set	BOOL	Memory location of the preset value and the associated offset value: **0: Hiperface Devicenet Adapter** 1: Encoder

6.3.7.5 General Services Relating to the Position Sensor Object

Service code	Service name	Description
$0E_{hex}$	Get_Attribute_Single	Delivers the content of the specified attribute
10_{hex}	Set_Attribute_Single	Changes the content of the specified attribute
16_{hex}	Save Parameter	Saves the current volatile parameters in the EEPROM – optional – (see also 6.3.8.2).
18_{hex}	Get_Member	Reads a value from an array (e.g. CAM low limits)
19_{hex}	Set_Member	Writes a value to an array (e.g. CAM low limits)

6.3.8 Notes on Scaling Function, Data Storage and Cam Switching Mechanism

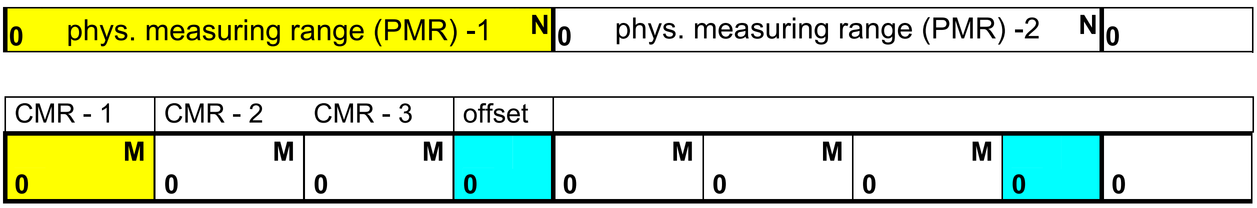
6.3.8.1 Using the Scaling Function

If a rotary absolute encoder exceeds the physical measuring range [PMR], it continues to count either with its minimum value [PminVal] (0) or its maximum value [PmaxVal], depending on the direction of rotation. Successful counting, relating to position readout, is only ensured if the scaled measuring range [CMR] is an integer multiple of [PMR]. The mapping of the [CMR] on the [PMR] with an offset (residual value through non-integer multiple) leads to an incorrectly scaled position value (no monotonous counting sequence).

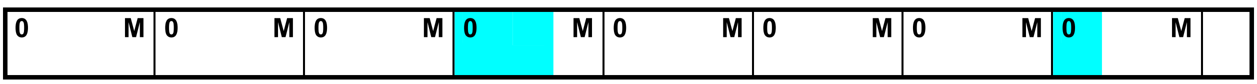
The following illustration shows an example of the relationship between the two measuring ranges.

[0...N] = Minimum/maximum values of the physical measuring range [PMR].

[0...M] = Minimum/maximum values of the scaled measuring range [CMR].



Without considering the resulting offset, the scaled position values are no longer correct. The encoder changes from one physical measuring range to the next.



An offset can be used to calculate a correctly scaled position value. When changing from one physical measuring range to the next, the offset value itself must be adapted.

Possible solutions:

a) Working with offset:
The scaled measuring range is mapped into the physical measuring range, and the resulting offset is adapted. The physical measuring range exceeds the limits or falls within the limits. The offset value must be stored in a non-volatile RAM (EEPROM) to ensure the correct conversion of the physical position value to the scaled position value after turning off and back on again.

☞	This function has **not** been implemented for the Hiperface Devicenet Adapter.

b) Working without offset:
The scaled measuring range is selected such that it can be completely mapped in the physical measuring range without an offset being generated.
This implies the need for a device-side check of the configured CMR value. If required, this must then be automatically adapted/corrected to a value mappable without offset.

Adapted Scaling Functions (Without Offset)

The scaled measuring range is mapped completely in the physical measuring range. The customer-specified value for CMR is matched to obtaining a 2^N value for the relationship between CMR and CPR (no **offset** needed). – The following settings are valid for this:

- R = $\mathbf{2^{**N}}$, mit N = {0, 1, ..., 12}; R = CMR/CPR (with CMR being a multiple of CPR)
- CMR ≤ PMR.

☞	In particular, the upper **limit values for CMR and CPR of the** respective **encoder** connected to the Hiperface Devicenet Adapter shall be taken into account. If, for scaling purposes, a **CMR value above the upper limit** of the connected encoder is set at the Hiperface Devicenet Adapter, the CMR value on the device side is automatically adapted to the next possible CMR value (see examples in the table). If, for scaling purposes, a **CMR value** below the upper limit but **outside a** CMR attributable **without offset** is set on the Hiperface Devicenet Adapter, the CMR value on the device side is automatically adapted to the next-higher permitted CMR value (see examples in the table). Basic requirement: For Devicenet commissioning, the encoder must already be connected to the Hiperface Devicenet Adapter. This must happen before the operating voltage of the Hiperface Devicenet Adapter is switched on.

The tables below show some configurations, to comply with the specified limits:

customer-specified value CPR	CMR	adapted values (device side) CPR	CMR	ScF	R = 2^N CPR
Example of a connected multiturn encoder with an output value at the Hiperface Devicenet Adapter of max. 262,144 steps/rev. x 4,096 rev.					
262,144 262,144	1,073,741,824 1,073,741,824 … 536,870,913	262,144 262,144	1,073,741,824 (2^{30}) 1,073,741,824 (2^{30})	1 1	2^{12} 2^{12}
262,144	536,870,912 … 268,435,457	262,144	536,870,912 (2^{29})	1	2^{11}
…	…	…	…	…	…
262,144	262,144 … 1	262,144	262,144 (2^{18})	1	1
…	…	…	…	…	…
4,096 4,096	1,073,741,824 1,073,741,824 … 16,777,217	4,096 4,096	16,777,216 (2^{24}) 16,777,216 (2^{24})	1/64 1/64	2^{12} 2^{12}
4,096	16,777,216 … 8,388,609	4,096	16,777,216 (2^{24})	1/64	2^{12}
4,096	8,388,608 … 4,194,305	4,096	8,388,608 (2^{23})	1/64	2^{11}
…	…	…	…	…	…
4,096	8,192 … 4,097	4,096	8,192 (2^{13})	1/64	2
4,096	4,096 … 1	4,096	4,096 (2^{12})	1/64	1
…	…	…	…	…	…
1,000 1,000	1,073,741,824 1,073,741,823 … 2,048,001	1,000 1,000	4,096,000 4,096,000	ca. 1/262 ca. 1/262	2^{12} 2^{12}
…	…	…	…	…	…
1,000	1,999 … 1,001	1,000	2,000	ca. 1/262	2
1,000	1,000 … 1	1,000	1,000	ca. 1/262	1
…	…	…	…	…	…

customer-specified value CPR	CMR	adapted values (device side) CPR	CMR	ScF	R = 2^N CPR
Example of a connected single-turn encoder with an output value at the Hiperface Devicenet Adapter of max. 32,768 steps/rev. x 1 rev.					
> = 32,769	beliebig	32,768	32,768	1	1
32,768	1,073,741,824 … 32,769	32,768	32,768	1	1
32,768	32,768	32,768	32,768	1	1
…	…	…	…	…	…

customer-specified value		adapted values (device side)		ScF	R = 2^N
CPR	CMR	CPR	CMR	CPR	
4,096 4,096	32,768 32,767 … 4,097	4,096 4,096	4,096 4,096	1/8 1/8	1 1
4,096	4,096	4,096	4,096	1/8	1
…	…	…	…	…	…
1,000 1,000	32,768 32,767 … 1,001	1,000 1,000	1,000 1,000	1/32,768 1/32,768	1 1
1,000	1,000 … 1	1,000	1,000	1/32,768	1
…	…	…	…	…	…

To operate a **rotary** encoder in 'Continuous Mode' (so-called rotary axis), CMR must be a multiple (R = 2^N) based on parameter 'CPR'.

Factor R is limited to **4096** (max. number of revolutions for multi-turn encoders).

Factor R is limited to **1** (max. number of revolutions for single-turn encoders).

customer-specified value		adapted values (device side)		Auflösung
CPR	CMR	CPR	CMR	(resultierender)
Example: a connected linear encoder with an output value at the Hiperface Devicenet Adapter with a max. measuring step of 1 µm and a max. measuring length of 40 m.				
< 1,000	any	1,000	max. Encoder-messbereich 40,000,000	1 µm
1,000	any	1,000	max. Encoder-messbereich 40,000,000	1 µm
…	…	…	…	…
3,333	any	3,333	12,001,200	3.333 µm
…	…	…	…	…
500,000	any	500,000	80,000	0.5 mm
…	…	…	…	…
1,000,000,000	any	1,000,000,000	40	1.0 m
…	…	…	…	…
> = 1,073,741,824	any	1,073,741,824	37	1073.74.. mm bzw. 1.07374.. m

For **linear encoders**, the CMR value is automatically adapted to the max. possible measuring range resulting from the resolution (measuring step). CPR gives the required resolution in nm, with greater values denoting a lower resolution.

For the linear encoder XKS09 the highest resolution recommended is 0.05 mm. For this the "position format" (Att. 15) has to be set to µm and the CPR (Att. 16) has to be set to 50,000 (see 6.3.7.2).

For thelinearen Encoder Lincoder L230 the highest possible resolution via the adapter is 19.531 µm.

Futher explanations concerning the parameter 'CPR' (Attribut 16) for linear encoders:

Scaling 'CPR'	Used format	Explanation
{ 00,00,12,3D} hex , (4,669)	counts (steps) == measuring {~ 4.7 µm}	Basic setting for linear encoder 'XKS09' with highest resolution.
{ 00,00,4C,4B} hex , (19,531)	counts (steps) == Maßeinheit {~ 19.5 µm}	Basic setting for linear encoder LinCoder L230 with highest resolution.
(10,000)	{ 10 µm == 0.01 mm }	Reduction to format {10 µm}
(100,000)	{ 100 µm == 0,1 mm }	….format { 0.1 mm}
(10,000,000)	{ 10 mm == 0.01 m }	….format { 0.01 m}
(2,540,000)	1/10 Inch (in)	… format { 0.1 in }
(25,400,000)	Inch (in)	… format { in }
(30,480,000)	1/10 Foot (ft)	….format { 0.1 ft }
(304,800,000)	Foot (ft)	….format { ft }
(1,000,000,000)	Meter (m)	….format { 1.0 m }
{ 40,00,00,00} hex , (1,073,741,824)	(1.073 m)	Setting with lowest resolution.

All indications of the 'CPR' column as a multiplw of nanometer.

Output position values of the Hiperface Devicenet Adapter

The value range for the output position values thus is 0 … (CMR-1), CMR being from the "Adapted values (device-side)" column.

6.3.8.2 Saving Parameter/Attribute Settings in the EEPROM

All non-volatile attributes, which can be changed during commissioning, are generally – automatically – saved in the EEPROM and are thus retained, even after a Power OFF/ON. The memory location can be selected to be either the Hiperface Devicenet Adapter or the connected encoder, depending on what is advantageous for the intended application.

6.3.8.3 Using the Cam Switching Mechanisms / the CAMs

The Hiperface Devicenet Adapter provides an electronic cam switching mechanism with 4 channels of 8 cams each. The cams can be adjusted in relation to the upper and lower switching points as well as the hysteresis to the switching points. The hysteresis serves to prevent bouncing, as may happen when the position value fluctuates around the switching point. For this, a value is subtracted from the min. switching point and a value added to the max. switching point.

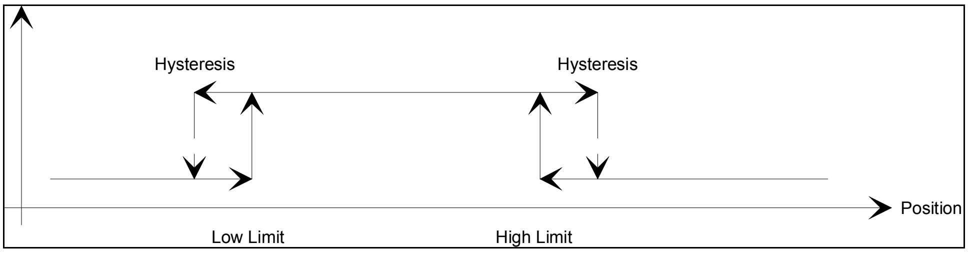


Figure 6–3: CAM Hysteresis

The table below shows how the current position value is compared with the adjusted switching points, in relation to the previous position value.

Previous pos. value	Min. switching point (LOW)	Max. switching point (HIGH)
Within switching point / (1)	Low limit – hysteresis	High limit + hysteresis
Outside switching point / (0)	Low limit	High limit

☞	The values of the switching point limits (LOW/HIGH) as well as the hysteresis are not checked for plausibility.

Cam Channel Status (attributes 35 and 36)

The status of the individual cams results from the individual bits of the array. If a cam is active, the associated bit is at '1'. If a cam is inactive, the associated bit is at '0'. If the value for a cam is set to '1' in the 'CAM Channel Polarity Register', the current status of the corresponding cam is inverted.

State	Polarity Bit = 0 (default)	Polarity Bit = 1 (inverted)
Position within limits	'CAM active' / [1]	'CAM active' / [0]
Position outside limits	'CAM inactive' / [0]	'CAM inactive' / [1]

