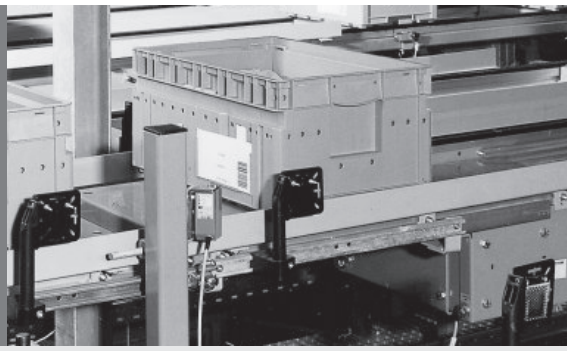
OPERATING INSTRUCTIONS



CMF400-2101
Fieldbus Gateway

Connecting a SICK Bar Code Scanner or Image Code Reader to DeviceNet

SICK
Sensor Intelligence.

Production status of the CMF400-2101

Device type	Designation	Revision index
CMF400-2101	Field bus gateway for DeviceNet	0000

Trademarks

Windows 95/98™, Windows NT™, Windows 2000™, Windows XP™ and Internet Explorer™ are registered trademarks or trademarks of the Microsoft Corporation in the USA and other countries.

Latest manual version

For the latest version of this manual (PDF), see **www.sick.com**.

Contents

1 Product Description 4
1.1 Features 4
1.2 Scope of delivery 4
1.3 Prerequisites for Installation and Commissioning 4
1.4 Basic Functions 5
2 Installation 7
2.1 Installation location in the connection module 7
3 Electrical Installation 8
3.1 Electrical Connections 8
4 Commissioning 9
4.1 Prerequisites 9
4.2 Configuration: Preparatory Steps 9
4.3 Configuring the Bar Code Scanner and Gateway Using CLV Setup 9
4.4 Configuring the Gateway with ComPro 13
5 Communication via DeviceNet 14
5.1 Data Exchange 14
5.2 Standard CMF Format, Byte Handshake, Digital I/O 14
5.3 Standard CMF Format, No Byte Handshake, Digital I/O 18
5.4 The DigitalInput and DigitalOutput Bytes 19
6 Troubleshooting 21
6.1 System LED: "SYS" (V3) 21
6.2 DeviceNet: LED "COM" (V2) 21
6.3 Bar Code Scanner LED: "SER" (V1) 21
7 Technical Data 23
7.1 Data Sheet: Fieldbus Gateway CMF400 for DeviceNet 23
7.2 Accessories 24
8 Appendix 25
8.1 EC Declaration of Conformity 25

Abbreviations used

LED **L**ight **E**mitting **D**iode

LSB **L**east **S**ignificant **B**yte

MSB **M**ost **S**ignificant **B**yte

OVDA **O**pen **D**eviceNet **V**endor **A**ssociation

PLC **P**rogammable **L**ogic **C**ontroller

The Bar Code Scanners and the Image Code Readers are simply referred to as “bar code scanner“ in the document.

1 Product Description

1.1 Features

- Field bus gateway as a plug-in module for connecting SICK bar code scanners CLV42x to 45x, ICR84x/85x, CLV480, CLV490, or CLX490 to DeviceNet.
- Can be used in the CDM420 and CDM490 connection modules (basic devices).
- Connection to the motherboard via the SMD connector.
- 5-pin shielded M12 plug on the front panel for connection to DeviceNet.
- Additional 2 digital inputs and outputs on the gateway.
- 3 LEDs for displaying statuses and malfunctions.
- 18 to 30 V DC power supply via the connection module.
- Configuration via the user interface of the connected bar code scanner.
- Certified by the **O**pen **D**eviceNet **V**endor **A**ssociation (OVDA).

1.2 Scope of delivery

- CMF400-2101 field bus gateway for DeviceNet
- Front panel set for the connection module with M12 plug for the bus connection
- Installation set
- "Manuals & Software Bar Code Scanners" CD (no. 2029112)

Note The latest versions of all the current publications/programs on the CD can also be downloaded from **www.sick .com**.

Further product information:

➢ See www.sick.com/cmf

EC delaration of conformity:

➢ On request

1.3 Prerequisites for Installation and Commissioning

- CDM420 or CDM490 connection module with operating instructions.
- Bar code scanner with CMF400-compatible firmware and operating instructions (see *Chapter 7.1 Data Sheet: Fieldbus Gateway CMF400 for DeviceNet, Page 23*).
- EDS file "SICK0808.EDS" for CMF400-2101 (on the "Manuals & Software Bar Code Scanners" CD).
- PC with "CLV Setup" user software from version 3.8 for configuring the CMF400 (also on the CD).
- 3-core RS 232 data cable (null modem cable) for connecting the PC to the CDM connection module
- If ComPro configuration tool is used: additionally the parameterization cable no. 2030490 (see also *Chapter 4.4 Configuring the Gateway with ComPro, Page 13*)

1.4 Basic Functions

1.4.1 Operating Principle

The CMF400-2101 field bus gateway is used for connecting a SICK bar code scanner to DeviceNet. The gateway enables user data to be transferred transparently between the bar code scanner and a field bus master (PLC). In the field bus, the gateway always acts as the slave. Data is exchanged cyclically with the field bus master. The bar code scanner can be connected to the gateway via its host interface (RS 232 variant) or the terminal interface (RS 232).

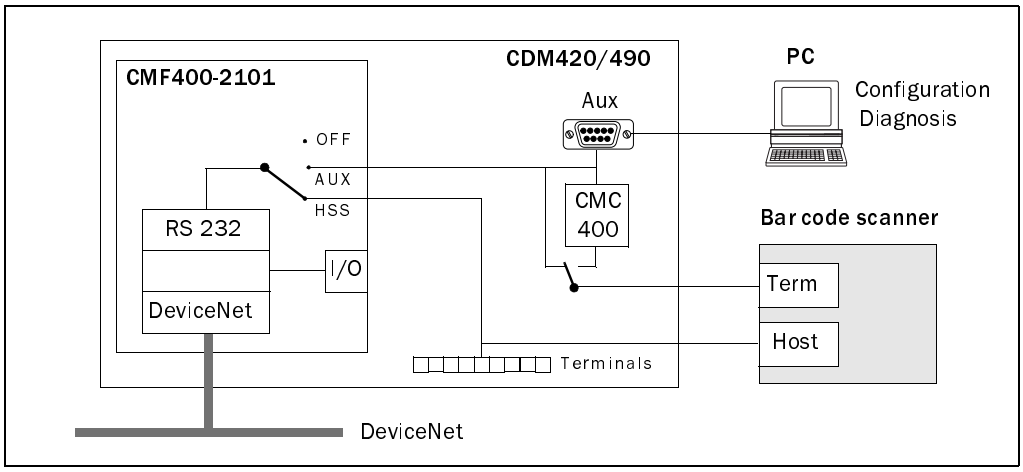


Fig. 1-1: Block diagram of the field bus gateway for DeviceNet in the connection module

Instead of a bar code scanner, a different device with an RS 232 interface and STX/ETX frame can also be connected to DeviceNet via the gateway. This enables hand scanners, for example, to be easily integrated in DeviceNet networks
(*see also* *Chapter 4.4 Configuring the Gateway with ComPro, Page 13*).

