Show/Hide Bookmarks

EDSDEN
00471571

Communication Manual

DeviceNet

Lenze

Show/Hide Bookmarks

1 Preface

Contents

Show/Hide Bookmarks

1.1 Introduction

The competitive situation in the mechanical and system engineering sector requires new means to optimise the production costs. This is why modular machine and system engineering is becoming increasingly more important, since individual solutions can now be set up easily and cost-effectively from a single modular system.

Lenze fieldbus systems in industrial applications

For an optimal communication between the single modules of a system, fieldbus systems are increasingly used for process automation. Lenze offers the following communication modules for the standard fieldbus systems:

- CAN (Lenze system bus)
- CANopen
- PROFIBUS-DP
- Interbus
- INTERBUS loop
- DeviceNet
- LON
- AS-i

The communication modules are especially designed for Lenze drive components and flexible use. You can use the same communication modules for Lenze servo inverters and Lenze frequency inverters.

This means for you: Easy communication.

- You must only know one communication system.
- Handling is always the same.
- You reduce your costs because you can make use of the knowledge gained once.
  - Training is only required once.
  - The planning time becomes shorter.

Decision support

The decision for a fieldbus systems depends on many different factors. The following overviews will help you to find the solution for your application.

PROFIBUS-DP

For bigger machines with bus lengths of more than 100 metres, INTERBUS or PROFIBUS-DP (PROFIBUS-**D**ecentralised **P**eriphery) are frequently used. The PROFIBUS-DP is always used together with a master control (PLC) – here the PROFIBUS master transmits e. g. the setpoints to the single PROFIBUS devices (e. g. Lenze controller).

When using the data baud rate of 1.5 Mbits/s typical for the PROFIBUS-DP, the sensors and actuators receive the process data. Due to the data transmission mode and the telegram overhead, a bus cycle time results at 1.5 Mbits/s, which is sufficient to control e. g. conveyors. If, for technical reasons, the process data must be transmitted faster to the sensors and actuators, the PROFIBUS can also be operated with a data transmission rate of maximally 12 Mbits/s.

Interbus

The INTERBUS proves its worth primarily in commercial sized installation with many nodes, for example in the automotive industry. The ring structure offers excellent diagnostic functions. It can be exactly detected, which node telegram is destroyed by electromagnetic interferences or if the INTERBUS cable is affected by a short circuit or earth fault. Moreover, the INTERBUS with a baud rate of 500 kbits/s is more efficient in the process data transmission than other bus systems with the same baud rate. If data is to be transmitted extremely fast, the INTERBUS can also be operated with 2 Mbits/s.

Lenze system bus (CAN)

With the 9300 servo controller series, Lenze has implemented the system bus based on CAN (**C**ontroller **A**rea **N**etwork). In this connection, functions of the CANopen communication profile according to DS301 were integrated. The main task of the system bus is to exchange data between the controllers and to easily communicate with sensors, actuators and operating/display units without the need of a master control. Furthermore, you can realise applications in which controllers are synchronised to each other by means of the system bus.

The CAN bus is available at a reasonable price and is suitable for smaller machines.

CANopen

CANopen is a communication protocol specified to the CiA (**C**AN **i**n **A**utomation) user group. Lenze can provide communication modules for communicating with CANopen masters. These modules are compatible with the specification DS 301 V4.01.

DeviceNet

The American automation specialist Allan Bradley developed the DeviceNet fieldbus based on the CAN controller. This communication profile is published by the ODVA user organisation. A great amount of sensors and actuators are available. Similar to CANopen, a DeviceNet master is used to control the DeviceNet.

LON

The Echelon company (USA) has developed the **L**ocal **O**peration **N**etwork for distributed industrial applications with non-time critical demands. This bus system is mainly used in building automation. Each device has an own intelligence so that a master control system is only required in a restricted way or not at all.

Show/Hide Bookmarks

AS-i

The AS-i (**A**ctuator-**S**ensor-**I**nterface) bus is frequently used at the lowest sensor and actuator level. Binary I/O signals can be transmitted very cost-effectively this way. The bus system is user-friendly, easy to configure and flexible in terms of installation. The data and the auxiliary power for the connected AS-i devices are transmitted via the two-core AS-i cable.

INTERBUS loop

The INTERBUS Loop was developed as a pure sensor/actuator bus and provides a quick and easy means of connecting digital and analog devices via insulation piercing connecting devices. INTERBUS Loop is located lower down the hierarchy than INTERBUS (remote bus) and can be easily connected to it via a bus terminal.