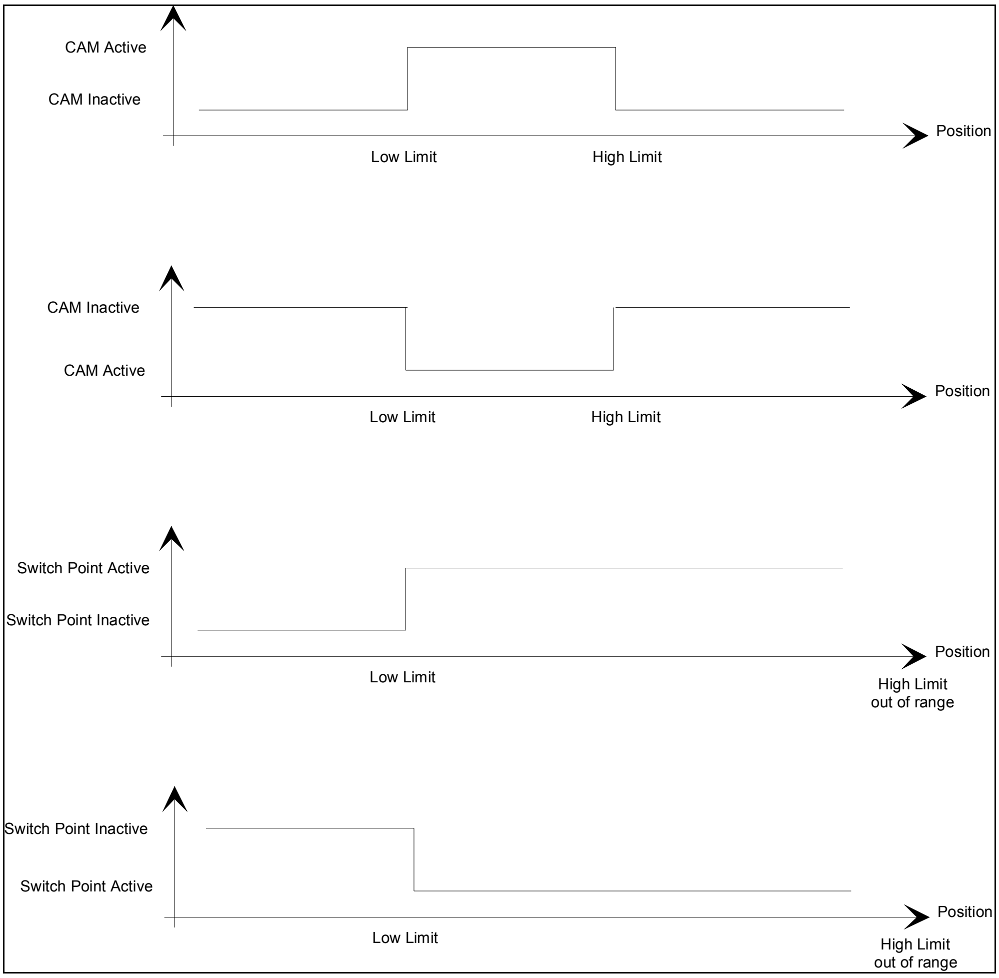


Figure 6–4: CAM Switching Points in Connection with the Polarity

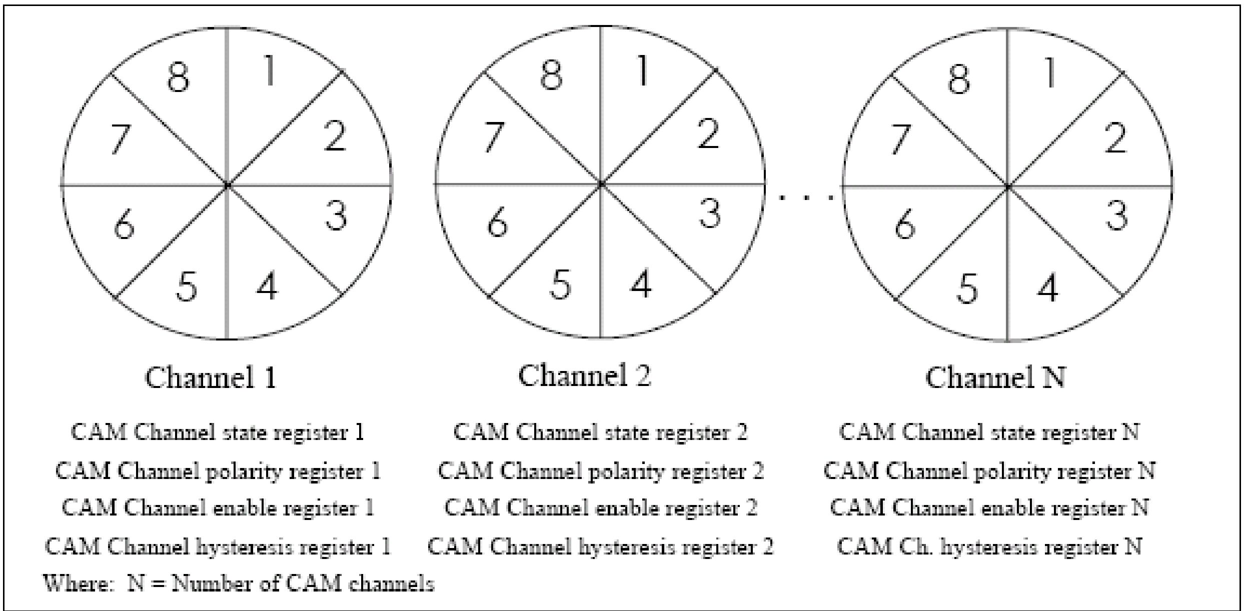
Cam Channel Enable register

If the corresponding bit for a cam is set to '1' (active), the status will be determined. If the corresponding bit for a cam is set to '0' (inactive), the cam status will not be determined. The rest status of the individual cams is '0'.

6.3.8.3.1 Explicit access to individual cams

For a simple GET/SET Single Attribute access (attributes 35 and 36), all data values are read/set. For access to individual data values, the following services are thus required:

Principle of the CAM Channels



In the Position Sensor Object, for the mapping of the cams, altogether 4 channels of 8 cams each have been implemented

Pa-rame-ter	Attribute of the Position Sensor Object	Description	Data Type	Array Size
1	38_{dec}	CAM Low Limit	DINT	4 * 8 * 4
2	39_{dec}	CAM High Limit	DINT	4 * 8 * 4
3	40_{dec}	CAM Hysteresis Limit	UINT	4 * 8 * 2
4	35_{dec}	CAM Channel State Register	BOOL	4
5	36_{dec}	CAM State Polarity	BOOL	4
6	37_{dec}	CAM Channel Enable Register	BOOL	4

{Array size for DINT and UINT = number of channels * number of cams/channel * data size in bytes --- example: CAM Low Limit = 4 Channels * 8 CAMs * 4 Bytes each}

Access is via attribute ID + member ID

Member ID = [(Channel No.) – 1] * 8 + (CAM No.)

Example 1: Channel 1, CAM 1 = Member ID 1

Example 2: Channel 4, CAM 8 = Member ID 32

The Explicit Messages are fragmented or unfragmented > 8 bytes, depending on the message length. The messages of the following services are fragmented:

Dienst (Service)	
Get_Attribute_Single [0E]$_{hex}$	Real data length response > 6 Bytes
Get_Attribute_Member [18]$_{hex}$	Real data length response > 6 Bytes
Set_Attribute_Single [10]$_{hex}$	Real data length request > 3 Bytes
Set_Attribute_Member [19]$_{hex}$	Real data length request > 1 Bytes

Mapping a data value, consisting of more than one byte, is performed according to Devicenet protocol in the "Little Endian" format (i.e. the LSB is transmitted first).

The MAC IDs are used for the following sample protocols:

MAC ID for the slave 63$_{dec}$ = 3F$_{hex}$

MAC ID for the master 01$_{dec}$ = 01$_{hex}$

To manually perform the services during commissioning, the RSNetWorx Class Instance Editor (see chapter 9) is recommended; it allows convenient entry of the parameters and automatically performs the fragmentation. The parameters are always entered in the hex. format.

Sample protocol for the 'Set Attribute Member' service, to define CAM Low Limits

This command request message set the Channel (2), CAM (4) member value (4 bytes) to [01.02.03.04]$_{hex}$., with a fragmented message.

Identifier (bit 10...0)			Data (8 bytes)				**--------->(Frag-1)**			
Gr	MAC ID	MSG ID	Head	Frag	Service	Class	Instance	Data 0	Data 1	Data 2
10	111111	100	81	00	19	23	01	26	0C	00

MSG ID	[04]: Master's Explicit Request Message
Head	Frag-Bit = 1, XID-Bit = 0, Source MAC-ID (Master)
Frag	[ttnn-nnnn] with tt = fragment type [First], nn-nnnn = fragment number [0]
Service	Request "Set_Member_Attribute"
Class	Position Sensor Object
Instance	Instance ID of Class
Data 0	Attribute ID. -- (38dec. = 0x26)
Data 1--2	Member ID. -- (12). -- UINT {0x00.0C }.

Identifier (bit 10...0)			Data (8 bytes)				**(Ack-1) <---------**			
Gr	MAC ID	MSG ID	Head	Frag	Rsp_0	Rsp_1	Rsp_2	Rsp_3	Rsp_4	Rsp_5
10	111111	011	81	C0	00					

MSG ID	[03]: Slave's Explicit/ Unconnected Response Message
Head	Frag-Bit = 1, XID-Bit = 0, Destination MAC-ID (Master)
Frag	[ttnn-nnnn] with tt = fragment type [Response], nn-nnnn = fragment number [0]
Rsp_0	Status (Success)

Identifier (bit 10...0)			Data (8 bytes)				**--------->(Frag-2)**			
Gr	MAC ID	MSG ID	Head	Frag	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5
10	111111	100	81	81	04	03	02	01		

Head	Frag-Bit = 1, XID-Bit = 0, Source MAC-ID (Master)
Frag	[ttnn-nnnn] with tt = fragment type [Last], nn-nnnn = fragment number [1]
Data 0 | 1 | | 2 | 3	Value for the selected attribute / member: {Lsb, Lsb+1, Lsb+2, Lsb+3 }

Identifier (bit 10...0)			Data (8 bytes)				**(Ack-2) <---------**			
Gr	MAC ID	MSG ID	Head	Frag	Rsp_0	Rsp_1	Rsp_2	Rsp_3	Rsp_4	Rsp_5
10	111111	011	81	C1	00					

Head	Frag-Bit = 1, XID-Bit = 0, Destination MAC-ID (Master)
Frag	[ttnn-nnnn] with tt = fragment type [Response], nn-nnnn = fragment number [1]
Rsp_0	Status (Success)

Additionally confirmation of the complete Sequence (as non-fragmented sequence)

Identifier (bit 10...0)			Data (8 bytes)				**(Final Ack) <---------**			
Gr	MAC ID	MSG ID	Head	Service	Data_0	Data_1	Data_2	Data_3	Data_4	Data_6
10	111111	011	01	99						

<u>Sample protocol for the 'Get Attribute Member' service, to read out CAM Low Limits</u>

This command request message gets the Channel (2), CAM (4) member value (4 bytes), with an unfragmented message.

Identifier (bit 10...0)			Data (8 bytes)				**--------->(Request)**			
Gr	MAC ID	MSG ID	Head	Service	Class	Inst	Data 0	Data 1	Data 2	Data 3
10	111111	100	01	18	23	01	26	0C	00	

MSG ID	[04]: Master's Explicit Request Message
Head	Frag-Bit = 0, XID-Bit = 0, Source MAC-ID (Master)
Service	Request "Get_Member_Attribute"
Class	Position Sensor Object
Instance	Instance ID of Class
Data 0	Attribute ID. -- (38 dec = 0x26)
Data 1--2	Member ID. -- { (k-1) * 8 + n }. -- (12). -- UINT {0x00.0C }.

Identifier (bit 10...0)			Data (8 bytes)				**(Final Ack) <---------**			
Gr	MAC ID	MSG ID	Head	Service	Data_0	Data_1	Data_2	Data_3	Data_4	Data_6
01	111111	011	01	98	04	03	02	01		

MSG ID	[03]: Slave's Explicit/ Unconnected Response Message
Head	Frag = 0, XID = 0, Destination MAC-ID (Master)
Service	Response to requested message
Data 0--3	Current value of this attribute {0x01.02.03.04 }.

6.3.9 Warnings and Alarms

6.3.9.1 Alarms

Bit	Description	False (0)	True (1)
0	Position error detected	No	Yes
1	Not supported	-	-
2-11	Reserved for Devicenet	-	-
12	EEPROM checksum error	No	Yes
13	Encoder startup error (e.g. no encoder detected)	No	Yes
14-15	Not supported	-	-

In the normal state, all alarm bits are '0'.

6.3.9.2 Warnings

Bit	Description	False (0)	True (1)
0	Frequency exceeded	No	Yes
1	Sender current critical (if supported by the encoder)	No	Yes
2	CPU Watchdog	OK	Reset trig-gered
3	Not supported	-	-
4	Not supported	-	-
5	Not supported	-	-
6	Dropped below velocity	No	Yes
7	Velocity exceeded	No	Yes
8	Not supported	-	-
9	Acceleration exceeded	No	Yes
10-12	Not supported	-	-
13	Encoder temperature (if supported by the encoder)	OK	Exceeded
14-15	Not supported	-	-

In the normal state, all warning bits are '0'.

7 Types of Communication (Master ↔ Hiperface Devicenet Adapter)

7.1 Devicenet Communication – General

Devicenet is based on a connection-oriented communication model. Direct access to data from the network is not possible. Data is always exchanged via the connections set up and allocated for this.

	The Hiperface Devicenet Adapter is an 'Input Device'. This means that the Hiperface Devicenet Adapter produces useful data but cannot consume any useful data from the Master.

7.1.1 Using the CAN Identifier

The 11-bit CAN Identifier is used to define the Connection ID for a message connection to be set up, and is divided into the Message Groups 1 to 4 of different sizes.

CAN Identifier Bits											**Identity Use**	**Bereich ($_{hex}$)**
10	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**		
0	Message ID				Source MAC ID						Message Group 1	000 - 3FF
1	0	MAC ID						Message ID			Message Group 2	400 - 5FF
1	1	Message ID			Source MAC ID						Message Group 3	600 - 7BF
1	1	1	1	1	Message ID						Message Group 4	7C0 - 7EF
1	1	1	1	1	1	1	x	x	x	x	Invalid CAN Identifiers	7F0 - 7FF

Message ID	Identifies a message within a message group of a node
Source MAC ID	Source node address of the message
Destination MAC ID	Target node address of the message

7.2 Predefined Master/Slave Connection Set

This is a predefined and thus simple method of establishing a connection.

For this, the 'Group 2 Only Unconnected Explicit Message Port' provides an interface with which a set of connections, which is already preconfigured in the device, can be allocated.

The basis of this model is a 1:n communication structure consisting of a central control unit (Master) and decentralised I/O devices (Slaves).

The predefined connection objects occupy the instances 1 … max. 5 in the 'Connection Object' (Class ID 5).

7.2.1 The CAN Identifier for the Predefined Master/Slave Connection Set

The distribution of the Connection IDs:

CAN Identifier Bits											Identity Use	Range (hex)
10	9	8	7	6	5	4	3	2	1	0		
0	**Message ID**				**Source MAC ID**						**Group 1 Messages**	**000-3FF**
0	1	1	0	1	Source MAC ID						Slave's I/O Change of State or Cyclic Message	
0	1	1	1	0	Source MAC ID						Slave's I/O Bit Strobe Response Message	
0	1	1	1	1	Source MAC ID						Slave's I/O Poll Response or COS/Cyclic Ack Message	
1	**0**	**MAC ID**						**Mess. ID**			**Group 2 Messages**	**400-5FF**
1	0	Source MAC ID						0	0	0	Master's I/O Bit Strobe Command Message	
1	0	Multicast MAC ID						0	0	1	Reserved for Master's Use – Use is TBD	
1	0	Destination MAC ID						0	1	0	Master's COS/Cyclic Ack Message	
1	0	Source MAC ID						0	1	1	Slave's Explicit/Unconnected Response Message	
1	0	Destination MAC ID						1	0	0	Master's Explicit Request Message	
1	0	Destination MAC ID						1	0	1	Master's I/O Poll Command or COS/Cyclic Message	
1	0	Destination MAC ID						1	1	0	Group 2 Only Unconnected Explicit Request Messages	
1	0	Destination MAC ID						1	1	1	Duplicate MAC ID Check Message	

Mapping the data with more than one byte is performed according to Devicenet protocol in the "Little Endian" format, i.e. the LSB is transmitted first.

Note that the IDs specified for the attributes (in the previous chapters) were given in decimal values and now had to be converted, in the following data string examples, to hexadecimal values!

1. Example of Using the CAN Identifier:

Identifier (bit 10...0)			Request to slave				→			
Gr	MAC ID	MSG ID	Head	Service	Class	Instance	Data 0	Data 1	Data 2	Data 3
10	111111	000	01							

MAC ID [**3F**] hex: Destination (Slave) -- Exception with MSG ID = 0
MSG ID { 0, 2, 4, 5, 6 7 }
Head contains the Source MAC-ID (Master): -- [**01**]

Identifier (bit 10...0)			Response Group-2				←			
Gr	MAC ID	MSG ID	Head	Service	Data_0	Data_1	Data_2	Data_3	Data_4	Data_6
10	111111	011								

MAC ID [**3F**] hex: Source (Slave)
MSG ID { 3, 7 }
Head contains the Destination MAC-ID (Master): -- [**01**]

2. Example of Establishing a Connection to a 'Group 2 Only' Server

Each connection is implemented via a 'Group 2 Only Unconnected Explicit Messaging' (Group 2, Message ID 6). In the predefined 'Master Slave Connection Set', an instance is created using the **Allocate_Master/Slave_Connection Set** service (4B $_{hex}$) of the 'Devicenet Object'.

This service receives the information on the selected connection mode through the **'Allocation Choice Byte' (Byte Data 0)**:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	Ack Sup-press	Cycle	Change of State	Reserved	Bit Strobed	Polled	Explicit Message

Allocate_Master / Slave_Connection Set

In the following example, the message connections 'Explicit' and 'Polled' are set up. The Explicit message connection is automatically set to the status 'Established' by using the 'Expected_Packet_Rate' [10 s]. After this time, the connection is released if there is no further access to the connection setup. The I/O message connection then is in the configuration status.

Identifier (bit 10...0)			Data (8 bytes)				→			
Gr	MAC ID	MSG ID	Head	Service	Class	Instance	Data 0	Data 1	Data 2	Data 3
10	111111	110	01	4B	03	01	03	01		

MAC ID	[3F]: Destination (Slave)
MSG ID	[06]: Group 2 Only Unconnected Explicit Request Messages
Head	Frag = 0, XID = 0, Source MAC-ID (Master)
Service	Request "Allocate_Master / Slave_Connection Set"
Class	Device Net Object
Instance	Instance ID of Class
Data 0	Allocation Choice Byte
Data 1	Allocater's MAC-ID

Identifier (bit 10...0)			Data (8 bytes)				←			
Gr	MAC ID	MSG ID	Head	Service	Data_0	Data_1	Data_2	Data_3	Data_4	Data_6
10	111111	011	01	CB	00					

MAC ID	[3F]: Source (Slave)
MSG ID	[03]: Slave's Explicit/ Unconnected Response Message
Head	Frag = 0, XID = 0, Destination MAC-ID (Master)
Service	Response to requested message
Data_0	Message Body Format (8 / 8)

Placing the connection from the config. status into the set-up status ‘Established’

To place an allocated I/O connection onto the 'Established State' the time monitoring of the connection is started by setting the 'Expected_Packet_Rate' ('Set_Attribute_Single' service to attribute 9 in the corresponding Connection Object). Below, this is exemplified for a Poll connection.