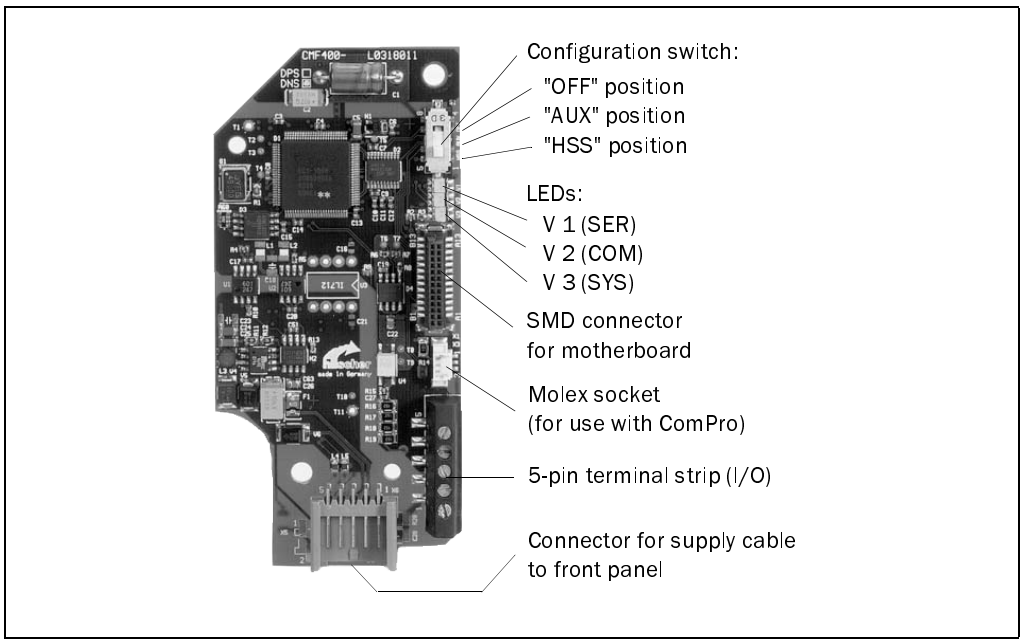


Fig. 1-2: Structure of the field bus gateway for DeviceNet

1.4.2 Configuration Switch and Displays

Configuration switch:

Function	Default
Switching the RS 232 gateway interface to different sources: HSS: Host interface of the bar code scanner AUX: Terminal interface of the bar code scanner OFF: Complete disconnection from the bar code scanner	HSS

Table 1-1: Function of the gateway configuration switch

Note Use the "OFF" (or "HST") position to configure or diagnose the connected bar code scanner via the AUX connector in the connection module.

LEDs:

LED	Function	Color	Status	Description
V1	SD	*Green*	On	Communication with bar code scanner OK
			Off	No communication with bar code scanner for 200 ms
	XOFF	*Yellow*	On	Gateway has sent XOFF. LED extinguishes when XON is sent.
			Off	Initial state (OK)
			Regular flashing	Gateway receive buffer or transmit buffer has overflowed. Terminate flashing by releasing receive and/or transmit buffer.
V2	MNS	*Green*	On	Gateway connected to the bus, communication establised
			Flashing	Gateway connected to the bus, no communication
		Red	On	critical link error, critical error
			Flashing	subordinated error and/or time monitoring error
		Green/Red	Flashing	Communication aborted
		-	Off	No power supply, not connected to the bus
V3	RUN	*Green*	On	Communication active
			Irregular flashing	Parameterization error
			Regular flashing	Ready to communicate
			Off	No communication
	RDY	*Yellow*	On	Gateway ready
			Cyclic flashing	Bootstrap loader active
			Irregular flashing	Hardware or system error
			Both off	Hardware defect or no power supply

Table 1-2: Function of the LEDs on the gateway

2 Installation

2.1 Installation location in the connection module

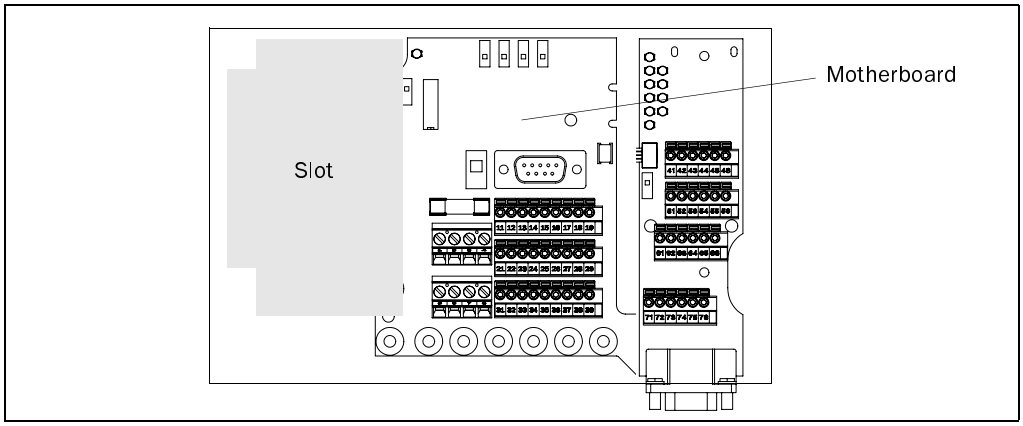


Fig. 2-1: Slot for the field bus gateway for DeviceNet in the connection module

WARNING

Risk of damage to the gateway due to electrostatic charge.

Do not touch the gateway without equipotential bonding because electrostatic discharge from your body may damage electronic components.

- ➢ Before removing the module from the ESD pouch, carry out equipotential bonding between the body and shield of the connection module.
- ➢ Handle the module on the sides only.
- ➢ When carrying out installation work, you are advised to wear a grounding wrist band.

2.1.1 Installing the Gateway in the Connection Module

1. Loosen the screws in the cover of the connection module and remove the cover.
2. Disconnect the connection module from the power supply.
   To do so, set switch S 1 (power) to the "OFF" position.
3. Align the gateway in such a way that the grey connector points towards the front of the connection module. Insert the module to the left of the motherboard (*Fig. 2-1*). The SMD connector must rest on the mating connector of the motherboard.
4. Secure the module using the two threaded pins provided in the installation kit.
5. Unscrew the old metal front panel and remove the shield connector.
6. Connect the shield connector to the new front panel (5-pin M12 plug for DeviceNet).
7. Screw the new front panel into place. The shield must be located at the bottom of the connection module.
8. Connect the M12 plug connection cable on the front panel to the grey gateway connector.
9. Connect the gateway shield (PE, terminal 1 on the 5-pin terminal strip) to the shield (terminal 6, 7, or 8) on the motherboard using the cable supplied.
10. Switch on the power supply for the connection module with switch S 1.
11. Replace and secure the cover.