1.2 Comparison of industrial fieldbus systems

	CAN / CANOpen	DeviceNet	PROFIBUS-DP	AS-i	Interbus	INTERBUS loop	LON
Topology	Line with terminating resistors	Line with terminating resistors	Line with terminating resistors	Line, tree, ring (possible)	Ring	Ring	Line (2 wire) or any other
Bus management	Multi master	Single master	Single master	Single master	Single master	only together with INTERBUS-S; single master (bus terminal)	Multi master
Max. number of devices (master and slaves)	64	64	124 (4 segments, 3 repeaters), max. 32 per segment	124 sensors/actors 1 master	512 slaves, 1 master	32 slaves	32385 devices distributed to 255 subnetworks with 127 devices each
Max. distance between devices without repeater	Dependent on the baud rate 1 km (50 kbits/s) 25 m (1 Mbits/s)	100 m (500 kbits/s) 250 m (250 kbits/s) 500 m (125 kbits/s)	1.2 km (93.75 kbits/s) 100 m (12 Mbits/s)	100 m	1.5 m (local bus) 400 m (remote bus) 2.5 km (optical fibre)	10 m (max. 100 m cable length without repeater)	2 km at 78 kbits/s (twisted pair), 6.1 km at 5.48 kbits/s (optical fibre plastics)
Max. distance between devices with repeater	General length reduction Dependent on the repeater used	Not specified	10 km (93.75 kbits/s)	300 m (2 repeaters)	13 km (remote bus), 100 km (optical fibre)	No repeater required	Almost any Expandable by subnetworks (without repeaters)
Transfer medium	Shielded, twisted pair cable	Shielded, twisted pair cable	Shielded, twisted pair cable	Unshielded and untwisted flat pair cable	Shielded, twisted 5-wire cable Optical fibre, infrared	Unshielded, twisted pair cable	Unshielded and untwisted pair cable Radio, optical fibre, power line
Auxiliary energy supply via bus cable	Possible via additional wires in the bus cable	Possible via additional wires in the bus cable	Possible via additional wires in the bus cable	Current supply via data cable (2 to 8 A)	Group via bus terminal (remote bus)	Current supply via data cable (approx. 1.5 A)	Possible via additional wires in the bus cable
Baud rate	10 kbits/s - 1 Mbit/s	125 kbits/s, 250 kbits/s, 500 kbits/s	9.6 kbits/s - 12 Mbits/s	167 kbits/s	500 kbits/s oder 2 Mbits/s	500 kbits/s	78 kbits/s - 1.25 Mbits/s
Typical update time (e.g. 8 devices, 4 bytes user data)	Approx. 1.32 ms at 1 Mbit/s (high priority)	Approx. 2.64 ms at 500 kbits/s (high priority)	Approx. 2.5 ms at 500 kbits/s	Typically 5 ms (every 4 bits)	At least 2 ms (process data)	At least 2 ms (process data)	Approx. 70 ms
Telegram length (user data)	0 to 8 bytes	0 to 8 bytes	0 to 246 bytes	4 bits	1 to 64 bytes data; up to 246 bytes parameters	1 to 64 bytes data; up to 246 bytes parameters	1 to 228 bytes data; typically approx. 11 bytes
Telegram length (total)	106 bits at 8 bytes user data	106 bits at 8 bytes user data	User data + 6 to 11 bytes	21 bits, of which: 14 bits master, 7 bits slave	User data + 6 bytes	User data + 6 bytes	Max. 255 bytes, User data + 27 bytes
Bus access methods	CSMA/CA message oriented	CSMA/CA message oriented	Cyclic polling	Cyclic polling	Time grid / distributed shift register	Time grid / distributed shift register	Modified CSMA/CD

	CAN / CANOpen	DeviceNet	PROFIBUS-DP	AS-i	Interbus	INTERBUS loop	LON
Lenze communication modules for Lenze basic devices							
• 9300 Servo Inverter and Servo PLC	on board (only parts of CANopen) CANopen 2175 (pluggable	2175 (pluggable)	2133 (pluggable)	Not available	2111 and 2113 (both pluggable)	2112 (pluggable)	2141 (pluggable)
• 8200 vector frequency inverter	Function module System bus (only parts of CANopen) E82ZAFCC010 E82ZAFCC100 or E82ZAFCC210 or pluggable 2175 (CANopen) 2171, 2172 (parts of CANopen)	Function module (in preparation) Pluggable 2175	Function module E82ZAFPC010 or 2133 (pluggable)	E82ZAFFC010 function module	Function module E82ZAFIC010 (can be integrated) or 2111 or 2113 (both pluggable)	2112 (pluggable)	2141 (pluggable)
• Frequency inverter 8200 motec	Function module System bus (only parts of CANopen) E82ZAFCC001	(In preparation)	Function module E82ZAFPC001	Function module E82ZAFFC001	Function module E82ZAFIC001 (can be integrated)	-	-
• Drive PLC	Function module System bus (only parts of CANopen) E82ZAFCC010 or 2175 (pluggable)	2175 (pluggable)	2133 (pluggable)	-	2111 and 2113 (both pluggable)	2112 (pluggable)	2141 (pluggable)
• starttec	Function module System bus (only parts of CANopen) E82ZAFCC001	(In preparation)	Function module E82ZAFPC001	Can be integrated into the basic device as variant	Function module E82ZAFIC001 (can be integrated)	-	-

1.3 About this Communication Manual

Target group

This Manual is intended for all persons who plan, install, commission, and maintain a network for a machine.