Identifier (bit 10...0)			Data (8 bytes)				→			
Gr	MAC ID	MSG ID	Head	Service	Class	Instance	Data 0	Data 1	Data 2	Data 3
10	111111	100	01	10	05	02	09	E8	03	

MSG ID	[04]: Master's Explicit Request Message
Head	Frag = 0, XID = 0, Source MAC-ID (Master)
Service	Request "Set_Attribute_Single"
Class	Connection Object
Instance	Instance ID of Class
Data 0	Attribute ID = Expected_Packet_Rate
Data 1 | 2	Time in ms (Low bytes) | (High bytes)

Identifier (bit 10...0)			Data (8 bytes)				←			
Gr	MAC ID	MSG ID	Head	Service	Data_0	Data_1	Data_2	Data_3	Data_4	Data_6
10	111111	011	01	90	E8	03				

MSG ID	[03]: Slave's Explicit/ Unconnected Response Message
Head	Frag = 0, XID = 0, Destination MAC-ID (Master)
Service	Response to requested message
Data 0 | 1	Actual time in ms (Low bytes) | (High bytes) -- (possible adjusted)

3. Example of Cancelling a Connection to a 'Group 2 Only' Server

To cancel one or several connection(s), the **'Release_Master/ Slave_Connection_Set'** service is used. Via the **'Release Choice Byte' (byte data 0)** it is determined which connections will be cancelled. In the following example, the two connections – which were set up in the above example – are being cancelled.

Identifier (bit 10...0)			Data (8 bytes)				→			
Gr	MAC ID	MSG ID	Head	Service	Class	Instance	Data 0	Data 1	Data 2	Data 3
10	111111	110	01	4C	03	01	03			

MSG ID	[06]: Group 2 Only Unconnected Explicit Request Messages
Head	Frag = 0, XID = 0, Source MAC-ID (Master)
Service	Release "Allocate_Master / Slave_Connection Set"
Class	Device Net Object
Instance	Instance ID of Class
Data 0	Release Choice Byte

Identifier (bit 10...0)			Data (8 bytes)				←			
Gr	MAC ID	MSG ID	Head	Service	Data_0	Data_1	Data_2	Data_3	Data_4	Data_6
10	111111	011	01	CC						

MSG ID	[03]: Slave's Explicit/ Unconnected Response Message
Head	Frag = 0, XID = 0, Destination MAC-ID (Master)
Service	Response to requested message

7.2.2 Explicit Message Connections

Explicit messages are used for the general data exchange: configuration data with low priority, general management data or diagnostic data can be transferred via this system.

This type of communication is always a point-to-point (peer-to-peer) communication in a Client/Server system, where the request from a client is always acknowledged by a response from the server.

The Hiperface Devicenet Adapter only supports an explicit message connection:

Gruppe/ Group	Description	Class ID	Instance ID
2	Explicit messaging connection (Connection Object)	5	1

Distinction is made between unfragmented and fragmented telegram sequences.

1. <u>**Example of Unfragmented 'Explicit Messaging'**</u>

Unfragmented means that the length of a message is smaller than or equal to 8 bytes and thus can be transmitted in a 'frame'.

<u>Set Attribute [Attribute 12 $_{dec}$ – Counting direction]</u>

An unfragmented 'Request Message' contains a 5-byte header [Head, Service, Class, Instance and Attribute]. The remaining usable data length for a 'Set' service is thus limited to 3 bytes.

Identifier (bit 10...0)			Data (8 bytes)				→			
Gr	MAC ID	MSG ID	Head	Service	Class	Instance	Data 0	Data 1	Data 2	Data 3
10	111111	100	01	10	23	01	0C	01		

MSG ID	[04]: Master's Explicit Request Message
Head	Frag = 0, XID = 0, Source MAC-ID = 1
Service	Request "Set_Attribute_Single"
Class	Position Sensor Object
Instance	Instance ID of Class
Data 0	Attribute ID
Data 1	Value for the this attribute {0, 1}

Identifier (bit 10...0)			Data (8 bytes)				←			
Gr	MAC ID	MSG ID	Head	Service	Data_0	Data_1	Data_2	Data_3	Data_4	Data_6
10	111111	011	01	90						

MSG ID	[03]: Slave's Explicit/ Unconnected Response Message
Head	Frag = 0, XID = 0, Destination MAC-ID (Master)
Service	Response to requested message

<u>Get Attribute [Attribute 12 $_{dec}$ – Counting direction]</u>

A 'Response Message' to a request contains 2 bytes of header information. This results in a usable data length for a 'Get' service of 6 bytes.

Identifier (bit 10...0)			Data (8 bytes)				→			
Gr	MAC ID	MSG ID	Head	Service	Class	Instance	Data 0	Data 1	Data 2	Data 3
10	111111	100	01	0E	23	01	0C			

MSG ID [04]: Master's Explicit Request Message
Head Frag = 0, XID = 0, Source MAC-ID (Master)
Service Request "Get_Attribute_Single"
Class Position Sensor Object
Instance Instance ID of Class
Data 0 Attribute ID = operating status

Identifier (bit 10...0)			Data (8 bytes)				←			
Gr	MAC ID	MSG ID	Head	Service	Data_0	Data_1	Data_2	Data_3	Data_4	Data_6
01	111111	011	01	8E	01					

MSG ID [03]: Slave's Explicit/ Unconnected Response Message
Head Frag = 0, XID = 0, Destination MAC-ID (Master)
Service Response to requested message
Data 0 Current value of this attribute

Saving the parameters in the EEPROM

The **'Save'** service of the 'Position Sensor Object' saves all parameters in the EEPROM. The 'Reset' and 'Restore' services are used in the same way, to restore the factory default values. An attribute ID is not required for this.

Identifier (bit 10...0)			Data (8 bytes)				→			
Gr	MAC ID	MSG ID	Head	Service	Class	Instance	Data 0	Data 1	Data 2	Data 3
10	111111	100	01	16	23	01				

MSG ID [04]: Master's Explicit Request Message
Head Frag = 0, XID = 0, Source MAC-ID (Master)
Service Request "Save"
Class Position Sensor Object

Identifier (bit 10...0)			Data (8 bytes)				←			
Gr	MAC ID	MSG ID	Head	Service	Data_0	Data_1	Data_2	Data_3	Data_4	Data_6
10	111111	011	01	96						

MSG ID [03]: Slave's Explicit/ Unconnected Response Message
Head Fray = 0, XID = 0, Destination MAC-ID (Master)
Service Response to requested message

2. Example of Fragmented 'Explicit Messaging'

Fragmented means that the length of the message to be transferred is greater than 8 bytes (incl. header data) and thus has to be sent in several frames. A fragmented 'Request' message contains, in the first frame, 6 bytes of header data [Head, Fragment No., Service, Class, Instance and Attribute]. All further frames only have 2 bytes of header data [Head, Fragment No.].

Hence, fragmentation is used in the following cases:

Service	
Get Attribute	Real data length > 6 Bytes
Set Attribute	Real data length > 3 Bytes

Set Attribute [Attribute 19 $_{dec}$ - Preset Value]

Dieses 'Request' Kommando setzt den 'Presetwert' (Preset Value - 4 bytes) auf [01.02.03.04]$_{hex}$.

Identifier (bit 10...0)			Data (8 bytes)				(Fragment 1) →			
Gr	MAC ID	MSG ID	Head	Frag	Service	Class	Instance	Data 0	Data 1	Data 2
10	111111	100	81	00	10	23	01	13	04	03

MSG ID	[04]: Master's Explicit Request Message
Head	Frag-Bit = 1, XID-Bit = 0, Source MAC-ID (Master)
Frag	[ttnn-nnnn] with tt = fragment type [First], nn-nnnn = fragment number [0]
Service	Request "Set_Attribute_Single"
Class	Position Sensor Object
Instance	Instance ID of Class
Data 0	Attribute ID
Data 1 | 2 |	Value for the selected attribute {Bit 07...00} | {Bit 15...08}

Identifier (bit 10...0)			Data (8 bytes)				(Ack 1) ←			
Gr	MAC ID	MSG ID	Head	Frag	Rsp_0	Rsp_1	Rsp_2	Rsp_3	Rsp_4	Rsp_5
10	111111	011	81	C0	00					

MSG ID	[03]: Slave's Explicit/ Unconnected Response Message
Head	Frag-Bit = 1, XID-Bit = 0, Destination MAC-ID (Master)
Frag	[ttnn-nnnn] with tt = fragment type [Response], nn-nnnn = fragment number [0]
Rsp_0	Status (Success)

Identifier (bit 10...0)			Data (8 bytes)				(Fragment 2) →			
Gr	MAC ID	MSG ID	Head	Frag	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5
10	111111	100	81	81	02	01				

Head	Frag-Bit = 1, XID-Bit = 0, Source MAC-ID (Master)
Frag	[ttnn-nnnn] with tt = fragment type [Last], nn-nnnn = fragment number [1]
Data 0 | 1 |	Value for the selected attribute {Bit 23...16} | {Bit 31...24}

Identifier (bit 10...0)			Data (8 bytes)				(Ack 2) ←			
Gr	MAC ID	MSG ID	Head	Frag	Rsp_0	Rsp_1	Rsp_2	Rsp_3	Rsp_4	Rsp_5
10	111111	011	81	C1	00					

Head	Frag-Bit = 1, XID-Bit = 0, Destination MAC-ID (Master)
Frag	[ttnn-nnnn] with tt = fragment type [Response], nn-nnnn = fragment number [1]
Rsp_0	Status (Success)

At the end, the sequence must be completed by a further, unfragmented message.

Identifier (bit 10...0)			Data (8 bytes)				(Final Ack) ←			
Gr	MAC ID	MSG ID	Head	Service	Data_0	Data_1	Data_2	Data_3	Data_4	Data_6
10	111111	011	01	90						

Get Attribute [Attribute 39 $_{dec}$ – CAM High Limits]

The example shows how a data array, which contains all entries of the upper cam switch limit values, can be read out. The slave (Hiperface Devicenet Adapter) responds to this request, based on the data volume, with a fragmented message.

Identifier (bit 10...0)			Data (8 bytes)				(Request) →			
Gr	MAC ID	MSG ID	Head	Service	Class	Instance	Data 0	Data 1	Data 2	Data 3
10	111111	100	01	0E	23	01	27			

MSG ID	[04]: Master's Explicit Request Message
Head	Frag = 0, XID = 0, Source MAC-ID (Master)
Service	Request "Get_Attribute_Single"
Class	Position Sensor Object
Instance	Instance ID of Class
Data 0	Attribute ID

Identifier (bit 10...0)			Data (8 bytes)				(Fragment 1) ←			
Gr	MAC ID	MSG ID	Head	Frag	Servi-ce	Rsp_0	Rsp_1	Rsp_2	Rsp_3	Rsp_4
10	111111	011	81	00	8E	FF	FF	FF	7F	FF

MSG ID	[03]: Slave's Explicit/ Unconnected Response Message
Head	Frag-Bit = 1, XID-Bit = 0, Destination MAC-ID (Master)
Frag	[ttnn-nnnn] with tt = fragment type [First], nn-nnnn = fragment number [0]
Service	Response to requested message
Rsp_0 3	Response Data (Byte 1...4) – {Channel 1 CAM 1: 0x7F.FF.FF.FF}
Rsp_4	Response Data (Byte 5) – {Channel 1 CAM 2: 0x__.__.__FF}

Identifier (bit 10...0)			Data (8 bytes)				(Ack 1) →			
Gr	MAC ID	MSG ID	Head	Frag	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5
10	111111	100	81	C0	00					

Head	Frag = 1, XID = 0, Source MAC-ID (Master)
Frag	[ttnn-nnnn] with tt = fragment type [Ack], nn-nnnn = fragment number [0]
Data 0	Status (Success)

Identifier (bit 10...0)			Data (8 bytes)				(Fragment 2) ←			
Gr	MAC ID	MSG ID	Head	Frag	Rsp_0	Rsp_1	Rsp_2	Rsp_3	Rsp_4	Rsp_5
10	111111	011	81	41	**FF**	**FF**	**7F**	**FF**	**FF**	**FF**

Head	Frag-Bit = 1, XID-Bit = 0, Destination MAC-ID (Master)
Frag	[ttnn-nnnn] with tt = fragment type [Middle], nn-nnnn = fragment number [1]
Rsp_0 3	Response Data (Byte 1..4) – {Channel 1 CAM 2: 0x7F.FF.FF.__}
Rsp_4	Response Data (Byte 5..7) – {Channel 1 CAM 3: 0x__.FF.FF.FF}

Identifier (bit 10...0)			Data (8 bytes)				(Ack 2) →			
Gr	MAC ID	MSG ID	Head	Frag	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5
10	111111	100	81	C0	00					

Head	Frag = 1, XID = 0, Source MAC-ID (Master)
Frag	[ttnn-nnnn] with tt = fragment type [Ack], nn-nnnn = fragment number [0]
Data 0	Status (Success)

…….. there follow the further message sequences for channel 1 to CAM 8, for Channel 2 from CAM 1 to CAM 8, …, to channel 4 of CAM 1 to CAM 8 --- {altogether 4 channels x 8 CAMs x 4 bytes = 128 bytes}.

7.2.3 I/O Messages

Using I/O messages, high-priority data can be exchanged, which goes from a producer to one or several consumer(s). Thus, I/O messages are used for fast or time-critical data transmissions. All 8 bytes of the CAN message can thus be used for data transmission. The Hiperface Devicenet Adapter only supports unfragmented I/O messages.

The Hiperface Devicenet Adapter supports the following three types of I/O connections:

- Poll I/O Connection
- Bit-Strobe I/O Connection
- Change of State (COS) / Cyclic IO Connection

The following table shows how, for the respective type of I/O connection, the individual data components of the Hiperface Devicenet Adapter can be implemented:

	I/O Connection					
	Poll		Bit-Strobe		COS (1)	
Attribute from the Position Sensor Object for setting up the connection type (see 6.3.7.3)	Att.-ID 100 Name: Assembly Poll		Att.-ID 101 Name: Assembly STRB		Att.-ID 102 Name: Assembly COS	
Input Assembly (from the Assembly Object – see 6.3.4.4 pp.) to select the data components via the associated instance **(Input Data)**	Instance	Data component	Instance	Data component	Instance	Data component
	1	Position (default)	1	Position (default)	1	Position (default)
	2	Position + Flag (Alarm, Warning)	2	Position + Flag (Alarm, Warning)	2	Position + Flag (Alarm, Warning)
	3	Position + Velocity	3	Position + Velocity	3	Position + Velocity (2)
	100	Position + CAM state	100	Position + CAM state	100	Position + CAM state

(1) The time needed to trigger a new message for COS if the value of a data component has changed, is 50 ms for the velocity and << 1 ms for all other data components.

(2) In the mains connestion category Change of state, with the data component speed value no new information is being released when the parameter Resolution (Att. 26) is set to the max. permissible value.

7.2.3.1 Poll I/O Connection

The Polling Mode is the standard mode of a Master/Slave communication. Normally, a Master cyclically polls all slaves. During a poll, a Master can transmit data to the Hiperface Devicenet Adapter (Poll Command Message) and, with the response (Poll Response Message), take on data from the Hiperface Devicenet Adapter.

Gruppe/ Group	Description	Class ID	Instance ID
2	Poll I/O Connection (Connection Object)	5	2

<u>1. Example of a Poll 'Request' / 'Response' Message</u>

For I/O messages, it is necessary to allocate the I/O connection via the 'Allocation Choice Byte' and to place it into the Established state (see 7.2.1 – 2nd example). The configured Assembly Instance (Input Data) determines the data components which are transmitted as a response sequence (Response) to the 'Poll Request'.