3 Electrical Installation

3.1 Electrical Connections

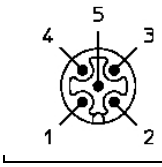
Note Diagrams showing you how to connect the CMF400 for DeviceNet are also available in the "CLV Connect" PC program. The software is available on the "Manuals & Software Bar Code Scanners" CD.
The software can also be downloaded from the SICK home page (www.sick.com) in the Internet. It can be called up using a standard HTML browser (e.g. Internet Explorer™).

3.1.1 Pin Assignment for the 5-Pin Terminal Strip (Gateway)

Pin	Signal	Function
1	PE	Shield (connection to connection module)
2	IN 1	Digital input (U_{in} = DC 0 to 30 V)
3	IN 2	Digital input (U_{in} = DC 0 to 30 V)
4	OUT 1	Digital output (U_{out} = DC 0 to V_s, max. 30 V)*)
5	OUT 2	Digital output (U_{out} = DC 0 to V_s, max. 30 V)*)
*) cable length max. 30 m (98.4 ft)		

Table 3-1: Pin assignment for the 5-pin terminal strip (gateway)

3.1.2 Pin Assignment of the 5-Pin M12 plug (DeviceNet) on the Front Panel



Pin	Signal	Function
1	Drain	Shield
2	V+	+ 24 V
3	V–	0 V
4	CAN H	Receive/transmit data - High
5	CAN L	Receive/transmit data - Low

Table 3-2: Pin assignment of the 5-pin M12 plug (DeviceNet)

Note Use resistors to terminate the backbone of the network at both ends.
Drop lines do not require a bus terminator.

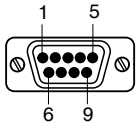
Conditions for DeviceNet

The maximum length of the DeviceNet depends on the data transmission rate used (*Table 3-3*) and must not exceeded.

Data transmission rate	Cable length of segment
125 kbit/s	Max. 500 m (1,640.4 ft)
250 kbit/s	Max. 250 m (820.2 ft)
500 kbit/s	Max. 100 m (328 ft)

Table 3-3: Data transmission rate as a function of the cable length

3.1.3 Pin Assignment for the 9-Pin D Sub Plug “Aux“ in the Connection Module



Pin	Signal	Function
2	RxD	Receiver
3	TxD	Transmitter
5	GND	Signal ground

Table 3-4: Pin assignment for the 9-pin D Sub plug “Aux“ (RS 232)

4 Commissioning

4.1 Prerequisites

To integrate the field bus gateway in DeviceNet, the gateway must be registered with the field bus master. SICK provides a device master file (EDS) for this purpose. This file must be transferred to the device database of the master and contains all the required device-specific gateway properties. The current version of the EDS file is available on the "Manuals & Software Bar Code Scanners" CD. The EDS file can also be downloaded from the SICK home page (www.sick.com/cmf).

4.2 Configuration: Preparatory Steps

Note **When configuring the gateway using the "CLV Setup" configuration software (PC connected to the terminal interface of the bar code scanner), ensure that the gateway configuration switch is not in the "Aux" position because the bar code scanner is then connected to the gateway and PC at the same time.**

1. Set the gateway configuration switch to "HSS".
2. Connect the bar code scanner to the connection module.
3. Switch on the power supply for the connection module (switch S 1 to ON).
4. Connect the PC to the terminal interface of the bar code scanner.
   To do so, connect the 3-core RS 232 data cable (null modem cable) to the internal 9-pin "Aux" connector on the connection module.
5. Switch on the PC and install the "CLV Setup" configuration software on the "Manuals & Software Bar Code Scanners" CD if you have not already done this.
6. Start the "CLV Setup" software.
   When it successfully establishes communication with the bar code scanner, CLV Setup copies the current scanner parameter set and displays the values on the tabs.
7. Configure the bar code scanner and field bus gateway as described below.

4.3 Configuring the Bar Code Scanner and Gateway Using CLV Setup

Note The bar code scanner firmware must support the gateway
(see *Chapter 7.1 Data Sheet: Fieldbus Gateway CMF400 for DeviceNet, Page 23*).

The sections below explain how to configure the gateway using bar code scanner CLV42x. If you are using a different type, the bar code scanner-specific tabs in CLV Setup are identical or similar.

4.3.1 Overview

- Configure the host interface of the bar code scanner.
- Configure the DeviceNet network address of the bar code scanner.
- Configure the field bus gateway.
- Download the new parameter set to the bar code scanner.
- Restart the bar code scanner and gateway.

4.3.2 Configuring the Host Interface of the Bar Code Scanner

Note Once the configuration parameters have been downloaded to the bar code scanner, the gateway automatically recognizes the settings using an AutoDetect function after it has been restarted.
All the network nodes must use the same communication parameters (apart from the network address).

➢ Choose the HOST INTERFACE tab.

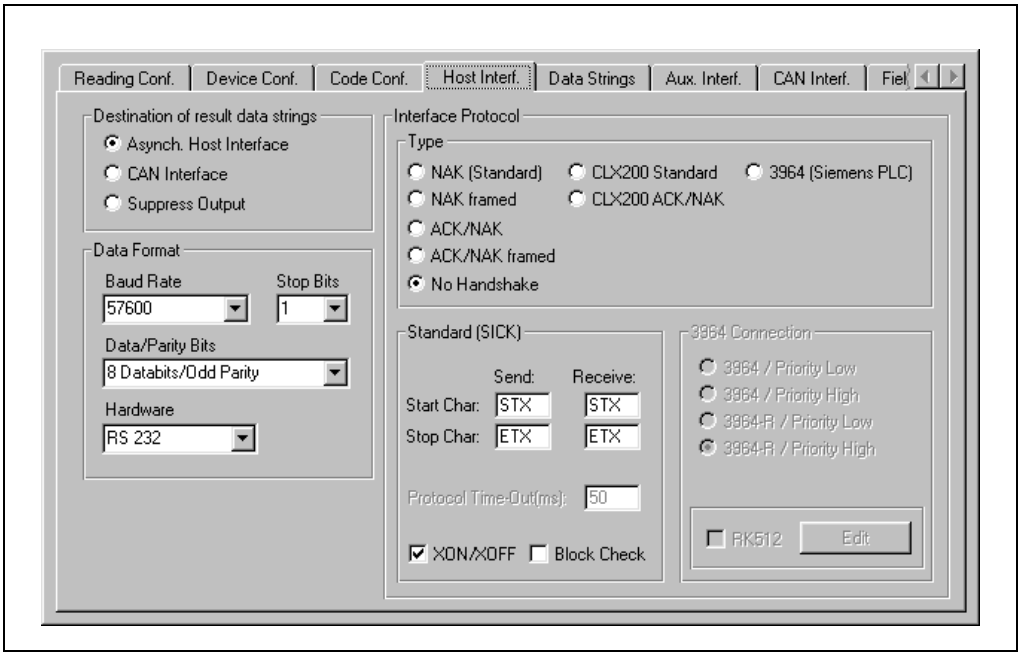


Fig. 4-1: CLV Setup: "Host Interface" tab

➢ Set the following values on the tab:

- Target of the result data string: Asynchronous host interface
- Baud rate: 9,600; 19,200; 38,400; or **57,600** Bd
- Stop bits: 1
- Data/parity bits: 8 bit/no parity or **8 bit/odd parity**
- Hardware: RS 232
- Interface protocol type: **No handshake** or ACK/NAK
- Standard start symbol: Transmit: STX, receive: STX
- Standard stop symbol: Transmit: ETX, receive: ETX
- XON/XOFF: **Active**/not active

The values in **bold** are the default gateway settings.