Contents

This Manual only describes Lenze communication modules of a bus system.

The Manual completes the Mounting Instructions coming with the device.

- The features and functions of the communication modules are described in detail.
- Typical applications are shown by examples.
- It also contains
  - safety instructions which must be observed at any means.
  - the essential technical data of the communication module.
  - information about versions of the basic Lenze devices to be used. Basic devices include servo inverters, frequency inverters, Drive PLC and (starttec) motor starters.
  - notes on troubleshooting and fault elimination.

This Manual does not describe software of a third party manufacturer. Lenze does not take responsibility for corresponding data given in this Manual. Information on how to use the software can be obtained from the documentation of the master system.

The theoretical background is only explained if absolutely necessary to understand a function of the corresponding communication module.

All brand names used in this Manual are trademarks of their respective owners.

How to find information

Every chapter is about a certain topic and gives you all necessary information.

- The table of contents and the index help you to find information on a certain topic.
- Descriptions and data of the Lenze products (controllers, Drive PLC, Lenze geared motors, Lenze motors) are included in the corresponding catalogues, Operating Instructions, and Manuals. You can either order the documentation required from your Lenze representative or download it from the Internet as a PDF file.

Paper or PDF

The Manual is designed as a loose-leaf collection so that we are able to inform you quickly and specifically about news and changes of our communication modules. Each page is marked by a publication date and a version number.

Tip!

Current documentations and software updates for Lenze products can be found on the Internet in the ”Downloads” area under

http://www.Lenze.com

1.4 Legal regulations

Labelling

Lenze communication modules and Lenze function modules are unambiguously identified by their nameplates.

Manufacturer

Lenze Drive Systems GmbH, Postfach 101352, D-31763 Hameln

CE conformity

Conforms to the EC Low Voltage Directive

Application as directed

The communication module and the function module

- must only be operated as described in this Communication Manual and under the conditions stated.
- are accessory modules which are used for Lenze controllers and Drive PLCs as an option. More details can be found in chapter ”General information”.
- must be connected and mounted in a way that it fulfils its function without being a hazard for persons.

Observe all notes given in the chapter ”Safety information”.

Please observe all notes and information on the corresponding communication module and function module given in this Communication Manual.This means:

- Read this part of the Communication Manual carefully before you start working on the system.
- This Communication Manual must always be available while the communication module or function module is in operation.

Any other use shall be deemed as inappropriate!

Liability

The information, data, and notes given in the Communication Manual met the state of the art at the time of printing. Claims on modifications referring to communication modules or function modules which have already been supplied cannot be derived from the information, illustrations, and descriptions.

The specifications, processes, and circuitry in this Communication Manual are for guidance only and must be adapted to your own specific application. Lenze does not take responsibility for the suitability of the process and circuit proposals.

The indications given in this Communication Manual describe the features of the product without warranting them.

Lenze does not accept any liability for damage and operating interference caused by:

- Disregarding the Communication Manual
- Unauthorised modifications to the communication module/function module
- Operating faults
- Improper working on and with the communication module/function module

Warranty

See Sales and Delivery Conditions of Lenze Drive Systems GmbH.

Warranty claims must be made immediately after detecting defects or faults.

The warranty is void in all cases where liability claims cannot be made.

Waste disposal

Material	recycle	dispose
Metal	●	-
Plastic	●	-
Assembled PCBs	-	●
Short Instructions/Operating Instructions	●	-

2 Safety instructions

Contents

2.1 Persons responsible for safety

Operator

An operator is any natural or legal person who uses the drive system or on behalf of whom the drive system is used.

The operator or his safety personnel is obliged

- to ensure the compliance with all relevant regulations, instructions and legislation.
- to ensure that only qualified personnel works on and with the drive system.
- to ensure that the personnel has the Operating Instructions available for all work.
- to ensure that all unqualified personnel are prohibited from working on and with the drive system.

Qualified personnel

Qualified personnel are persons who - due to their education, experience, instructions, and knowledge about relevant standards and regulations, rules for the prevention of accidents, and operating conditions - are authorised by the person responsible for the safety of the plant to perform the required actions and who are able to recognise potential hazards. (Definition for skilled personnel to VDE 105 or IEC 364)

2.2 General safety instructions

- These safety instructions do not claim to be complete. If you have any questions or problems please contact your Lenze representative.
- The communication module meets the state of the art at the time of delivery and generally ensures safe operation.
- The data in this manual refer to the stated hardware and software versions of the communication modules.
- The communication module may create a hazard for personnel, for the equipment itself or for other property of the operator, if:
  - non-qualified personnel work on and with the communication module.
  - the communication module is used improperly.
- The specifications, processes, and circuitry described in this Manual are for guidance only and must be adapted to your own specific application.
- Provide appropriate measures to prevent injury to persons or damage to material assets.
- The drive system must only be operated when it is in perfect condition.
- Retrofitting or changes of the communication module are generally prohibited. In any case, Lenze must be contacted.
- The communication module is a device intended for use in industrial power systems. During operation, the communication module must be firmly connected to the corresponding controllers. In addition, all measures described in the Manual of the controller used must be taken. Example: Mounting of covers to ensure protection against accidental contact.