Identifier (bit 10...0)			Data (8 bytes)				→			
Gr	MAC ID	MSG ID	Data 0	Data 1	Data2	Data 3	Data 4	Data 5	Data 6	Data 7
10	111111	101								

MAC ID [3F]: Destination (Slave)
MSG ID [05]: Master's I/O Poll Command / COS Message
Data None

Identifier (bit 10...0)			Data (8 bytes)				←			
G	MSG ID	MAC ID	Data 0	Data 1	Data2	Data 3	Data 4	Data 5	Data 6	Data 7
0	1111	111111	01	1F	01	00				

MAC ID [3F]: Source (Slave)
MSG ID [0F]: Slave's Poll Response or COS Message (Group 1)
Data 0 | 1 | 2 | 3 | Position value Bit 07 -- 00 | 15 -- 08 | 23 -- 16 | 31 -- 24 |

7.2.3.2 Bit-Strobe I/O Connection

In the Bit Strobe Mode, the Master sends a Bit Strobe Command to each slave whose MAC ID is contained in the Master’s Scanlist. The Bit Strobe Command is not evaluated for content but only used as a trigger for transmitting the response.

Gruppe/ Group	Description	Class ID	Instance ID
2	Bit Strobe I/O Connection (Connection Object)	5	3

The configured Assembly Instance 'Input Data' defines the data component(s) which is/are sent as a response to a Bit Strobe request. Caution: Here, the MAC ID is the Master address!

Identifier (bit 10...0)			Data (8 bytes)				→			
Gr	MAC ID	MSG ID	Data 0	Data 1	Data2	Data 3	Data 4	Data 5	Data 6	Data 7
10	000001	000								

MAC ID [01]: Source (**Master**)
MSG ID [00]: Master's I/O Bit-Strobe Command Message
Data 0...7 optional (No data, or all 8 Byte used)

Identifier (bit 10...0)			Data (8 bytes)				←			
G	MSG ID	MAC ID	Data 0	Data 1	Data2	Data 3	Data 4	Data 5	Data 6	Data 7
0	1110	111111	01	1F	01	00				

MAC ID [3F]: Source (Slave)
MSG ID [0E]: Slave's Bit Strobed Response Message (Group 1)
Data 0 | 1 | 2 | 3 | Position value Bit 07 -- 00 | 15 -- 08 | 23 -- 16 | 31 -- 24 |

7.2.3.3 Change of State (COS) / Cyclic I/O Connection

A COS / Cyclic connection differs from the two other I/O connection types in that the producer (i.e. the Hiperface Devicenet Adapter) generates messages by itself and thus acts as a Client. This can happen in an event- or time-controlled manner.

For COS, the message is generated in an event-controlled manner, i.e. only if a data value (relating to the selected 'Assembly Instance') has changed.

In the second case, a message is generated in a time-controlled manner (Cyclic). Thus, after expiry of the Cycle Time, a transmission is started regardless of whether or not a value has changed. In addition, a delay (Production Inhibit Time) can be defined in the millisecond range (0 to 65535 ms), to reduce the bus load.

The COS / Cyclic connection can be established as acknowledged or unacknowledged.

Gruppe/ Group	Description	Class ID	Instance ID
2	COS / Cyclic (Connection Object)	5	4

8 Assembly / Installation

8.1 General Notes

Hiperface Devicenet Adapters are state-of-the-art devices. The adapter should be fitted by an expert with knowledge of electrical and precision engineering. The adapter must only be used for the purpose for which it was designed.

The Hiperface Devicenet Adapter has two connections via M12 plug connectors and can thus be inserted directly into a trunk line or, via a drop line, of the bus network.

8.2 Safety Notes

⚠	The Safety Notes from Chapter 2 must be observed!

8.3 Montage

3 holes are provided for mounting the unit to your base plate, using 3 screws M4x10 mm (provided by the customer).

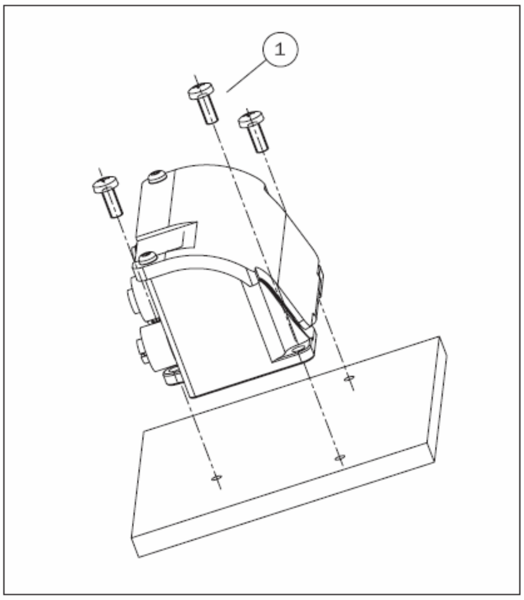


Figure 8–1: Assembly Drawing

⚠	The interface adapter shall be assembled such that it is not exposed to prolonged direct sunlight.

8.3.1 Maintenance Notes

Hiperface-Devicenet-Adapters are maintenance-free. We recommend, at regular intervals

- to check the mechanical connection
- to check screw connections and plug connections.

8.4 Encoder Detection and Devicenet Resolution

The following encoders are detected by the Hiperface Devicenet Adapter:

Designation/Encoder Series	Max. Number of Steps	Max. Steps/Rev.
SRS…	1	262.144 (18 Bit)
SCK…	1	262.144 (18 Bit)
SKS…	1	32.786 (15 Bit)
SEK…	1	4.096 (12 Bit)
SRM…	4.096	262.144 (18 Bit)
SCL…	4.096	262.144 (18 Bit)
SKM…	4.096	32.786 (15 Bit)
Designation/Encoder Series		**Resolution**
LinCoder L230 …		auf Anfrage
XKS …		0,05 mm (t1)

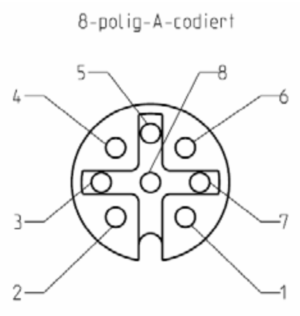
Figure 8–2: Encoder Detection

(t1)	The basic resolution provided is higher. However, adjusting the resolution to be more accurate than 0.05 mm is not useful. The wire drive mechanism is not designed for this.

☞	All standard encoders from the respective encoder series are detected. For customer-specific variations, the connectibility must always be checked.

8.5 PIN and Wire Allocation

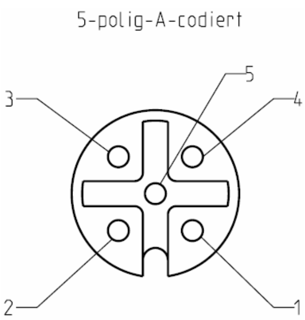
Hiperface input (X1) --- M12 box



PIN	Wire colours	Signal	Explanation
1	brown	REFSIN	Process data channel
2	white	+ SIN	Process data channel
3	black	REFCOS	Process data channel
4	pink	+ COS	Process data channel
5	yellow	Daten +	RS485 parameter channel
6	purple	Daten -	RS485 parameter channel
7	blue	GND	Ground connection
8	red	+ UB	Encoder supply voltage via the adapter
	Drain/screen		Housing potential

Figure 8–3: PIN and Wire Allocation – Hiperface

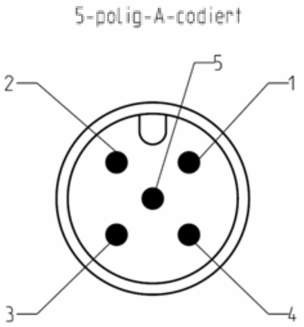
Devicenet OUT (X2) --- M12 socket



PIN	Wire colours	Signal	Explanation
1		Drain/ Schirm	Bus drain/screen has no connection to the housing
2		V +	Supply voltage via the bus
3		V -	Ground connection (GND)
4		CAN_H	H-line Devicenet
5		CAN_L	L-line Devicenet

Figure 8–4: PIN and Wire Allocation – Devicenet OUT

Devicenet IN (X4) --- M12 plug



PIN	Wire colours	Signal	Explanation
1		Drain/ Schirm	Bus drain/screen has no connection to the housing
2		V +	Supply voltage via the bus
3		V -	Ground connection (GND)
4		CAN_H	H-line Devicenet
5		CAN_L	L-line Devicenet

Figure 8–5: PIN and Wire Allocation – Devicenet IN

8.5.1 Installation Notes re: Voltage Supply

Generally, the operating voltage is supplied via the bus line. The following limitation applies:

☞	Max. current flow across the plugs of the Hiperface Devicenet Adapter is limited to 2 A.

8.5.2 Screening

For Devicenet, the housing of the Hiperface Devicenet Adapter **must not** be connected with the screen of the bus line! The screen is placed on a pin of the plug connectors.
Via the metal parts of the machine/system, the housing lies on earth potential. If the housing is not connected with earthed metal parts of the system, separate earthing is recommended in order to prevent external field interspersions into the device.

8.6 Device Handling in the Network

The following features of the Hiperface Devicenet Adapter are configured via the hardware:

- Station address (Node ID) --- also possible via bus protocol.
- Bus termination --- external.
- Preset function.

The following measures are required to execute one of these functions:

- Using a Torx screwdriver (size Tx10), undo the screws of the housing lid and (fold) open the housing lid.
- Following successful adjustment, the housing lid must be closed again, and the Torx screws must be tightened to 0.7 to 0.8 Nm, to ensure the IP protection specified in the data sheet.

☞	During device handling, ensure that the sealing label in the area of the viewing window for the LEDs is not being damaged.

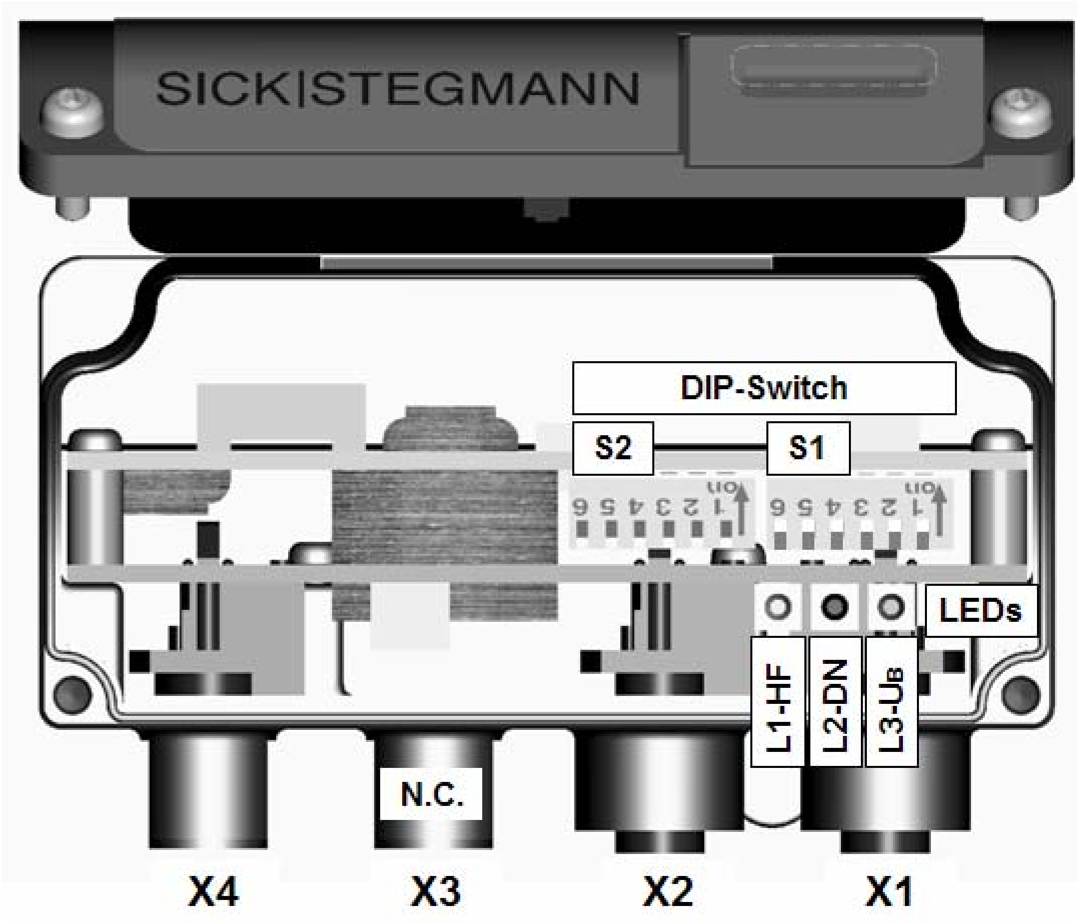


Figure 8–6: Device Handling in the Network 'Device View'

DIP-Switch 2 (S2)						DIP-Switch 1 (S1)					
6	5	4	3	2	1	6	5	4	3	2	1
OFF	OFF	OFF	OFF	OFF	OFF	ON	ON	ON	ON	ON	ON
-	-	-	-	-	Selection	2^5	2^4	2^3	2^2	2^1	2^0
Set	Baud rate				Address						

Representation of the as-delivered state.

Figure 8–7: Handling in the Network 'DIP Switch Adjustment'

8.6.1 Address Setting

The station (node) address (MAC ID) can be set using DIP switches or via the bus protocol. The factory setting (default) is address "63". The address must not be the same as for another subscriber in the network.

Setting DIP switch 2 DIP-1 (selection of address source)	**Address setting DIP switch 1 DIP-1…6**	**Saved address in the EEPROM**	**Devicenet uses this node address (MAC ID)**
OFF	0 bis 63	any	Value of DIP-Switch
ON	any	0 bis 63	Value from EEPROM

S1-DIP-6	**S1-DIP-5**	**S1-DIP-4**	**S1-DIP-3**	**S1-DIP-2**	**S1-DIP-1**	**Adresse**
2^5 (msb)	**2^4**	**2^3**	**2^2**	**2^1**	**2^0 (lsb)**	
0	0	0	0	0	0	0
0	0	0	0	0	1	1
...	...	...	...	...	...	...
1	1	1	1	1	0	62
1	1	1	1	1	1	63

To read in a changed value of the DIP switches, the supply voltage must be turned off and then back on!

The factory default setting is address '63' and address source DIP switch.

8.6.1.1 Address Table

Address table							
decimal	binary	decimal	binary	decimal	binary	decimal	binary
0	0000000	32	0100000	64	1000000	96	1100000
1	0000001	33	0100001	65	1000001	97	1100001
2	0000010	34	0100010	66	1000010	98	1100010
3	0000011	35	0100011	67	1000011	99	1100011
4	0000100	36	0100100	68	1000100	100	1100100
5	0000101	37	0100101	69	1000101	101	1100101
6	0000110	38	0100110	70	1000110	102	1100110
7	0000111	39	0100111	71	1000111	103	1100111
8	0001000	40	0101000	72	1001000	104	1101000
9	0001001	41	0101001	73	1001001	105	1101001
10	0001010	42	0101010	74	1001010	106	1101010
11	0001011	43	0101011	75	1001011	107	1101011
12	0001100	44	0101100	76	1001100	108	1101100
13	0001101	45	0101101	77	1001101	109	1101101
14	0001110	46	0101110	78	1001110	110	1101110
15	0001111	47	0101111	79	1001111	111	1101111
16	0010000	48	0110000	80	1010000	112	1110000
17	0010001	49	0110001	81	1010001	113	1110001
18	0010010	50	0110010	82	1010010	114	1110010
19	0010011	51	0110011	83	1010011	115	1110011
20	0010100	52	0110100	84	1010100	116	1110100
21	0010101	53	0110101	85	1010101	117	1110101
22	0010110	54	0110110	86	1010110	118	1110110
23	0010111	55	0110111	87	1010111	119	1110111
24	0011000	56	0111000	88	1011000	120	1111000
25	0011001	57	0111001	89	1011001	121	1111001
26	0011010	58	0111010	90	1011010	122	1111010
27	0011011	59	0111011	91	1011011	123	1111011
28	0011100	60	0111100	92	1011100	124	1111100
29	0011101	61	0111101	93	1011101	125	1111101
30	0011110	62	0111110	94	1011110	126	1111110
31	0011111	63	0111111	95	1011111	127	1111111

8.6.2 Baud Rate Setting

The data transmission rate (baud rate) is set via DIP switch 2 (S2) – DIP 2 to 5. The baud rate setting must be the same for all subscribers in the Devicenet network.