XON/XOFF:
When the XON/XOFF protocol is active, the gateway can interrupt the data flow from the bar code scanner by sending the control character "XOFF" to the bar code scanner. If the bar code scanner then receives the control character "XON", it resumes data transmission. The gateway sends XOFF when five of the six buffers are full and a 6th telegram is received from the bar code scanner. As soon as at least two buffers are free, the gateway sends XON to the bar code scanner. The bar code scanner only processes the control characters "XON" and "XOFF" at its host interface. The gateway processes a maximum telegram length of 512 bytes (without STX/ETX frame).

4.3.3 Configuring the DeviceNet Network Address of the Bar Code Scanner

➢ Choose the DEVICE CONFIGURATION tab.

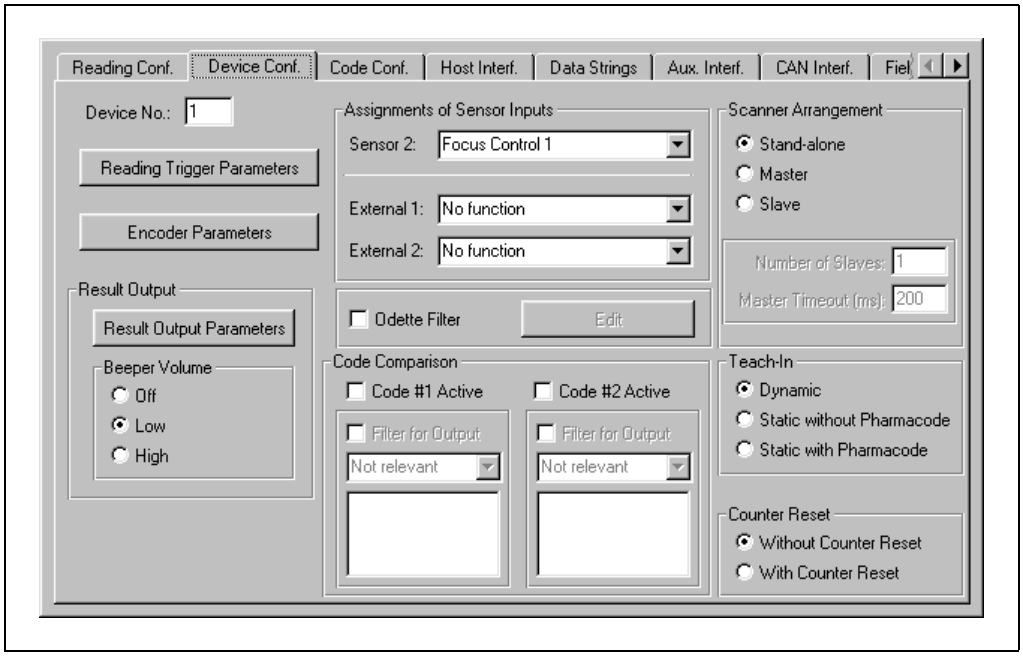


Fig. 4-2: CLV Setup: "Device Configuration" tab

➢ In the DEVICE No. field, assign a unique device address. The node address must be unique in the DeviceNet network (range 1 to 99).

Note If a CMC400 cloning module is also used in the connection module, the device address is set using the two rotary coding switches on the CMC400.

4.3.4 Configuring the Field Bus Gateway

➢ Choose the FIELD BUS GATEWAY tab.

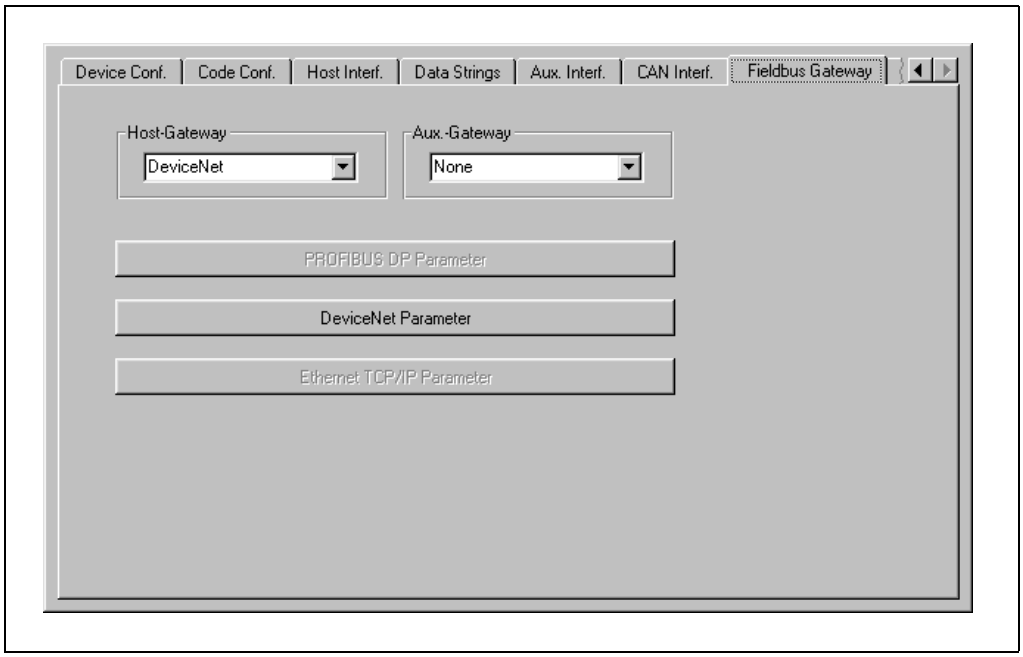


Fig. 4-3: CLV Setup: "Field Bus Gateway" tab

➢ Depending on the planned gateway data connection to the bar code scanner, choose the list entry DEVICENET in the HOST GATEWAY or AUX GATEWAY field.

➢ Click DEVICENET PARAMETERS.
The DEVICENET PARAMETERS dialog box is displayed.