2.3 Definition of notes used

The following signal words and symbols are used in this documentation to indicate dangers and important information:

Safety instructions

Structure of safety instructions:

Pictograph and signal word!

(characterises the type and severity of danger)

Note

(describes the danger and gives information about how to prevent dangerous situations)

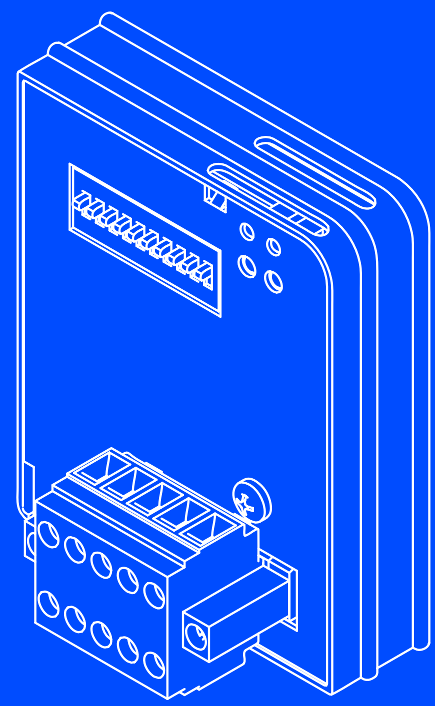
Pictograph and signal word	Meaning
 **Danger!**	**Danger of personal injury through dangerous electrical voltage.** Reference to an imminent danger that may result in death or serious personal injury if the corresponding measures are not taken.
 **Danger!**	**Danger of personal injury through a general source of danger** Reference to an imminent danger that may result in death or serious personal injury if the corresponding measures are not taken.
 **Stop!**	**Danger of property damage.** Reference to a possible danger that may result in property damage if the corresponding measures are not taken.

Application notes

Pictograph and signal word	Meaning
 **Note!**	Important note to ensure trouble-free operation
 **Tip!**	Useful tip for simple handling
	Reference to another documentation

Communication Manual

DeviceNet



EMF2179IB

Communication module

Lenze

5 EMF2179IB communication module

Contents

5.1 Before you start

Tip!

Current documentation and software updates concerning Lenze products can be found on the Internet in the ”Services & Downloads” area under

http://www.Lenze.com

5.1.1 Your opinion is important to us

These instructions were created to the best of our knowledge and belief to give you the best possible support for handling our product.

If you have suggestions for improvement, please e-mail us to:

feedback-docu@Lenze.de

Thank you for your support.

Your Lenze documentation team

5.1.2 Document history

Edition date	Altered chapters	Notes
08 /2006	-	First edition

5.2 General information

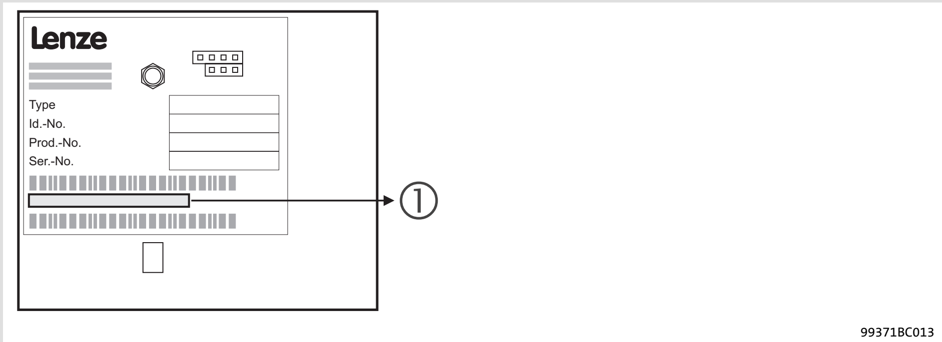
Validity

These instructions are valid for

- ► EMF2179IB communication modules (DeviceNet) as of version 1A.20.

These instructions are only valid together with the Operating Instructions for the standard devices permitted for the application.