DIP-Switch 2 (S2)					
DIP-5	**DIP-4**	**DIP-3**	**DIP-2**	**Baud rate source**	**Data rate**
OFF	OFF	OFF	OFF	DIP-Switch	125 Kbaud (default)
OFF	OFF	OFF	ON	DIP-Switch	250 Kbaud
OFF	OFF	ON	OFF	DIP-Switch	500 Kbaud
OFF	OFF	ON	ON	DIP-Switch	125 Kbaud
…				DIP-Switch	125 Kbaud
ON	ON	OFF	ON	DIP-Switch	125 Kbaud
ON	ON	ON	OFF	EEPROM	acc. to EEPROM-value
ON	ON	ON	ON	Devicenet network	Auto Baud (1)

(1) To successfully operate this function, there must be at least two further subscribers in the network who actively communicate with each other.

To read in a changed value of the DIP switches, the supply voltage must be turned off and back on!

The factory default setting is 125 kBaud.

8.6.3 Counting Direction

OFF	CW	ascending	Factory settings (default)
ON	CCW	descending	

The counting direction is determined by commissioning parameter setting using a configuration tool such as RS-NetWorx via the bus protocol.

8.6.4 Preset Function

The Hiperface Devicenet Adapter is set to a special, predefined value if the PRESET function is executed [by pushing DIP switch S2 (DIP-6) for at least 1 s to position ON]. Afterwards, the DIP switch must be pushed back to the OFF position. --- The factory default value is OFF.

The preset function is not intended for dynamic adjustment processes. The function is meant for electronic adjustment during commissioning, to allocate a certain position value to any mechanical shaft position of the encoder. In case of constantly recurring activation of the preset function, the corresponding memory locations in the EEPROM are destroyed, with time lapsing!

The use of the preset function leads to a change in the position value output by the Hiperface Devicenet Adapter. This could cause an unexpected movement which may lead to damage of the system, other objects or bodily harm.

8.6.5 Bus Termination

The bus termination must be performed with an external bus terminator (120 ohms – M12 thread).

If the Devicenet subscribers are wired in 'line topology', the internal termination resistor must only be switched on **at the respective last subscriber at both ends of a line** (i.e. the two – physically – most distant devices).

8.7 Status/Display Information

The devices have three LEDs displaying the status information.

LED		State	Explanation
L1-HF	yellow	OFF	HIPERFACE OK (nominal state)
		FLASHING	HIPERFACE initialisation
		ON	HIPERFACE error (only if encoder observation is activated - default setting)

LED		No operating voltage, not online	Online, connection not OK resp. not established	Online, connection established (nominal state) [1]	Connection Time Out resp. slight error	Critical connection error
L2-DN	red/ green	OFF	FLASHING GREEN	GREEN	FLASHING RED	RED
$L3\text{-}U_S$	green	OFF	GREEN	GREEN	GREEN	GREEN

[1]	Valid data communication only means that master and slave can communicate with one another via telegrams. It does not mean that the data within the remote telegrams is also correct --- e.g. incorrect position value, incorrect offset, ...

8.7.1 Status/Display Information During Power-UP

Before the status of the L2-DN network LEDs goes to a stable state, a test is performed during Power-UP, as shown below:

- L2-DN OFF
- L2-DN GREEN for 0.25 s approx.
- L2-DN RED for 0.25 s approx.
- L2-DN OFF
- L2-DN correct status display

9 Commissioning Notes for the Use of RSNetWorx

In practice, RSNetWorx is frequently used. To support the commissioning staff during handling, some screenshots are added below, together with explanations. Its should be noted that the different versions available on the market cannot be considered. The screenshots are examples and do not claim to fully show all possibilities.

9.1 Reading in EDS File

It is recommended to first read in the electronic data sheet (EDS) of the Hiperface Devicenet Adapter and thus supplement the device library. The electronic data sheet describes the basic device parameters in factory settings, which can then be adapted for the corresponding application. This enables easy commissioning.

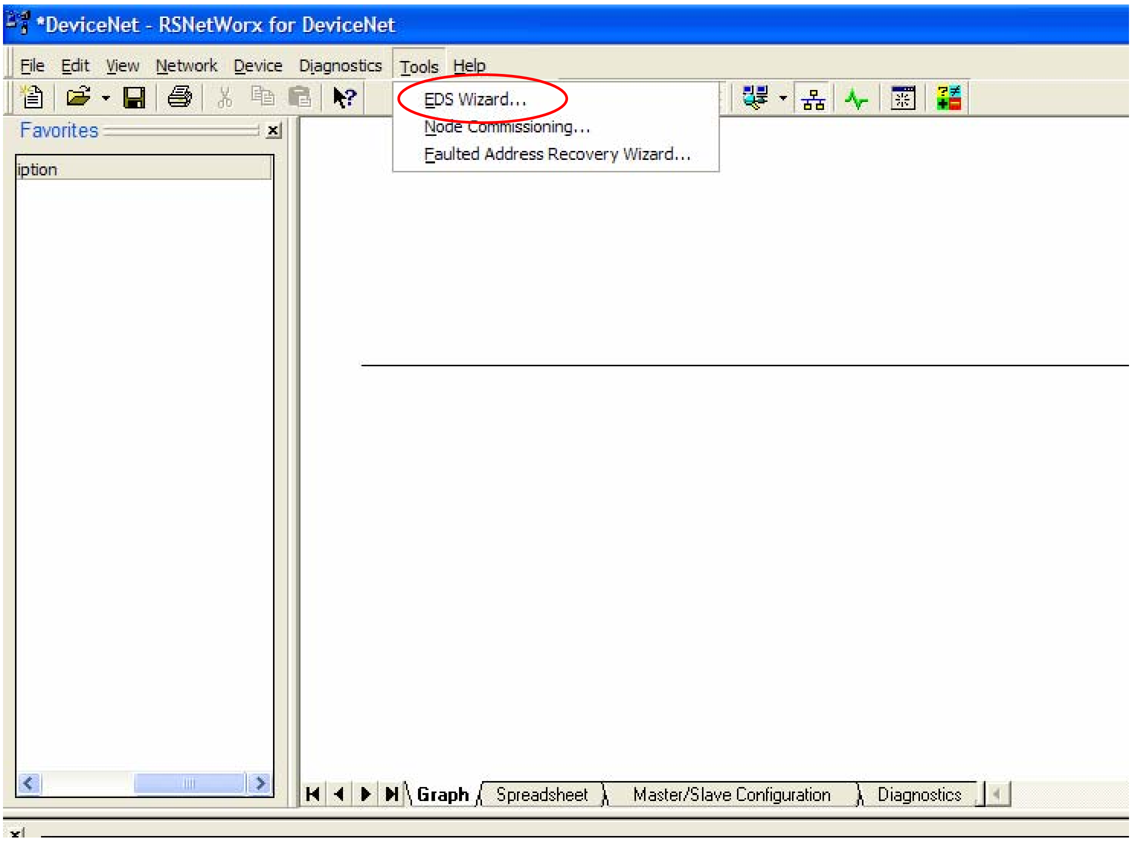


Figure 9–1: Integrating EDS File

During integration of the EDS file, ensure the additional integration of the matching manufacturer-specific device icon (change icon and, via browse, find and select the source file).

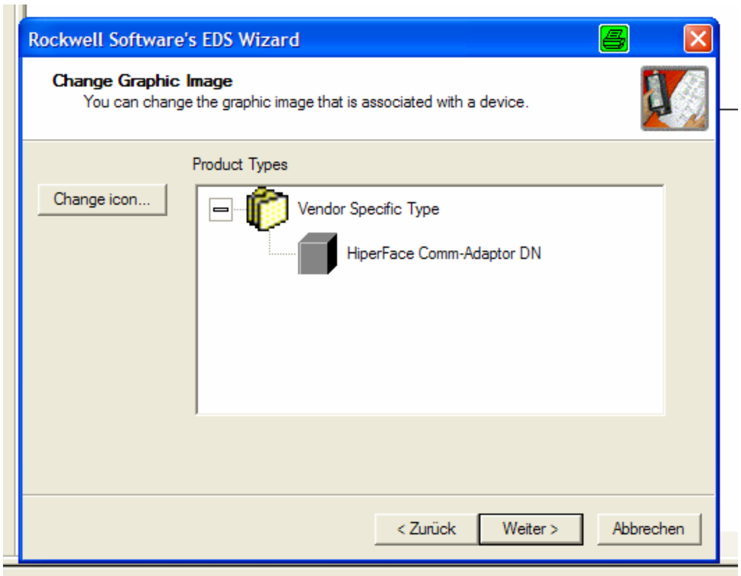


Figure 9–2: Integrating Device Icon

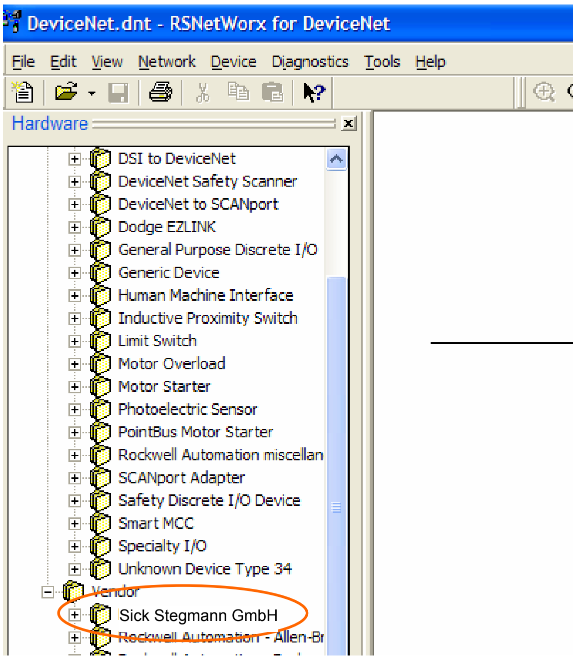


Figure 9–3: RSNetWorx Device Library

9.2 Scanning the Network for Subscribers

If the device is connected, the baud rate, the MAC ID (device address) are set and if the EDS file and icon are integrated into the library, the network can be scanned for subscribers.

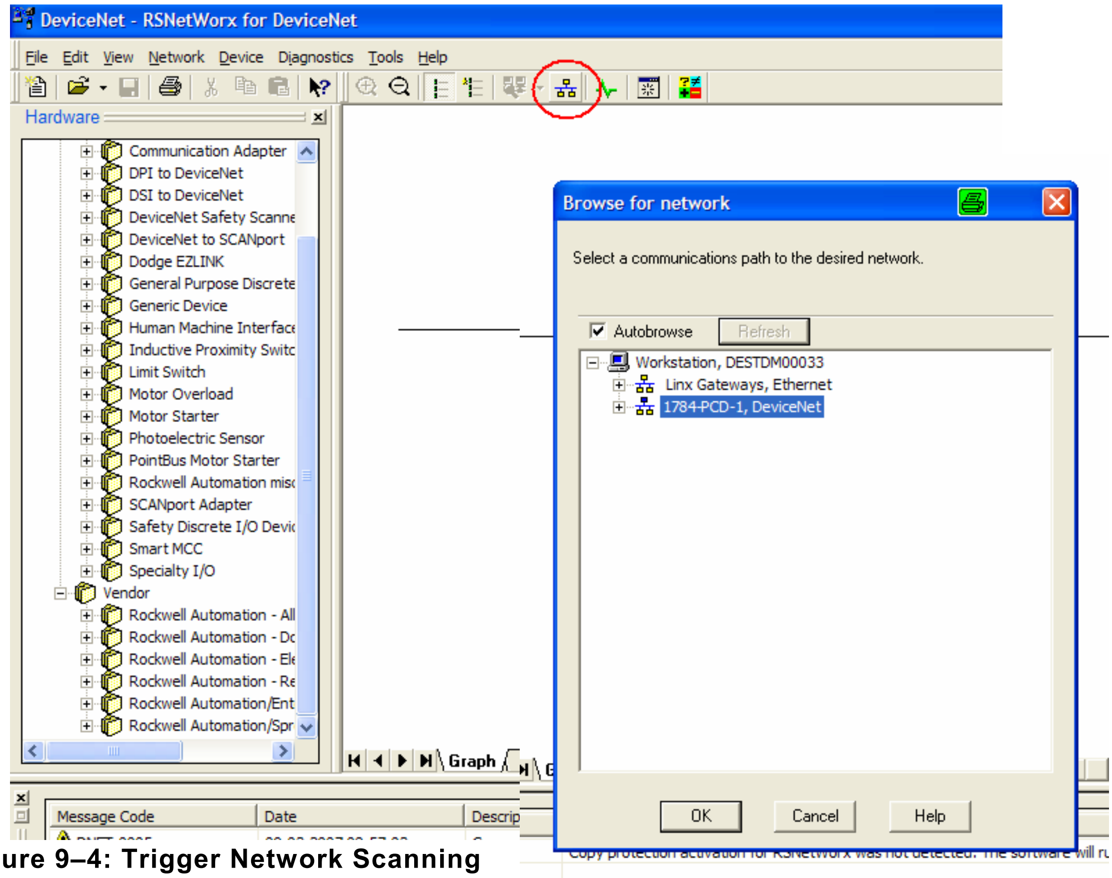


Figure 9–4: Trigger Network Scanning

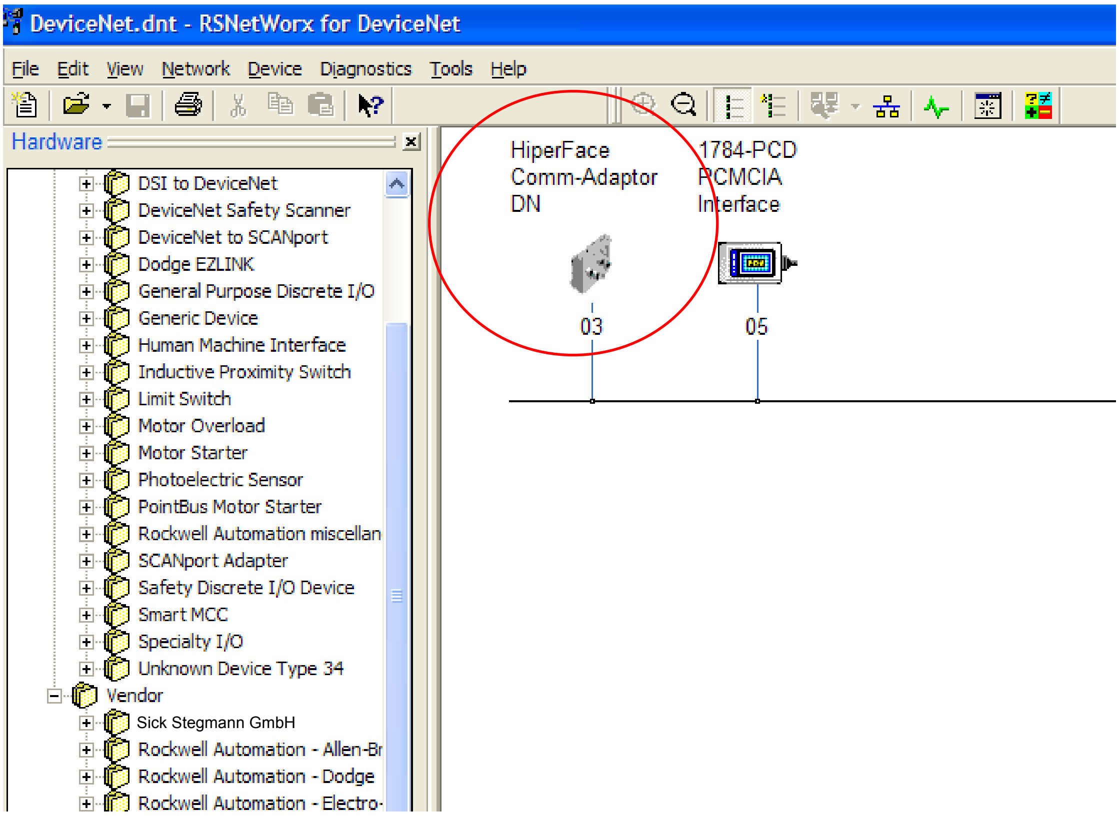


Figure 9–5: Network Scanning – View of Devices Scanned

9.3 Setting Device Properties

This is where the instance attributes and the manufacturer-specific attributes of the Position Sensor Object as well as the individual attributes of further objects can be adapted to the application. – Selection via double-click on the icon of the Hiperface Devicenet Adapter.