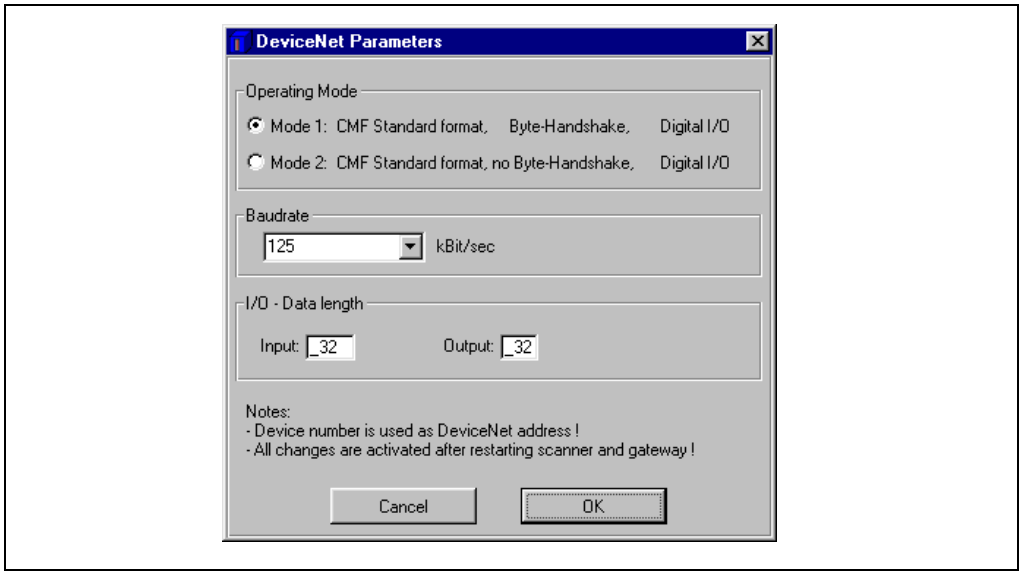


Fig. 4-4: CLV Setup: "DeviceNet Parameters" dialog box

➢ Choose the required gateway transmission method in the OPERATING MODE sector.

For a more detailed description of the individual transmission methods, see *Chapter 5 Communication via DeviceNet, Page 14*.

➢ Set the following values on the tab:

- Target of the result data string: Asynchronous host interface
- Baud rate: **125**, 250, 500 kBd
- Useful data length for incoming: 8 to **32** to 240 Byte
- Useful data length for outgoing: 8 to **32** to 240 Byte

The values in **bold** are the default gateway settings.

4.3.5 Downloading the Parameter Set and Restarting the System

1. Download the new parameter set to the bar code scanner.
   To do so, choose the  icon in the toolbar.
   The PARAMETER DOWNLOAD dialog box is displayed.
2. Confirm the dialog box by choosing the **Permanent** storage option.
3. Save the modified parameter set as a new configuration file in CLV Setup.
4. To restart the bar code scanner **and** gateway, briefly disconnect the connection module from the power supply (e.g. switch S 1 to "OFF" and back again).
   The new settings are activated.

4.3.6 Further Activities

1. If the bar code scanner communicates with the gateway via its terminal interface, set the gateway configuration switch to "Aux".
2. Connect the connection module to DeviceNet (M12 plug on the front panel).
3. Import the EDS file to the field bus master and, if necessary, configure.
4. Test the installation.

4.4 Configuring the Gateway with ComPro

Instead of a bar code scanner, a different device with an RS 232 interface and STX/ETX frame can also be connected to DeviceNet via the gateway. Since CLV Setup cannot be used to configure the (external) device, a special configuration tool called ComPro is required. For configuring, the parameterization cable no. 2030490 is additionally required to the 3-core RS 232 data cable (null modem cable), see *Table 7-2, Page 24*.

In the following cases, configuration must be carried out with ComPro in order to integrate a device in DeviceNet via the gateway in the connection module:

- The SICK bar code scanner firmware does not allow you to configure the gateway.
- A SICK hand scanner is operated on the connection module.
- An external device with an RS 232 interface and STX/ETX frame is operated on the connection module.

Note If the field bus gateway has already been configured using ComPro, it cannot be reconfigured using CLV Setup via the connected bar code scanner.
In this case, the gateway always uses the settings made with ComPro.

➢ To configure the gateway using CLV Setup again, you first have to reset the internal gateway database using ComPro.

Pin Assignment for the 3-Pin Molex Socket (Gateway)

Pin	Signal	Function
1	GND	Signal ground
2	TxD	Transmitter
3	RxD	Receiver

Table 4-1: Pin assignment for the 3-pin Molex socket of the gateway (RS 232)

➢ Connect the PC to the 3-pin Molex socket on the gateway.
To do so, connect the 3-core RS 232 data cable (null modem cable) to the PC. Connect the free D Sub socket of the RS 232 data cable to the D Sub plug of the parameterization cable no. 2030490 (1:1 connection). Connect the parameterization cable to the Molex socket.

5 Communication via DeviceNet

5.1 Data Exchange

The gateway enables data to be exchanged between the connected bar code scanner and a bus master. At the RS 232 interface, the telegrams are provided with a frame by means of the control characters "STX" and "ETX". The frame is not transmitted on the field bus.

The field bus must ensure that the input and output data is transmitted to and from the bus cyclically and consistently. The lengths of the input and output data from the field bus can be defined separately from each other. The permissible value range in each case is 8 to 240 bytes.

The communication procedure between the gateway for DeviceNet and the bus master is defined with 5 control bytes. The three transmission methods available are described below. You can decide which method to use when carrying out configuration with CLV Setup (see *Chapter 4.3.4 Configuring the Field Bus Gateway, Page 11*).

5.2 Standard CMF Format, Byte Handshake, Digital I/O

This transmission method can be easily handled in the field bus master (PLC).

Two structures comprising a header and the actual user data are created for the cyclical input and output data of the field bus. In the header, a handshake uses counters to control transmission of the user data.

Digital Output	InTelegram CountBack	OutTelegram Count	OutTelegram LenRest	OutTelegram
8 bits	8 bits	8 bits	16 bits (INTEL format: LSB/MSB)	N_{out} = 3 to 235 bytes of user data

Table 5-1: Output data (receive telegram from the gateway)

Digital Input	InTelegram Count	OutTelegram CountBack	InTelegram LenRest	InTelegram
8 bits	8 bits	8 bits	16 bits (INTEL format: LSB/MSB)	N_{in} = 3 to 235 bytes of user data

Table 5-2: Input data (send telegram from the gateway)

The fields in the tables above are relevant for the rest of this section and will be described in more detail in the next chapter:

- The *DigitalInput* and *DigitalOutput* bytes are described in *Chapter 5.4 The DigitalInput and DigitalOutput Bytes, Page 19*.
- The *InTelegramCountBack* and *InTelegramCount* counters control user data transmission from the gateway to the field bus master, whereby the gateway is the active node (client) that must activate user data transmission by incrementing *InTelegramCount*. Both counters must have the same status beforehand. The field bus master is the passive node (server).
- The *OutTelegramCount* and *OutTelegramCountBack* counters control user data transmission from the field bus master to the gateway, whereby the field bus master is the active node (client) that must activate user data transmission by incrementing *OutTelegramCount*. Both counters must have the same status beforehand. The gateway is the passive node (server).