Identification



① → 33.2179IB | 1A | 20

Series (33.2179IB)
Hardware version (1A)
Software version (20)

Application range

The communication module can be used together with basic devices as of the following nameplate data:

Type	Design	Version HW	Version SW	Variant	Explanation
33.820X	E./C.	2x.	1x.	Vxxx	8201 - 8204
33.821X	E./C.	2x.	2x.	Vxxx	8211 - 8218
33.822X	E.	1x.	1x.	Vxxx	8221 - 8227
33.824X	E./C.	1x.	1x.	Vxxx	8241 - 8246
82EVxxxxxBxxxXX		Vx	1x		8200 vector
82CVxxxxxBxxxXX		Vx	1x		8200 vector, cold plate
82DVxxxKxBxxxXX		Vx	1x		8200 vector, thermally separated
EPL 10200	E	1x	1x		Drive PLC
33.93XX	xE.	2X	1x	Vxxx	9321 - 9332
33.938X	xE.	1x	0X		9381 - 9383
33.93XX	xC.	2X	1x	Vxxx	9321 - 9332, cold plate
33.93XX	EI / ET	2X	1x	Vxxx	9300 Servo PLC
33.93XX	CI / CT	2X	1x	Vxxx	9300 Servo PLC, cold plate
ECSxPxxxx4xxxxXX		1A	6.0		ECS, Posi & Shaft
ECSxSxxxx4xxxxXX		1A	6.0		ECS, Speed & Torque
ECSxAxxxx4xxxxXX		1A	2.3		ECS, Application

Features

The internationally standardised CAN bus in particular stands out because of

- relatively short transmission times
- low connection costs

These features have also lead to a wide spread of CAN products in other industrial sectors.

DeviceNet

- is based on the CAN technology.
- allows communication between control systems as well as between simple industrial devices like sensors (e.g. initiators) and actuators (e.g. electromagnetically operated pneumatic valve), or also frequency inverters or servo inverters.

The EMF2179IB communication module

- is an 'ONLY-SERVER module of group 2.
- is an attachable additional module for Lenze standard devices.
- offers a simple connection possibility by pluggable terminal strips with double screw connection, 5-pole.
- optionally up to 12 process data words (depending on the standard device)
- access to all Lenze parameters

The front panel DIP switch provides a comfortable setting

- of the baud rate for DeviceNet (125 kBit/s, 250 kBit/s and 500 kBit/s)
- of the node address (max. 63 nodes)
- of the software compatibility to the EMF2175IB communication module

5.3 Technical data

5.3.1 General data and operating conditions

Communication-relevant data	Values
Communication media	DIN ISO 11898
Network topology	Line terminated at both ends (R = 120 Ohm)
Number of nodes	Max. 63
Cable length	Max. 500 m (depending on the baud rate)
Communication profile	DeviceNet

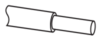
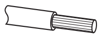
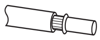
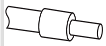
General electric data	Values
Voltage supply (internal / external)	See 🕮 5.4-11

Operating conditions	Values	Deviations from standard
Climatic conditions		
Storage	1 K3 to IEC/EN 60721-3-1	-25 °C ... + 60 °C
Transport	2 K3 to IEC/EN 60721-3-2	
Operation	3 K3 to IEC/EN 60721-3-3	0 °C ... + 55 °C
Enclosure	IP20	
Degree of pollution	2 to IEC/EN 61800-5-1	

5.3.2 Protective insulation

Insulation between bus and ...	**Type of insulation** (to EN 61800-5-1)
• Remote earth / PE	Functional insulation
• External supply	No insulation
• Power stage	
– 820X / 821X	Basic insulation
– 822X / 8200 vector	Reinforced insulation
– 93XX / 9300 servo PLC	Reinforced insulation
– ECS (axis module)	Reinforced insulation
• Control terminals	
– 820X / 821X / 8200 vector	Functional insulation
– 822X	Basic insulation
– 93XX / 9300 servo PLC	Basic insulation
– ECS (axis module)	Basic insulation

5.3.3 Terminal data

Electrical connection	**Plug connector with double screw connection**	
Possible connections		rigid: 1.5 mm² (AWG 16)
		flexible:
		without wire end ferrule 1.5 mm² (AWG 16)
		with wire end ferrule, without plastic sleeve 1.5 mm² (AWG 16)
		with wire end ferrule, with plastic sleeve 1.5 mm² (AWG 16)
Tightening torque	0.5 ... 0.6 Nm (4.4 ... 5.3 lb-in)	
Bare end	6 mm	

5.3.4 Protocol data

Field	**Values**
Max. number of nodes	63
Process data words (16 bit)	1 word ... 12 words
Supported services	Reset, Get_Attribute_Single, Set_Attribute_Single, Allocate_Master/Slave_Connection_Set, Release_Group_2_Identifier_Set

5.3.5 Communication times

Note!

The communication times in the DeviceNet depend on the
- Signal propagation delay of the EMF 2179IBcommunication module
- Processing time in the controller
- Telegram run time on the bus cable
  - Baud rate
  - Telegram length
  - Inter scan delay - time of the scanner
  - Number of the DeviceNet - nodes

820X processing times

In contrast to the parallel process data processing for the 821X / 822X / 824X / 8200 vector device series, the process and parameter data in the 820X device series are processed successively. Therefore the response times to the process data depend on the preceding actions.