9.3.1 Properties – General

Display of the general device properties (name changeable and address matching the setting)

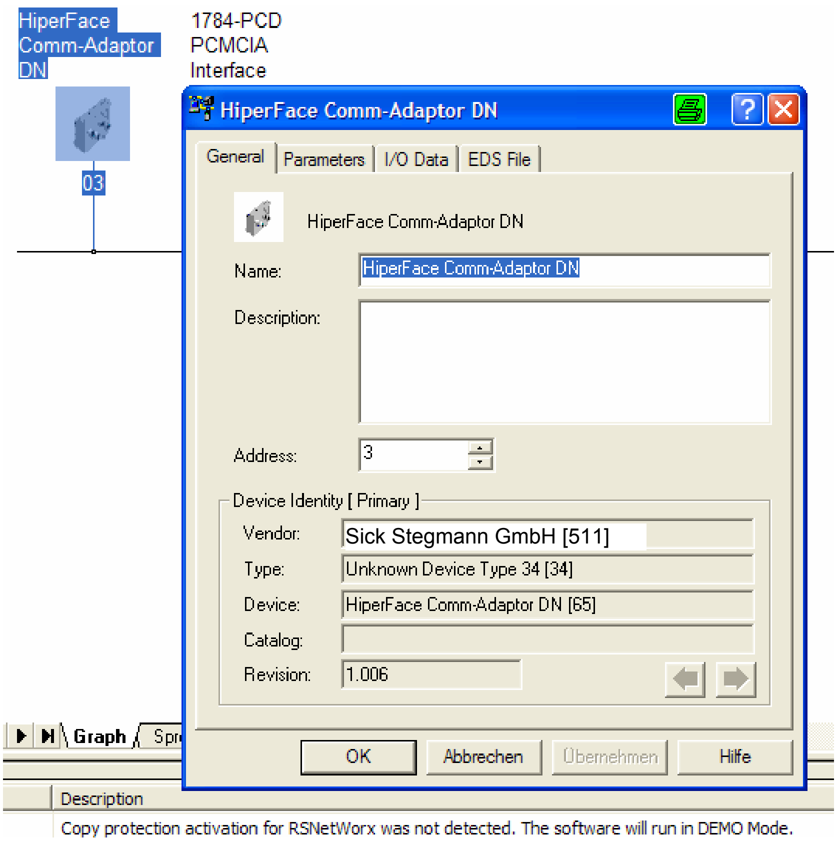


Figure 9–6: Device Properties – General

9.3.2 Properties – Parameters

If the Hiperface Devicenet Adapter is switched to online, the set parameters can be sent via a download to the Hiperface Devicenet Adapter or, for checking, collected via an upload from the Hiperface Devicenet Adapter.

☞	Note that the line number of the individual parameters does not match the Attribute ID.

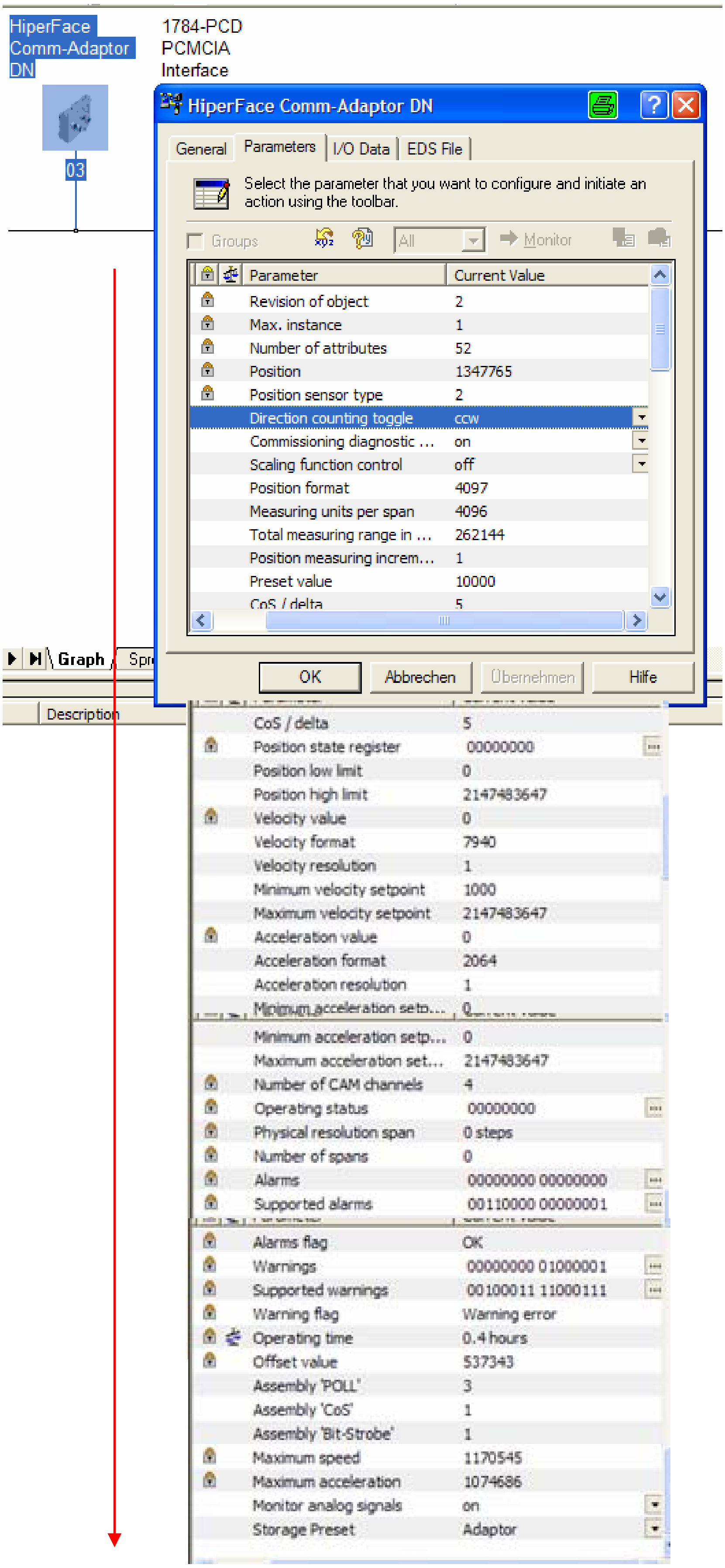


Figure 9–7: Device Properties – Parameters

9.3.3 Properties – I/O Data

Shows the default I/O data of the Hiperface Devicenet Adapter (editing the I/O data see 9.6).

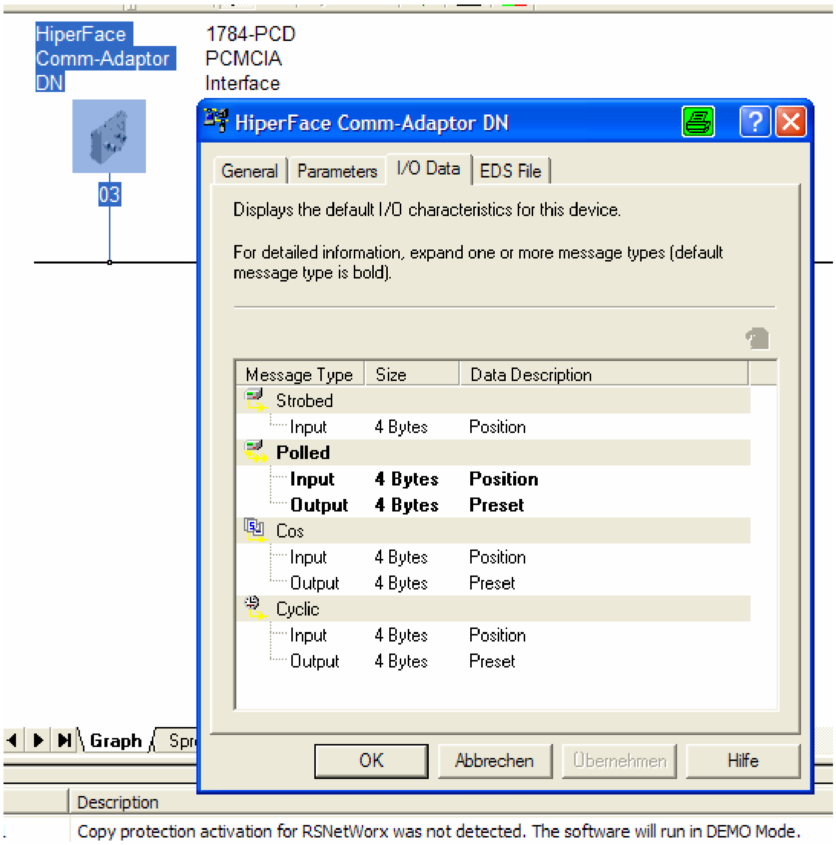


Figure 9–8: Device Properties – I/O Data

9.3.4 Properties – EDS File

Anzeige allgemeiner Informationen zum EDS File zur Information.

9.4 Node Parametrisation – Node Commissioning

Address setting (must match with the hardware setting or the EEPROM setting) and setting of the baud rate (this must be set identically for all devices in the bus network).

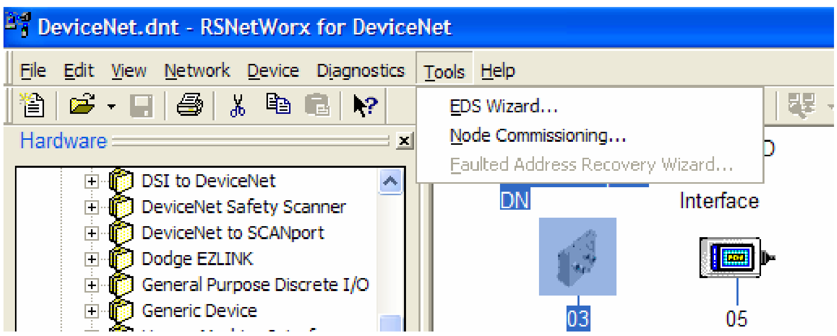


Figure 9–9: Device Properties – Node Commissioning 1

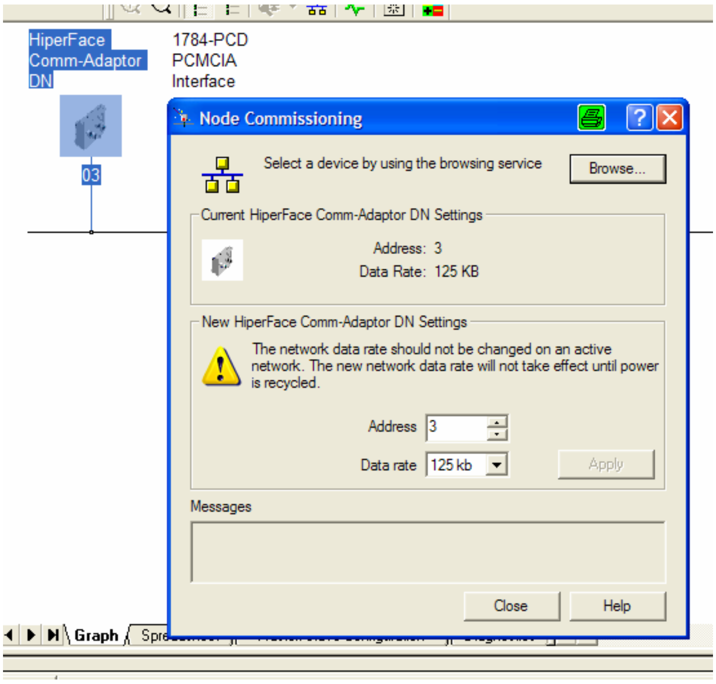


Figure 9–10: Device Properties – Node Commissioning 2

9.5 Class Instance Editor

Using the Class Instance Editor provides an easy method of, for example, sending GET/SET telegrams to network subscribers, during commissioning. General services for all classes are being supported. The user need not take account of the details of fragmentation for long telegrams. This is being done automatically by the Class Instance Editor.

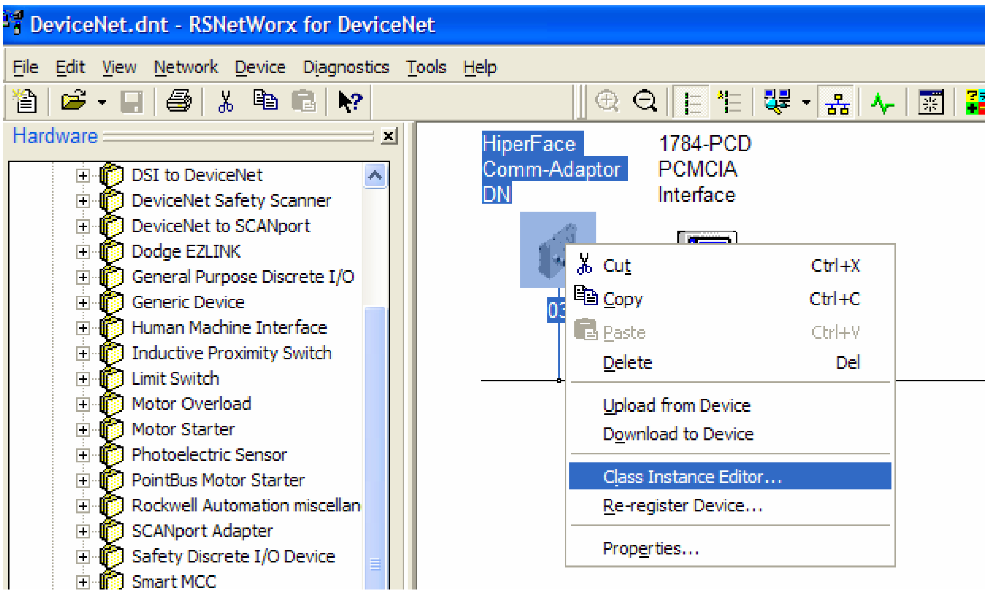


Figure 9–11: Class Instance Editor 1

Note that all entries in the Class Instance Editor must be made in hexadecimal values!

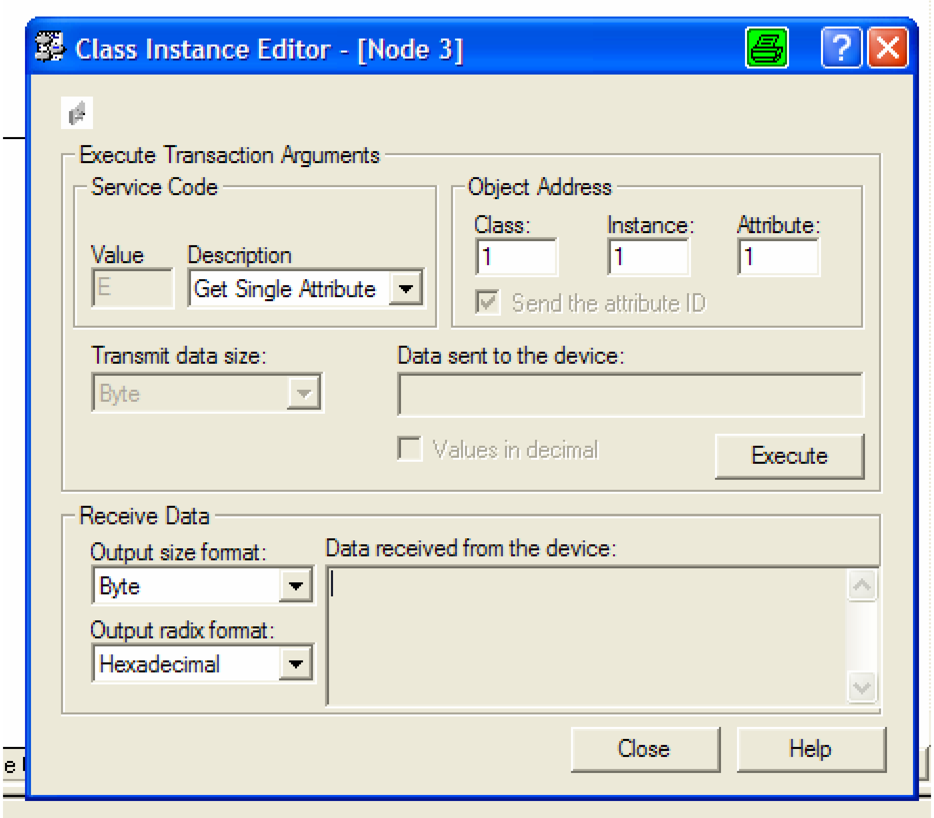


Figure 9–12: Class Instance Editor 2

9.6 I/O Daten editieren

To be able to edit I/O data, a Network Scanner must be available. Prior to this, the network devices to be edited shall be entered in the Scanlist of the scanner (please do not confuse with the “Scanning the Network for Subscribers” function in 9.2).