- *OutTelegramLenRest* specifies the data length currently remaining to be transmitted from the field bus master to the gateway.
- *InTelegramLenRest* specifies the data length currently remaining to be transmitted from the gateway to the field bus master.
- In *OutTelegram*, the field bus master enters the user data (left justified) to be transmitted to the gateway.
  The length of this field is N_{Out} = parameterized output data length minus 5.
- In *InTelegram*, the gateway enters the user data (left justified) to be transmitted to the field bus master.
  The length of this field is N_{In} = parameterized input data length minus 5.

5.2.1 Direction of Data: Bar Code Scanner to Field Bus Master

Note The following section describes the procedure for transmitting data from the bar code scanner to the field bus master. Data is transmitted from the field bus master to the bar code scanner following a similar procedure.

Transmitting incoming telegrams:

Incoming telegrams are telegrams that the gateway receives from the bar code scanner at the RS 232 interface and then forwards to the field bus master via the input buffers. A distinction is drawn here between two different types of telegram:

- Telegrams that can be transmitted within a single incoming field bus telegram (user data fits in the *InTelegram* element).
- Telegrams that have to be divided into successive blocks due to their length (user data does **not** fit in the *InTelegram* element).

Initialization:

During reset or power-up, that is, before the first incoming telegram is transmitted, the gateway sets the *InTelegramCount*, *OutTelegramCountBack*, and *InTelegramLenRest* to zero. Accordingly, the field bus master must set its *InTelegramCountBack*, *OutTelegramCount*, and *OutTelegramLenRest* to zero.
See also "Resynchronization".

Resynchronization:

If the gateway registers a field bus error (interruption in communication, incorrect counter statuses), it terminates data transmission in the affected data direction(s). The gateway then requests the field bus master to resynchronize transmission. To do so, the gateway sets the corresponding counters (*InTelegramCount* and/or *OutTelegramCountBack*) to zero. The field bus master then has to set its corresponding counters (*InTelegramCountBack* and/or *OutTelegramCount*) to zero too. The initialization status (see *"Initialization"*) is now restored and the active node (client) can repeat or start data transmission.

Counter overflow:

When incremented to 255, the *InTelegramCount* counter is assigned the value "1" (0 → 1 → 2 → to → 254 → 255 → 1 → 2 → to). The value "0" only appears during initialization or if an error occurs (see *"Resynchronization"*).

Transmitting unblocked individual telegrams:

As soon as an incoming telegram that is to be transmitted to the field bus master is available, the gateway checks whether the previous data transmission has been completed (*InTelegramCountBack* is identical to *InTelegramCount*). The incoming telegram is then

entered in the *InTelegram* field (left justified). The bytes in the *InTelegram* field that are not required are set to zero. The length of the incoming telegram is then entered in the *InTelegramLenRest* field. Finally, the *InTelegramCount* counter is incremented as described in *"Counter overflow"*.

The field bus master is always activated when the *InTelegramCount* field is incremented, that is, when the value is different to the *InTelegramCountBack* counter value. The field bus master monitors whether *InTelegramLenRest* is less than or equal to N_{In} (parameterized incoming data length minus 5) and then copies the *InTelegramLenRest* bytes in the *InTelegram* field to a different buffer.

The field bus master then sets its *InTelegramCountBack* counter to the current *InTelegramCount* value. This signals to the gateway that the field bus master is ready to transmit data again.

If the field bus master requires some time to evaluate and process the bar code scanner data, it can delay acknowledging the *InTelegramCountBack* counter accordingly.

Transmitting blocked telegrams:

If the length (N_{In}) of the *InTelegram* field is not sufficient for transmitting an incoming telegram, the incoming telegram is divided into several blocks. These blocks are transmitted in succession. The same fields are used as for the individual telegram.

With the first block, the first part of the incoming telegram is entered in *InTelegram*. The entire length of the incoming telegram is entered in *InTelegramLenRest*. The field bus master responds once *InTelegramCount* has been incremented. It recognizes from the telegram length (*InTelegramLenRest*) entered, which is greater than the *InTelegram* field, that a blocked data transmission is running here. It transmits the entire *InTelegram* field to a buffer and then sets *InTelegramCountBack* to the current value at *InTelegramCount*.

The gateway now recognizes that the field bus master has processed the first part of the incoming telegram. The next part of the incoming telegram is then entered in the *InTelegram* field. *InTelegramLenRest* is assigned a value reduced by the length of the *InTelegram* = N_{In} field. Finally, *InTelegramCount* is incremented again.

The reincremented *InTelegramCount* value reactivates the field bus master, which checks *InTelegramLenRest* again. If this value is now less than or equal to N_{In}, the remaining telegram is copied to the buffer and telegram transmission is complete. If the value is greater than N_{In}, the entire telegram is copied to the buffer and the field bus master waits for the next block.

Example:

The incoming telegram (user data) from the bar code scanner is "123456789".
The field bus master responds to the bar code scanner with "OK" (in this example only; "OK" is not a meaningful telegram for the bar code scanner!).

The input and output data length of the field bus is in each case 10 bytes, which means that N_{In} and N_{Out} both comprise 5 bytes.

The counters are equalized, that is, the previous data transmissions are complete in both directions.

The data bytes are specified in the order in which they appear on the field bus. The length of the incoming telegram (9 bytes), for example, is specified in the *InTelegramLenRest* field as 09h 00h (INTEL format: LSB/MSB!):

1. No new data

Digital Output	InTelegram CountBack	OutTelegram Count	OutTelegram LenRest	OutTelegram
00h	34h	18h	04h 00h	E N D E 00h

Table 5-3: Output data (receive telegram from the gateway)

Digital Input	InTelegram Count	OutTelegram CountBack	InTelegram LenRest	InTelegram
00h	34h	18h	03h 00h	- - - 00h 00h

Table 5-4: Input data (send telegram from the gateway)

2. Incoming telegram, first block (block not yet processed by field bus master)

Digital Output	InTelegram CountBack	OutTelegram Count	OutTelegram LenRest	OutTelegram
00h	34h	18h	04h 00h	E N D E 00h

Table 5-5: Output data (receive telegram from the gateway)

Digital Input	InTelegram Count	OutTelegram CountBack	InTelegram LenRest	InTelegram
00h	35h	18h	09h 00h	1 2 3 4 5

Table 5-6: Input data (send telegram from the gateway)

3. Incoming telegram, first block (block processed by field bus master)

Digital Output	InTelegram CountBack	OutTelegram Count	OutTelegram LenRest	OutTelegram
00h	35h	18h	04h 00h	E N D E 00h

Table 5-7: Output data (receive telegram from the gateway)

Digital Input	InTelegram Count	OutTelegram CountBack	InTelegram LenRest	InTelegram
00h	35h	18h	09h 00h	1 2 3 4 5

Table 5-8: Input data (send telegram from the gateway)

4. Incoming telegram, second and final block (block not yet processed by field bus master)

Digital Output	InTelegram CountBack	OutTelegram Count	OutTelegram LenRest	OutTelegram
00h	35h	18h	04h 00h	E N D E 00h

Table 5-9: Output data (receive telegram from the gateway)

Digital Input	InTelegram Count	OutTelegram CountBack	InTelegram LenRest	InTelegram
00h	36h	18h	04h 00h	6 7 8 9 00h

Table 5-10: Input data (send telegram from the gateway)

5. Incoming telegram, second and final block (block processed by field bus master)

The field bus master simultaneously transmits the outgoing telegram "OK" to the bar code scanner; the gateway has not yet processed the outgoing telegram.