Furthermore, the processing times of the individual telegrams depend on the conditioning of the actual values (process data from the drive). If this data (status word, actual frequency value) is not required, it can be deactivated by means of the "bit 15" status word (PE inhibit).

The individual telegram run times are as follows:

Parameter data	PE inhibit = 0	PE inhibit = 1
Processing time	62...140 ms	62...70 ms

Process data	PE inhibit = 0	PE inhibit = 1
Processing time with regard to a change of one value to the drive	27...105 ms	27...35 ms
Processing time with regard to a change of two values to the drive	62...140 ms	4...70 ms
Processing times for process data from the drive	108...140 ms	Not possible

Processing times 821X/822X/824X/8200 vector

Parameter data	**Process data**
30 ... 50 ms	3 ... 5 ms The processing times for the process data refer to the sync telegram.

93XXprocessing times

There are no interconnections between parameter data and process data.

Parameter data	Process data
Approx. 30 ms + 20 ms of tolerance (typical) For some codes, the processing time can be longer (see 9300 System Manual).	3 ms + 2 ms of tolerance

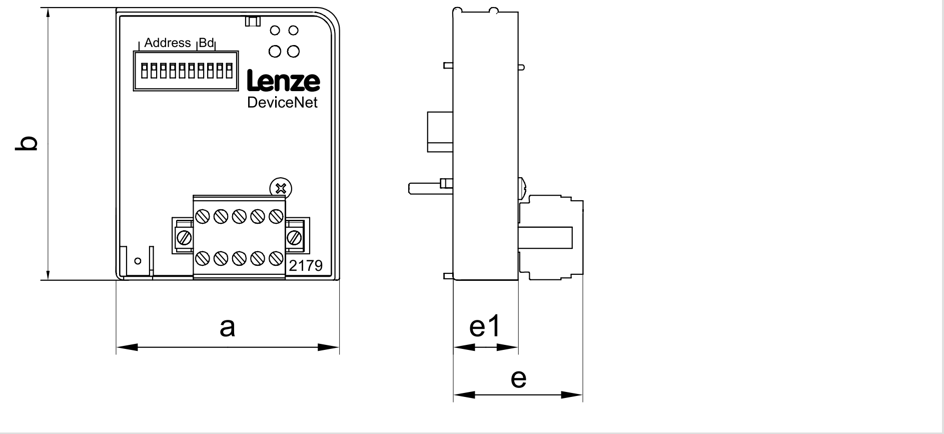
Drive PLC / 9300 servo PLC / ECS application processing times

Parameter data	Process data
Approx. 30 ms + 20 ms (typical) For some codes, the processing time can be longer.	Depending on the process image

ECS, posi&shaft / ECS, speed&torque processing times

Parameter data	Process data	
Approx. 30 ms + 20 ms (typical) For some codes, processing can take longer.	ECS, posi & shaft ECS, speed & torque	6 ms 2 ms

5.3.6 Dimensions



a	62 mm
b	75 mm
e	36 mm
e1	18 mm

5.4 Installation

5.4.1 Elements at the front of the communication module

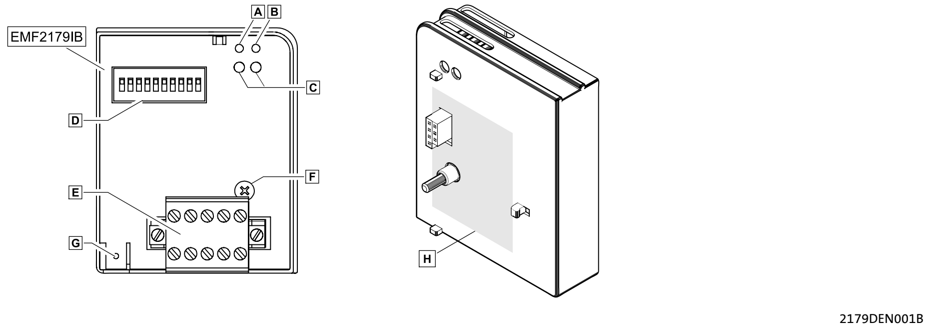


Fig. 5.4-1 EMF2179IB communication module (DeviceNet)

Connections

Pos.	Description	Notes
E	Plug connector with double screw connection, 5-pole	🕮 5.4-11
G	PE shield cable connection	

DIP switch

Pos.	Description	Notes
D	DIP switches for setting ● controller address (S1 - S6) ● baud rate (S7, S8) ● software compatibility with 2175 communication module (S10)	🕮 5.5-2

Displays

Pos.	Description	Notes
A	Connection status to the drive controller (two-coloured LED)	🕮 5.5-11
B	Connection status to the bus (two-coloured LED)	
C	Drive (green and red drive LED)	