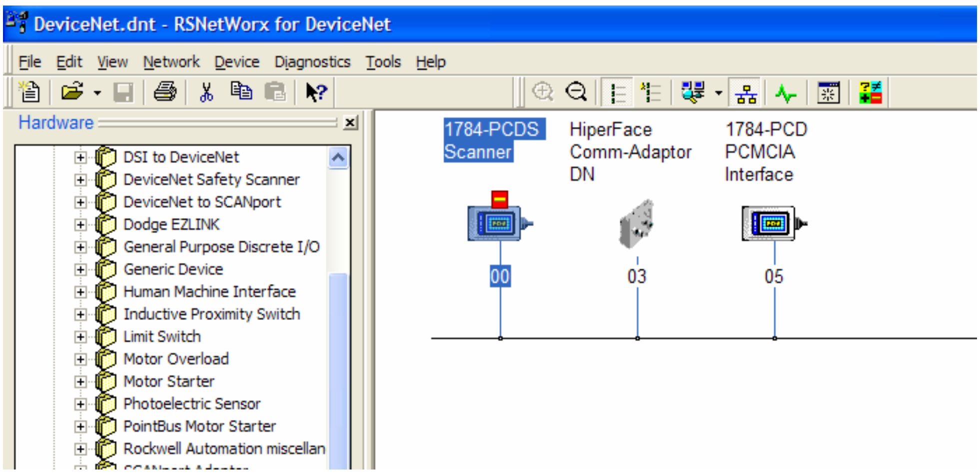


Figure 9–13: Network Scanner

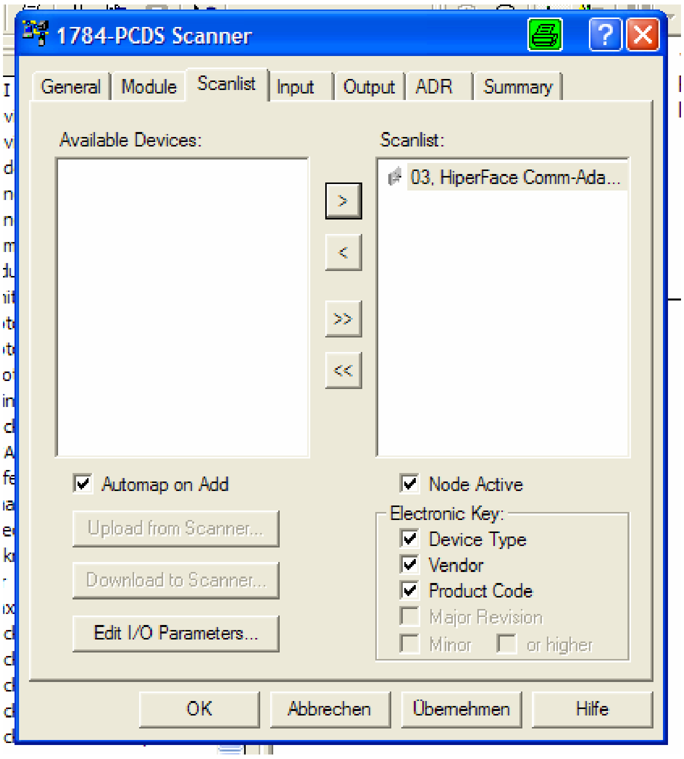


Figure 9–14: Network Scanner - Scanlist

I/O Editor:

Here, it can be defined in which of the Strobed, COS/Cyclic or Polled operating modes a network subscriber is to work (always ensure the correct setting of the required data lengths).

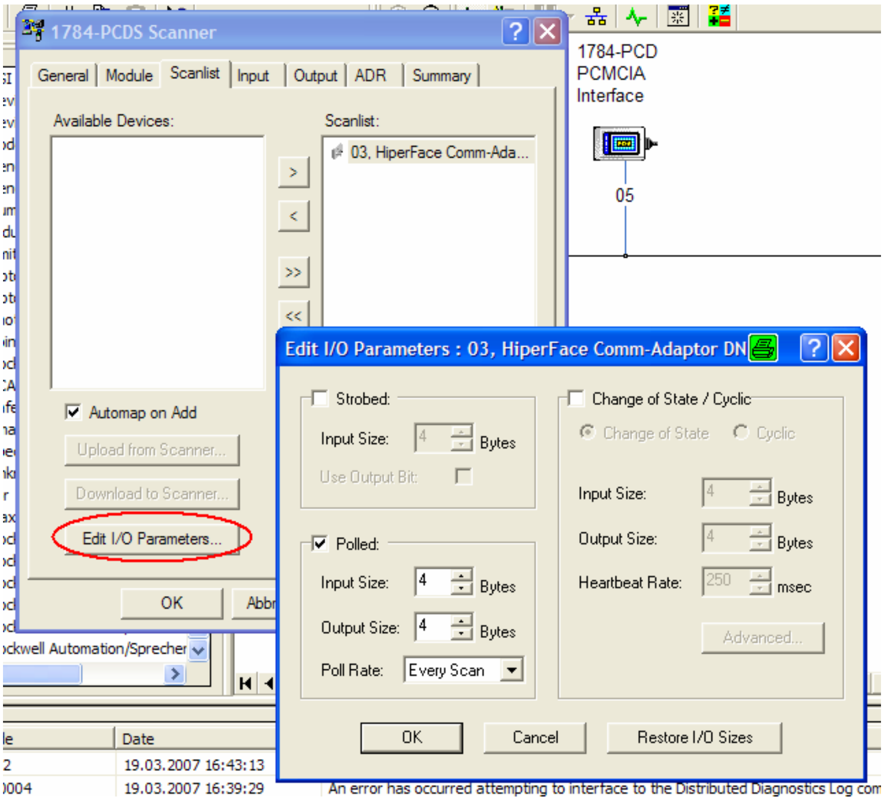


Figure 9–15: Network Scanner – I/O Editor

In addition to the correct setting of the I/O data values when setting the operating mode of a device (Scanlist, submenu: I/O Parameters), correspondingly, the correct setting of the size of the

I/O data for the device for input and output must also be effected (Data Mapping). If the second setting is forgotten then, in case of allocations which are too small, parts of the data words transmitted to the controller will be missing during data evaluation.

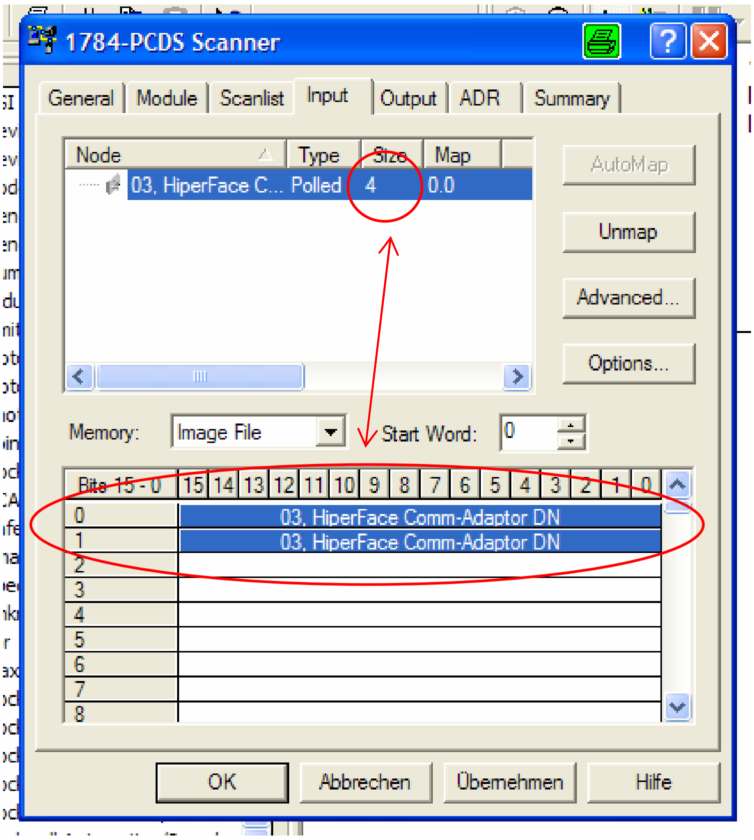


Figure 9–16: Network Scanner – Input/Output

9.7 Network / Node Diagnostics

RSNetworx provides a user-friendly diagnostics facility, to show the state of the individual subscribers in the network.

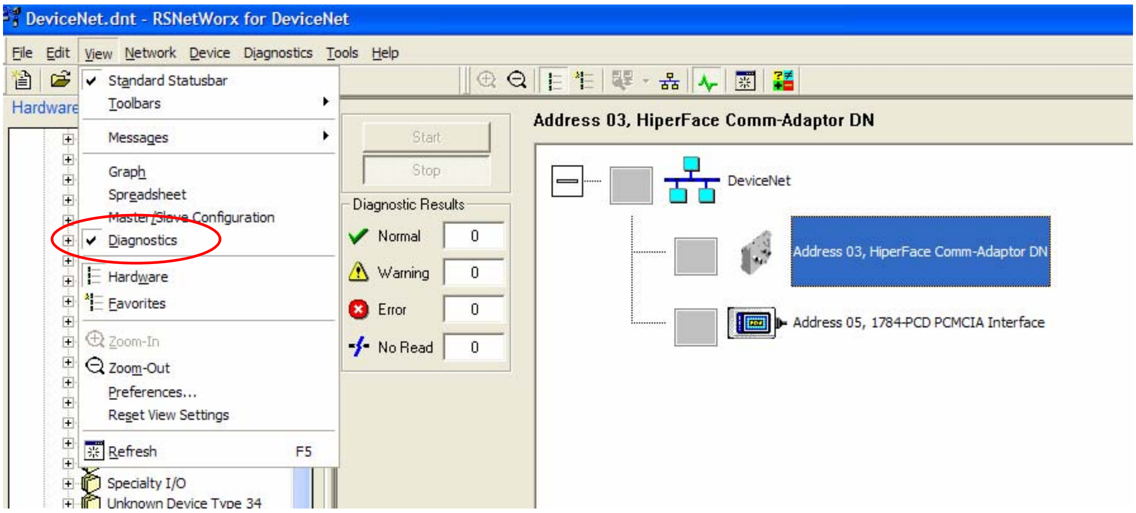


Figure 9–17: Diagnostics – Device Status in the Network

10 Device Ramp-up to the Operational State

The following requirements must be met for this:

- The Hiperface Devicenet Adapter is correctly connected to the Devicenet network
- The corresponding encoder is correctly connected to the Hiperface Devicenet Adapter
- The node address (MAC ID) and the baud rate are correctly set according to the Devicenet network specification
- The Hiperface Devicenet Adapter parameters are configured corresponding to the requirements of the application
- The Devicenet network meets the specification regarding cabling and termination

⚠	To commission the Hiperface Devicenet Adapter, the Safety Notes from Chapter **Fehler! Verweisquelle konnte nicht gefunden werden.** must be observed.

10.1 Operating Modes

The Hiperface Devicenet Adapter has the following operating modes:

- Power-up / Reset Mode
- Run Mode
- Error Mode

During a Power-up or Reset Mode

- the Hiperface Devicenet Adapter performs a Diagnostic Test including an EEPROM test and a self-test.
- the Hiperface Devicenet Adapter reads the node address (MAC-ID) and checks for duplicate addresses issued in the network.

☞	The DIP switch setting is only read in the Reset Mode

10.2 Power-up Characteristic of the Hiperface Devicenet Adapter

The Devicenet network State Machine, which is implemented in all devices, describes tasks which must be performed prior to the communication via the network.

During the self-test (LED flashing green – see also 8.7.1) the Hiperface Devicenet Adapter-specific data is loaded by the EEPROM into RAM, and the objects used from the Devicenet Objects Model are initialised. EEPROM reading errors are registered and will set an alarm flag.

After the self-test the device sends, within one second, two successive identical request telegrams, to check the network for duplicate addresses issued (MAC IDs). These message telegrams contain the manufacturer ID (Vendor ID – 2 bytes) and the device identification via a unique device serial number (4 bytes).

The following representation describes such a message telegram, sent by a device with MAC ID $3F_H$ (63_{dec}) and the Vendor ID from Sick-Stegmann (511_{dec}):

Identifier (bit 10...0)			Data (8 bytes)				←			
Gr	MAC ID	MSG ID	Head	Data_0	Data_1	Data_2	Data_3	Data_4	Data_5	Data_6
10	1.1111-1	111	00	FF	01	SN_Lo	...	...	SN_Hi	

Identifier	5.FF$_H$
MSG ID	[07]: Duplicate MAC ID Check Message
Head	Port Number: [**0**xxx-xxxx]$_B$ Request Message, [**1**xxx-xxxx]$_B$ Response Message
Data_0..._1	Vendor ID (Low Byte, High bytes):): -- Stegmann **[01.FF]**$_H$
Data_2..._5	**(*1)** Serial Number (Low Byte...High bytes)

(*1) The **<u>Serial Numbers</u>** are issued according to this scheme:

32 BIT (4 BYTE) - No.			
SN_Lo+3 = SN_Hi	**SN_Lo+2.**	**SN_Lo+1.**	**SN_Lo**
31 30 29 28 27 26 25 24	23 22 21 20 19 18 17	16 15 14 13 12 11	10 9 8 7 6 5 4 3 2 1 0
Device Code, 65 $_{dec}$	Year, {0 – 99 $_{dec}$}	Week {1 – 52 $_{dec}$}	Consecutive Number {0 – 2047 $_{dec}$}

The Device Code is an internal definition and can differ from the designation on the type label.

If, when executing the 'Reset Mode', the device receives a message (DUP MAC ID Check Request/ Response Message) via a duplicated node address (MAC ID) or if a 'CAN BUS OFF' state was detected, the device changes to an 'Error Mode' (Communication Fault State). If, during device ramping, no error was detected, the Hiperface Devicenet Adapter changes to the 'RUN Mode' a can be addressed by a Master, as a slave device.

> To exit the 'Error Mode' in case of a critical and irrecoverable error, the supply voltage must be turned off or the device must be ‘uncabled’ from the network.