Digital Output	InTelegram CountBack	OutTelegram Count	OutTelegram LenRest	OutTelegram
00h	36h	19h	02h 00h	O K 00h 00h 00h

Table 5-11: Output data (receive telegram from the gateway)

Digital Input	InTelegram Count	OutTelegram CountBack	InTelegram LenRest	InTelegram
00h	36h	18h	04h 00h	6 7 8 9 00h

Table 5-12: Input data (send telegram from the gateway)

6. Incoming telegram "OK" to the bar code scanner (the gateway has processed the outgoing telegram)

Digital Output	InTelegram CountBack	OutTelegram Count	OutTelegram LenRest	OutTelegram
00h	36h	19h	02h 00h	O K 00h 00h 00h

Table 5-13: Output data (receive telegram from the gateway)

Digital Input	InTelegram Count	OutTelegram CountBack	InTelegram LenRest	InTelegram
00h	36h	19h	04h 00h	6 7 8 9 00h

Table 5-14: Input data (send telegram from the gateway)

Both data transmissions are complete. All the counters have been equalized.

Note on 5. and 6.:
The field bus master does not have to set the unused bytes in the *OutTelegram* field to zero, as in this example. These bytes have no effect on the gateway.

5.3 Standard CMF Format, No Byte Handshake, Digital I/O

This transmission method is intended for **test purposes during commissioning**. It does **not necessarily ensure reliable data transmission** between the gateway and field bus master because incoming telegrams may be lost due to the missing acknowledgement!

This transmission method is identical to that described in *Chapter 5.4 The DigitalInput and DigitalOutput Bytes, Page 19*, with the following differences:

- It does not support **blocking**.
  For this reason, incoming telegrams that are too long are truncated, and outgoing telegrams that are too long are not sent to the bar code scanner (*see below*).
- The gateway increments/decrements the *InTelegramCount* counter, but the field bus master does not have to increment/decrement the *InTelegramCountBack* counter. This means that the field bus master (PLC) does not have to carry out any work for incoming telegrams.

For outgoing telegrams, the field bus master must continue incrementing/decrementing *OutTelegramCount*, *OutTelegramLenRest*, and *OutTelegram*.

Digital Output	InTelegram CountBack	OutTelegram Count	OutTelegram LenRest	OutTelegram
8 bits	8 bits	8 bits	16 bits (INTEL format: LSB/MSB)	N_{out} = 3 to 235 bytes of user data

Table 5-15: Output data (receive telegram from the gateway)

Digital Input	InTelegram Count	OutTelegram CountBack	InTelegram LenRest	InTelegram
8 bits	8 bits	8 bits	16 bits (INTEL format: LSB/MSB)	N_{in} = 3 to 235 bytes of user data

Table 5-16: Input data (send telegram from the gateway)

If the user data in the incoming telegram from the bar code scanner is longer than N_{In}, the data is truncated. The field bus master recognizes this by the fact that *InTelegramLenRest* is greater than N_{In}. If the field bus master sets *OutTelegramLenRest* so that it is greater than N_{Out} and activates data transmission, the gateway does not transmit data, as it would otherwise send a truncated telegram to the bar code scanner!

5.4 The DigitalInput and DigitalOutput Bytes

The first byte in the header for transmission method "001 (standard)" and "002 (standard no ack)" contains the status of the switching inputs and outputs. The *DigitalInput* byte also contains status bits and a heartbeat bit.

DigitalInput is assigned as follows:

Bit	Name	Meaning
D7	*SccError*	The interface controller has reported an SCC error (parity, framing, etc.).
D6	*BufferOverrun*	All the output buffers in the gateway are full. The gateway does not accept any transmitted data when this error is present. The field bus master can repeat data transmission later stage and then, if necessary, report an error to the user.
D5	*NackScanner*	The bar code scanner has again rejected a user data telegram with "NAK".
D4	*TimeoutScanner*	The bar code scanner has not responded to a user data telegram within the 50 ms monitoring time.
D3	*PlcError*	The gateway has detected a handling error in the field bus master (PLC). The gateway does not accept any transmitted data when this error is present and requests resynchronization with the field bus master. The PLC program must be corrected in line with the error.
D2	*Heartbeat*	The gateway makes this heartbeat bit available to the field bus master. The heartbeat bit toggles between 0 and 1 at 1-second intervals.
D1	*In2*	Status of switching input IN 2 (terminal 3)*): Voltage off: *In2* = 0 Voltage on: *In2* = 1 The input is debounced by the gateway with 20 ms.
D0	*In1*	Status of switching input IN 1 (terminal 2)*): Voltage off: *In1* = 0 Voltage on: *In1* = 1 The input is debounced by the gateway with 20 ms.
*) Terminals on the 5-pin gateway terminal strip		

Table 5-17: "DigitalInput" byte assignment

Important:

The gateway sets status bits (error bits) D7 to D3 when it acknowledges data transmission from the field bus master with *OutTelegramCountBack*, that is *OutTelegramCount* and *OutTelegramCountBack* are the same again. Status bits D7 to D3 are reset by the gateway

when the field bus master activates the next data transmission with *OutTelegramCount*. The field bus master, therefore, must always evaluate the status bits before transmitting the next batch of data to the gateway.

The output buffer in the gateway does **not** allow the status bits (D7, D5, and D4) to be assigned to a specific data transmission procedure initiated by the field bus master!

DigitalOutput is assigned as follows:

Bit	Name	Meaning
D7	-	Reserved
D6	-	Reserved
D5	-	Reserved
D4	-	Reserved
D3	-	Reserved
D2	-	Reserved
D1	*Out2*	Status of switching output OUT 2 (terminal 5)*): *Out2* = 0: Voltage off *Out2* = 1: Voltage on
D0	*Out1*	Status of switching output OUT 1 (terminal 4)*): *Out1* = 0: Voltage off *Out1* = 1: Voltage on
*) Terminals on the 5-pin gateway terminal strip		

Table 5-18: "DigitalOutput" byte assignment

6 Troubleshooting

If an error occurs during data transmission, the gateway indicates this via its three LEDs.

6.1 System LED: "SYS" (V3)

The duo LED "SYS" (SYS = system) comprises the individual LEDs "RDY" (RDY = ready) and "RUN". The "RDY" LED signals whether or not the gateway is ready for operation. while the "RUN" LED signals whether or not the gateway is using its communication functions.
The following applies:

Function	Color	Status	Description
RUN	*Green*	On	Communication active
		Irregular flashing	Parameterization error
		Regular flashing	Ready to communicate[1)]
		Off	No communication
RDY	*Yellow*	On	Gateway ready
		Cyclic flashing	Bootstrap loader active
		Irregular flashing	Hardware or system error
		Off	Hardware defect
1) When the gateway is waiting to establish a connection with the field bus master.			