Other elements

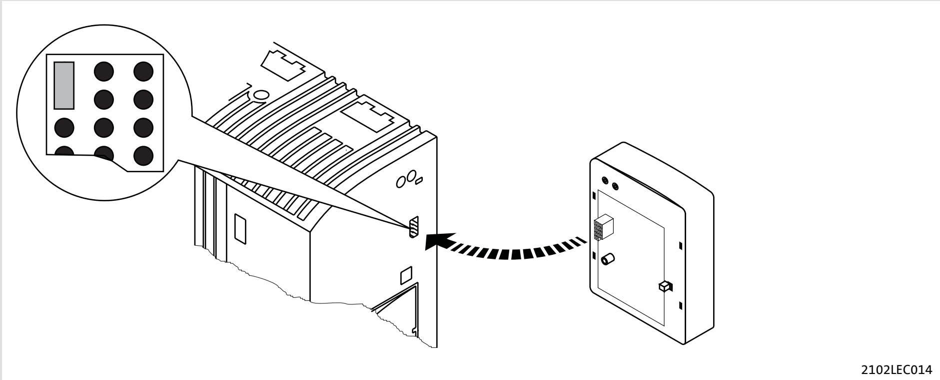
Pos.	Description	Notes
F	Fixing screw	
H	Nameplate	🕮 5.2-1

Note!

Only for 820X and 821X:

If required, use an additional PE shield cable which avoids EMC-related communication interference in surroundings with extreme disturbances.

5.4.2 Mechanical installation



- Plug the communication module onto the basic device (here: 8200 vector).
- Screw the communication module to the basic device to ensure a good PE connection.

Note!

For the internal supply of the communication module through the 8200 vector frequency inverter, the jumper in the interface opening must be adapted (see fig. above). Please observe the notes (5.4-11).

5.4.3 Electrical installation

Wiring to a host

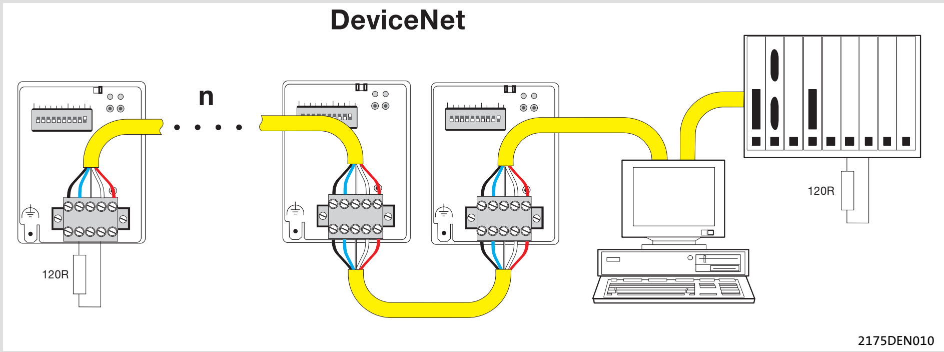
Danger!

Additional electrical isolation must be installed if

- an 820X and 821X controller is connected to the host and
- reliable electrical isolation (reinforced insulation) in accordance with EN 61800-5-1 is needed.

For this, you can use an interface module for the host with an additional electrical isolation (see the corresponding manufacturer's information).

For wiring, the electrical isolation of the supply voltage must be taken into account. The supply voltage is assigned to the same potential as the data bus.



n	Number of nodes
120R	Terminating resistor

A DeviceNet line can have max. 63 participants. The participants are

- the connected basic devices
- the DeviceNet master (scanner)
- all other components which take part in the communication.

In there, the basic devices with plugged-on communication modules and the DeviceNet master can communicate.

A PC with installed software (e.g. RSNetWorx) is used to integrate the communication modules.

Note!

Please observe that

- the shield on the voltage supply is to be connected together with the ”V-” connection to GND once. For this, use the centre point of the DeviceNet line, if possible.
- for each participant the shield of the DeviceNet cable is only connected to the ”shield” connection of the plug connector.
- the DeviceNet line is terminated by 120-Ohm resistors at both ends.

EMC-compliant wiring

Please observe the following for wiring according to EMC guidelines:

Note!

- With 820X and 821X controllers, communication can be impaired by electromagnetic interferences. For safe communication, use an additional cable between the PE connection of the basic device and the PE connection of the communication module.

  This is not necessary for all other controllers that can be used together with the communication module.
- Differences in potential between the devices can be avoided by using an equalizing conductor with a large cross-section (reference: PE).
- Separate control cables from motor cables.
- Connect the data cable shields *at both ends*.
- Please see the information on wiring according to EMC guidelines in the Operating Instructions for the basic device.

Specification of the transmission cable

The devices are connected to the bus system via a fieldbus cable according to the DeviceNet^{TM} specification (DeviceNet Adaption of CIP, Edition 1.1, Volume Three). Companies like Belden Wire & Cable, Olflex Wire & Cable, C&M Corp. and Madison Cable produce DeviceNet^{TM} “Thick” and “Thin” cables

Stop!

If you do not want to use the “Thick” or “Thin” cables, the cable you use must meet the demands of the DeviceNet specification.