10.3 State Machine

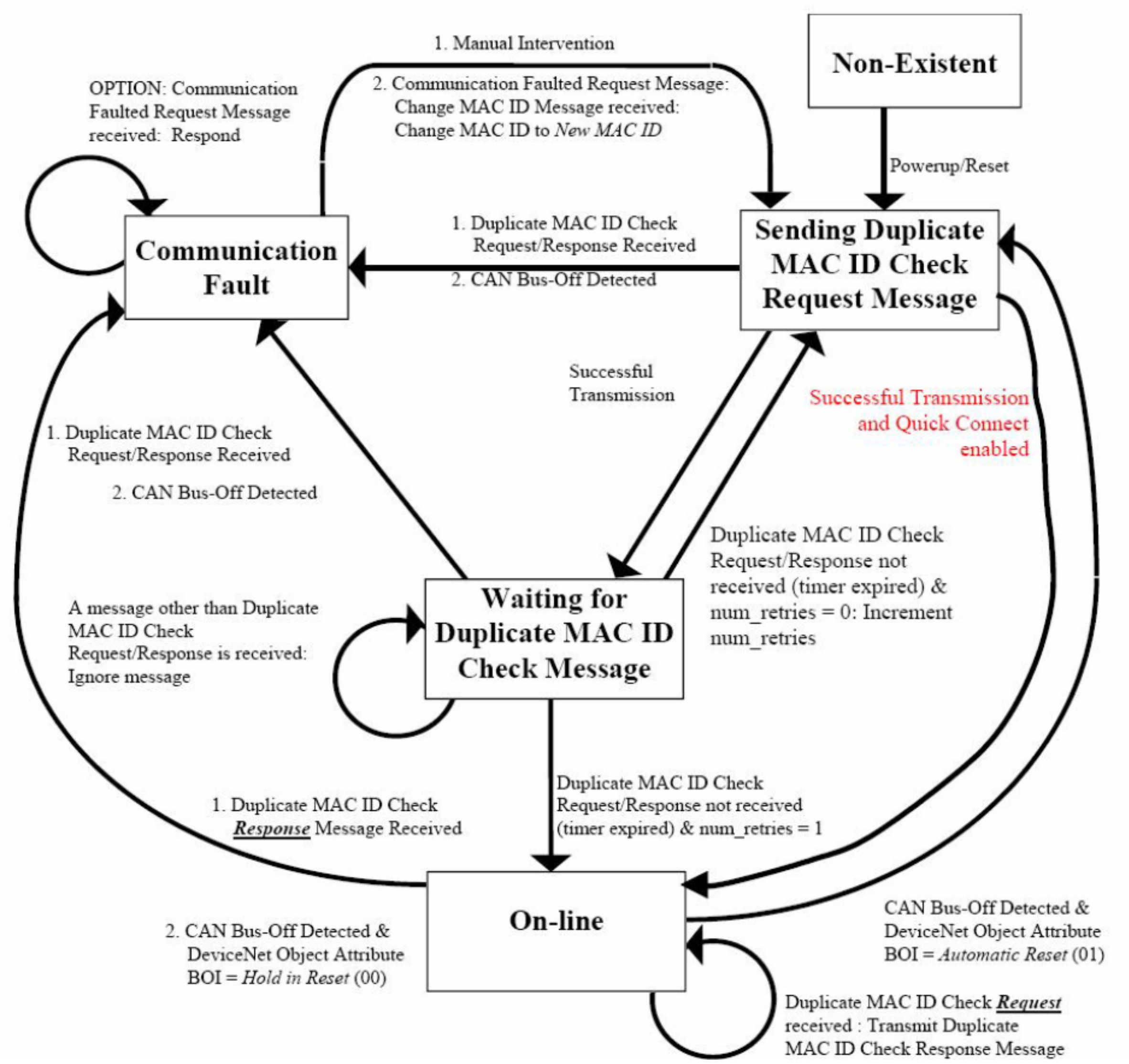


Figure 10–1: State Machine

11 General and Brief Technical Description

11.1 Description of the Rotary Encoders

The absolute rotary encoders with Hiperface interface have up to 1024 sine/cosine periods resolution per revolution. The Hiperface Devicenet Adapter reads in the digital encoder resolution (via the RS485 parameter channel) and then refines the resolution by evaluating the information of the analogue sine/cosine signals (process data channel). The number of revolutions is determined via a gearbox mechanism in the encoder.

11.2 Description of the Linear Encoders

A – non-contact – reading head determines the absolute position along a material measure (e.g. a magnetic tape with sequential code), which is fitted along the measurement path. The reading head itself consists of a number of sensors, which determine the position absolutely – The Hiperface Devicenet Adapter reads, from the reading head, the encoder resolution (via the RS485 parameter channel) and then refines the resolution by evaluating the information of the analogue sine/cosine signals (process data channel).

Wire drive encoders also belong to the linear encoder group although, internally, they have an integrated rotary encoder. Here, the length of measuring wire pulled-out is mapped via a measuring drum, to the position value of a rotary encoder. For the customer, the system is handled like a linear encoder.

11.3 Devicenet Interface – Brief Overfiew of Specific Features

- Bus interface to ISO 11898 CAN High Speed for the CAN Specification 2.0B, galvanically separated.
- Functionality according to Devicenet Protocol Specification Release 2.0 Vol. 1 and 3.
- Device profiles acc. to DN Specification Vol. 1 Chapter 6: Device Profiles (Encoder Device 22h) based on the Position Sensor Object
- Configuration using hardware settings (DIP switches):
  - ❖ Electronic adjustment (PRESET) – call up value from EEPROM
  - ❖ Address setting (MAC ID - 0…63)
  - ❖ Data transmission rate (baud rate – 125…500 kBaud)
- Configurable parameters of the Hiperface Devicenet Adapter via the protocol:
  - ❖ Operating modes: Predefined Master Slave Connection Set and Explicit Messages – (I/O, Polling, Bit-Strobe und COS)
  - ❖ Counting direction
  - ❖ Scaling parameters (CPR, CMR)
  - ❖ Electronic adjustment (PRESET)
  - ❖ Address setting (MAC ID)
  - ❖ Data transmission rate (baud rate)
  - ❖ Units for velocity and acceleration
  - ❖ Switching cam adjustments (4 channels of 8 CAMs each)
- Storage of all important data parameters in the EEPROM.
- Formation of new position values at **intervals of < = 25 ms**.

Status information	Network status (red/green), operating voltage (green), [Hiperface (yellow)].
Bus termination	external M12 bus termination resistance -- (only fitted to the last bus subscriber of the line).
Electrical connection	3 x screw-in system M12 (Bus In – 5-way socket, Bus Out – 5-way pin, [encoder – 8-way socket])

[ ] Concerning Hiperface connection

. Appendix A - Problem Solving

DeviceNet will not function correctly if design rules are not followed. Even a network previously thought to be functioning correctly may begin to exhibit anomalous operation due to incorrect system design.

To solve problems that can occur, you need to understand the physical DeviceNet layout, the devices on the network and how all pieces work together.

Diagnostic tools and device indicators help identify the operational state of devices and network communication problems.

Cable Installation and Design Problems

Cable installation and design has to do with the physical layout and connections on the network. Walk the network if possible to determine the actual layout.

Ensure that you have a diagram of the physical layout and a record of the following information on the layout.

- Number of nodes.
- Individual, branched and cumulative drop lengths.
- Total trunk length, including long drop near the ends.
- Power supply cable length and gauge.

- Terminator locations and size.
- Break the earth ground of the V- and Shield and verify >1.0 MOhm to frame ground with power supply off.
- Check for short circuits between CAN_H and CAN_L, or CAN_(H, L) to Shield, V-, V+.
- Length and gauge of the earth ground connection.
- Total power load and its distribution points.

LED Status Check

The Network Status LED shows the status at power-up and during operation of the network.

LED	Status	Indicates	Action
Off	• Not Powered • Not On-Line	• No power to device • Failed Duplicate MAC ID check	• Check that one or more nodes are communicating on the network • Check that at least one other node on the network is operational and the data rate is the same as the Hiperface-Devicenet-Adapter
Flashing Green	• On-Line • Not connected	• Passed Duplicate MAC ID check • And No connection established	• No action is needed. The LED is flashing to signify that there are no open communication connections between the Hiperface-Devicenet-Adapter and any other device. Any connection (IO or explicit message) made to the encoder will cause the LED to stop flashing and remain Steady-ON for the duration of any open connection.
Steady Green	• On-Line • Connected	• One or more connections established	• No action is needed.
Flashing Red	• On-Line • Time-Out	• At least one IO connection has timed out	• Re-initiate IO messaging by master controller • Reduce traffic or errors on the network so that messages can get through.

LED	Status	Indicates	Action
Steady Red	• Network Failure	• Failed Duplicate MAC ID check • Bus-OFF	• Ensure that all nodes have unique addresses. • If all node addresses are unique, examine network for correct media installation. • Ensure that all nodes have the same data rate.

If the Network Status LED goes Steady-Red at power-up, it could mean there is a Duplicate MAC ID. The user response should be to test all devices for unique addresses. If the symptom persists, it means a Bus-OFF error.

- Check data rate settings.
- If symptom persist, replace node address (with another address and correct data rate).
- If symptom persists, check the topology.
- If symptom persists, check power for noise.

Scanner Problems

If using a scanner, check the scan list, data rate and addresses of devices. Verify series and revision of the scanner.

If the scanner goes Bus-OFF after a reset the problems is some combination of:

- defective node device.
- incorrect node data rate.
- bad network topology.
- faulty wiring.
- faulty scanner.
- faulty power supply.
- bad grounding and / or electrical noise.

Wiring Problems

Various situations in and around cables can cause problems on the network. Things to do are:

- check that connectors and glands are screwed tightly.
- check connectors for contamination by foreign materials.
- check that nodes are not touching extremely hot or cold surfaces.
- check that cables are kept a few inches away from power wiring.
- check that cable are not draped on electrical motors, relays, contactors or solenoids.
- check that cables are not constrained so as to place excessive tension on connectors.

Power Supply Problems

Add up the current requirements of all devices drawing power from the network. This total should be the minimum current rating in selecting the power supply used. In addition check:

- length and current level in trunk and drop cables.
- size and length of the cable supplying power to the trunk.
- voltage measured at the middle and ends of the network (see also Appendix B – Common Mode Voltage).

Adjusting the Physical Network Configuration

Some ways to help improve the efficiency of your physical network configuration are:

- Shorten the overall length of the cable system.
- Move the power supply in the direction of an overloaded cable section.
- Move higher current loads closer to the power supply.
- Add another power supply to an overloaded network.
- Move the power supply from the end to the middle of the network.

Points to remember

- Pressing the reset button on the scanner does not reset the network.
- Cycling power on the network can cause the scanner to go Bus-OFF.
- On some DeviceNet nodes (such as photoelectric sensors) the Bus-OFF (solid red) condition can be cleared by cycling the 24V power or by pressing a reset button once or twice.
- Extreme care must be given to the task of setting initial addresses and baud rates because one incorrect node can cause other nodes to appear to be bad (solid red).
- If the scanner is Bus-OFF, nodes will not necessarily reallocate (they will stay flashing green or red) even if they are functioning correctly.
- DeviceNet management and diagnostic tools can be used to identify the functioning nodes on the network, and their type. When devices are reset or re-power they will transmit two Duplicate MAC-ID Check Request messages which provide a convenient method of verifying baud rate, MAC-ID, Vendor ID, and Serial Number of the device.
- If a node goes Bus-OFF (solid red indicator), and is replaced and still goes Bus-OFF, the problem is not the node but rather the setting of the address or data rate OR a network wide problem related to topology, grounding, electrical noise or an intermittent node.
- Intermittent power connections to nodes will provoke frequent (but perhaps incomplete) duplicate MAC-ID checks and possibly cause other nodes to go Bus-OFF.
- Intermittent data connections to a strobed node will cause corrupted frames that also may cause other nodes to go Bus-OFF. The source of the difficulty may be far from the node which evidences the symptom.

. Appendix B – Common Mode Voltage

Common Mode Voltage

When current is drawn through the power pair on the Devicenet trunk line, the length of the cable and current draw becomes important. The thick wire, normally used for trunk line, has a resistance of 0.0045 Ohms/ foot. So as the distance from the DeviceNet power supply connection becomes greater, the power pair will act as a resistor whose value will be equal to 0.0045 multiplied by the Distance from the Power Supply (In Feet). At any particular point on the power pair the Common Mode Voltage will equal the Current being drawn on the power pair at that point times the Resistance of the Power Pair. (V = I x 0.0045 x Distance)

The effect of the Common Mode Voltage is that the V+ line will lower gradually from the 24VDC at the power supply as you move farther down the trunk line. More importantly the V- wire will gradually raise from the 0VDC value at the power supply along the length of the trunk line. On most networks the amount of voltage the V+ lowers and V- raises are equal. So even though there may be exactly 24VDC measured at Network Power Supply, further down the cable the voltage on the V+ and V- wires may only be 20VDC. This effect is due to Common Mode Voltage and should the voltage drop become too large the network will fail to operate properly. Since the CAN-H (White Wire) and the CAN-L (Blue Wire) both are referenced to the V- wire, if the V- line varies more than 4.65VDC at any two points on the network the CAN transceivers will fail to operate properly.

An easy way to measure for Common Mode Voltage problems is to go to the farthest ends of the network and measure between Red V+ and Black V- wires. This voltage should NEVER be less than 15 Volts.

Network Voltage/ OHM Readings

It needs to be understood that DeviceNet is actually a three wire Differential Voltage network. Communication is accomplished by switching the CAN-H (White wire) and CAN-L (Blue wire) signals relative to the V- line (Black Wire).

Important NOTE

The CAN to V- voltages given in the rest of this chapter assume **NO Common Mode Voltage** effect is occurring anywhere on the V- wire of the network. On a network with Common Mode Voltage influence, the voltages will be higher depending on where you take the measurement. Nodes closest to the power supply will exhibit voltages higher due to the Common Mode Voltage, while nodes at the farthest end of the network away from the power supply will exhibit lower voltages.

The CAN-H swings between 2.5 VDC (Recessive State) and 4.0 VDC (Dominant State) while the CAN-L swings between 1.5 VDC (Dominant State) and 2.5 VDC (Recessive State)

Without a network master connected to the DeviceNet, the CAN-H and CAN-L lines should read between 2.5 VDC and 3.0 VDC relative to V- and the voltages should be identical (Recessive State). Measure these voltages right at the SDN scanner connection which is normally also where the power supply is connected to the network. Use a voltmeter in DC mode.

With a network master connected to the DeviceNet **and polling the network,** the CAN-H to V- voltage will be around +3.2 VDC. The CAN-L to V- voltage will be around 2.4 VDC. The reason these values appear a little different than the ranges shown on the scope trace, is that the signals are switching, which slightly affects the DC value being read by the VOM.

With the 24VDC power supply not energised you can measure the resistance between the CAN-H and CAN-L signals. The ohm reading between the CAN-H and CAN-L lines should be 60 ohms (two 120 ohm resistors in parallel), however with a large amount of devices connected to the network the resistance could be as low as 50 Ohms.

. List of Figures

. Notes

9149676/2012-01-30 · Subject to change without notice · A4 2c int37

Australia
Phone +61 3 9497 4100
1800 334 802 – tollfree
E-Mail sales@sick.com.au

Belgium/Luxembourg
Phone +32 (0)2 466 55 66
E-Mail info@sick.be

Brasil
Phone +55 11 3215-4900
E-Mail sac@sick.com.br

Canada
Phone +1(952) 941-6780
1 800-325-7425 – tollfree
E-Mail info@sickusa.com

Ceská Republika
Phone +420 2 57 91 18 50
E-Mail sick@sick.cz

China
Phone +852-2763 6966
E-Mail ghk@sick.com.hk

Danmark
Phone +45 45 82 64 00
E-Mail sick@sick.dk

Deutschland
Phone +49 211 5301-301
E-Mail kundenservice@sick.de

España
Phone +34 93 480 31 00
E-Mail info@sick.es

France
Phone +33 1 64 62 35 00
E-Mail info@sick.fr

Great Britain
Phone +44 (0)1727 831121
E-Mail info@sick.co.uk

India
Phone +91–22–4033 8333
E-Mail info@sick-india.com

Israel
Phone +972-4-999-0590
E-Mail info@sick-sensors.com

Italia
Phone +39 02 27 43 41
E-Mail info@sick.it

Japan
Phone +81 (0)3 3358 1341
E-Mail support@sick.jp

Magyarország
Phone +36 1 371 2680
E-Mail office@sick.hu

Nederlands
Phone +31 (0)30 229 25 44
E-Mail info@sick.nl

Norge
Phone +47 67 81 50 00
E-Mail austefjord@sick.nl

Österreich
Phone +43 (0)22 36 62 28 8-0
E-Mail office@sick.at

Polska
Phone +48 22 837 40 50
E-Mail info@sick.pl

România
Phone +40 356 171 120
E-Mail office@sick.ro

Russia
Phone +7 495 775 05 30
E-Mail info@sick.ru

Schweiz
Phone +41 41 619 29 39
E-Mail contact@sick.ch

Singapore
Phone +65 6744 3732
E-Mail admin@sicksgp.com.sg

Slovenija
Phone +386 (0)1-47 69 990
E-Mail office@sick.si

South Africa
Phone +27 11 472 3733
E-Mail info@sickautomation.co.za

South Korea
Phone +82-2 786 6321/4
E-Mail info@sickkorea.net

Suomi
Phone +358-9-25 15 800
E-Mail sick@sick.fi

Sverige
Phone +46 10 110 10 00
E-Mail info@sick.se

Taiwan
Phone +886 2 2375-6288
E-Mail sales@sick.com.tw

Türkiye
Phone +90 216 528 50 00
E-Mail info@sick.com.tr

United Arab Emirates
Phone +971 4 8865 878
E-Mail info@sick.ae

USA/México
Phone +1(952) 941-6780
1 800-325-7425 – tollfree
E-Mail info@sickusa.com

More representatives and agencies
at **www.sick.com**

Sensor Intelligence.