Table 6-1: Function of the "SYS" LED (V3) on the gateway

6.2 DeviceNet: LED "COM" (V2)

The duo LED "COM" (COM = communication) indicates the status of DeviceNet.
The following applies:

Function	Color	Status	Description
MNS	*Green*	On	Gateway connected to the bus, communication establised
		Flashing	Gateway connected to the bus, no communication
	Red	On	critical link error, critical error
		Flashing	subordinated error and/or time monitoring error
	Red/ Green	Flashing	Communication aborted
	-	Off	No power supply, not connected to the bus

Table 6-2: Function of the "COM" LED (V2) on the gateway (slave)

6.3 Bar Code Scanner LED: "SER" (V1)

The duo LED "SER" (SER = serial), which comprises the green LED "SD" (serial data LED) and the yellow LED "XOFF", is available for the bar code scanner.

Function	Color	Status	Description
SD	*Green*	On	Communication with bar code scanner OK
		Off	No communication with bar code scanner for 200 ms

Table 6-3: Function of the "SER" LED (V1) on the gateway

Function	Color	Status	Description
XOFF	*Yellow*	On	Gateway has sent XOFF. LED extinguishes when XON is sent.
		Off	Initial state (OK)
		Regular flashing	Gateway reception or transmission buffer has overflowed. Flashing terminated when a receive or transmit buffer is released. Note: The "Regular flashing" status has priority over the "On" status.

Table 6-3: Function of the "SER" LED (V1) on the gateway (contd.)

Notes:

The "SD" LED is operated as follows:

1. Once a telegram has been successfully transmitted from the bar code scanner to the gateway, the LED lights up in the following cases:
   - The gateway is in the *Init state* OPERATING or is waiting for this status.
   - The telegram contains user data.
   - The user data telegram can be copied to an input buffer.
2. The LED also lights up when the gateway has successfully transmitted a user data telegram to the bar code scanner, that is, an "ACK" has been received from the bar code scanner (in *SER Handshake* = ACK/NAK mode only).

The LED extinguishes if conditions 1 and 2 do not occur again after more than 200 ms.

7 Technical Data

7.1 Data Sheet: Fieldbus Gateway CMF400 for DeviceNet

Type	CMF400-2101
Order no.	1026242
Supported bar code scanners	CLV42x From firmware version V 1.60 CLV43x to 45x From firmware version V 3.50 ICR84x From firmware version V 2.01 ICR85x From firmware version V 1.70 CLV480, CLV/X490 From firmware version V 3.50
Basic devices (connection modules)	CDM420-0001 (no. 1025362) CDM490-0001 (no. 1025363)
Bar code scanner data interface	RS 232
Data transmission rate	9,600 to 57,600 Bd
Data format	1 start bit, 8 data bits, 1 stop bit, no/odd parity
Protocol start character	STX (02 hex), transmit and receive
Protocol stop character	ETX (03 hex), transmit and receive
DeviceNet interface	Electrically isolated
Data transmission rate	125, 250, 500 kbit/s
Station type	Group 2 Only Slave
No. of receive buffers	6 for data from the bar code scanner
No. of transmit buffers	3 for data from the bus master
Telegram length	8 to 240 bytes, max. 512 bytes (with blocking)
Switching inputs	2 x digital
Switching outputs	2 x digital, cable length max. 30 m (98.4 ft)
Registration with master	With EDS file ("SICK0808.EDS")
Indicators	3 x LED (for indicating statuses and malfunctions/errors)
Configuration	1 x slide switch Via CLV Setup configuration software Via ComPro configuration software (alternative)
Electrical connections	1 x 26-pin SMD connector to the basic device 1 x 5-pin M12 plug for DeviceNet on front panel 1 x 5-pin terminal strip for digital inputs/outputs
Power supply	18 to 30 V DC via the basic device
Power consumption	2 W
Enclosure rating	IP 65 (to DIN 40 050), built-in in the basic devices
EMC tested	According to EN 55011, Class B/EN 61000-6-3
Temperature (operating/storage)	0 °C to +40 °C / –20 °C to +70 °C (+32 °F to +104 °F/–4 °F to +158 °F)
Max. rel. humidity	90 %, without condensation
Attachment	With 2 threaded pins on spacer washer

Table 7-1: Technical data for the CMF400 for DeviceNet

7.2 Accessories

Part No.	Description	Wires	Length	Connection
2030490	Parameterization cable for connecting to the CMF400-2101, with 9-pin D Sub plug and 3-pin Molex plug. Required if the ComPro configuration software is used.	3 x 0.14 mm^2 (26 AWG)	0.25 m (0.82 ft)	PC to CMF400
2014054	RS 232 data cable (SICK null modem cable), dia. 5 mm, shielded, with two 9-pin D Sub sockets. Pin2 (RxD) and Pin 3 (TxD) crossed.	3 x 0.34 mm^2 (22 AWG)	3 m (9.84 ft)	PC to CDM420/490

Table 7-2: Available accessories: cables

8 Appendix

8.1 EC Declaration of Conformity

Fig. 8-1 shows the scaled down copy of the EC Declaration of Conformity (page 1).

➢ Complete copy of EC Declaration of Conformity on request.

SICK

EC Declaration of conformity

Ident-No. : 9085578 O639

The undersigned, representing the following manufacturer

SICK AG
Nimburger Straße 11
79276 Reute
Deutschland

herewith declares that the product

CMF400

is in conformity with the provisions of the following EC directive(s) (including all applicable amendments), and that the standards and/or technical specifications referenced in page 2 have been applied.

(place),..Reute..... (date),..2004-07-05

ppa. Pierenkemper
(Manager Development
Division Auto Ident)

ppa. Walter
(Manager Production
Division Auto Ident)

Fig. 8-1: Copy of the Declaration of Conformity (Page 1, scaled down)

Australia
Phone +61 3 9497 4100
1800 33 48 02 – 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 5091-4900
E-Mail sac@sick.com.br

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 (0)2 11 53 01-270
E-Mail info@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 +91–22–2822 7084
E-Mail info@sick-india.com

Italia
Phone +39 02 27 40 93 19
E-Mail info@sick.it

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

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

Norge
Phone +47 67 81 50 00
E-Mail austefjord@sick.no

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

Republic of Korea
Phone +82-2 786 6321/4
E-Mail kang@sickkorea.net

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

Russia
Phone +7 95 775 05 30
E-Mail info@sick-automation.ru

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

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

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

Sverige
Phone +46 8 680 64 50
E-Mail info@sick.se

Taiwan
Phone +886 2 2365-6292
E-Mail sickgrc@ms6.hinet.net

Türkiye
Phone +90 216 388 95 90 pbx
E-Mail info@sick.com.tr

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

More representatives and agencies
in all major industrial nations at
www.sick.com

SICK
Sensor Intelligence.