A cable with features that do not comply with the demands is not permissible and must not be used !

Features of the “Thick” cable according to DeviceNet specifications

General features	
Configuration	Two shielded symmetrical cables with a common axis and drain wire in the centre
Total shielding	65% coverage AWG 36 or min. 0.12 mm tinned braid (individually tinned)
Drain wire	Copper 18 min.; min. 19 cores (individually tinned)
Outer diameter	10.41 mm (min.) to 12.45 mm (max.)
Roundness	Radius deviation must be within 15% of 0.5 outer diameter
Jacket marking	Vendor name & part no. and additional markings
Spec. DC resistor (braid, tape, drain)	5.74 Ohms/1000 m (nominal up to 20°C)
Certifications (U.S. and Canada)	NEC (UL), CL2/CL3 (min.)
Bend radius	20 x diameter (installation) / 7 x diameter (fixed)
Ambient operating temperature	-20°C to +60°C at 8 Ampere; derate current linearly to zero at 80°C
Storage temperature	-40 to +85°C
Pull tension	845.5 N. max.

Features of the data line	
Conductor pair	Copper 18 min.; min. 19 cores (individually tinned)
Insulation diameter	3.81 mm (nominal)
Colours	Light-blue, white
Pair twist / m	Approx. 10
Tape shield over conductor pair	2000/1000, Al/Mylar, Al side out, w-shorting fold (pull-on applied)
Impedance	120 Ohms +/- 10% (at 1 MHz)
Capacitance between conductors	39.37 pF / m at 1 kHz (nominal)
Capacitance between a condcutor and the conductor connected to the shield	78.74 pF / m at 1 kHz (nominal)
Capacitive unbalance	3937 pF/1000 m at 1 kHz (nominal)
Spec. DC resistor at 20ºC	22.64 Ohms/1000 m (max.)
Attenuation	0.43 dB/100 m at 125 kHz (max.) 0.82 dB/100 m at 500 kHz (max.) 1.31 dB/100 m at 1.00MHz (max.)

Features of the voltage line	
Conductor pair	Copper 15 min.; min. 19 cores (individually tinned)
Insulation diameter	2.49 mm (nominal)
Colours	Red / black
Pair twist / m	Approx. 10
Tape shield over conductor pair	1000/1000, Al/Mylar, Al side out, w-shorting fold (pull-on applied)
Spec. DC resistor at 20°C	11.81 Ohms/1000 m (max.)

Features of the "Thin" cable according to DeviceNet specifications

General features	
Configuration	Two shielded symmetrical cables with a common axis and drain wire in the centre
Total shielding	65% coverage AWG 36 or min. 0.12 mm tinned braid (individually tinned)
Drain wire	Copper 22 min.; min. 19 cores (individually tinned)
Outer diameter	6.096 mm (min.) to 7.112 mm (max.)
Roundness	Radius deviation must be within 20% of 0.5 outer diameter
Jacket marking	Vendor name & part no. and additional markings
Spec. DC resistor (braid, tape, drain)	10.5 Ohm/1000 m (nominal at 20°C)
Certifications (U.S. and Canada)	NEC (UL), CL2 (min.)
Bend radius	20 x diameter (installation) / 7 x diameter (fixed)
Ambient operating temperature	-20°C to +70°C at 1.5 Ampere; derate current linearly to zero at 80°C
Storage temperature	-40°C to +85°C
Pull tension	289.23 N_{max}

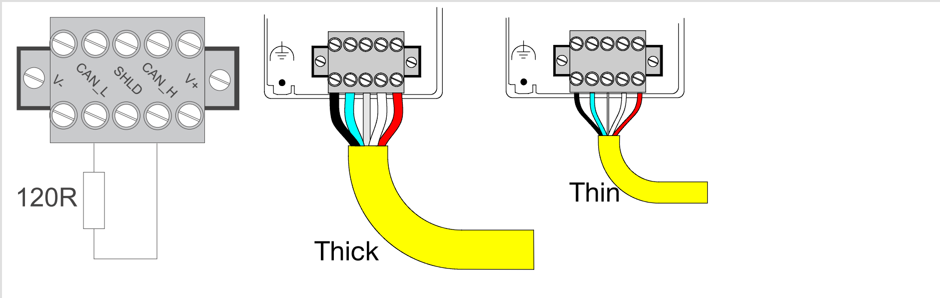
Features of the data line	
Insulation diameter	1.96 mm (nominal)
Conductor pair	Copper 24 min.; min. 19 cores (individually tinned)
Colours	Light-blue, white
Pair twist / m	Approx. 16
Tape shield over conductor pair	1000/1000, Al/Mylar, Al side out, w-shorting fold (pull-on applied)
Impedance	120 Ohm +/- 10% (at 1 MHz)
Propagation delay	4.46 ns/m (max.)
Capacitance between conductors	39.37 pF / m at 1 kHz (nominal)
Capacitance between a condcutor and the conductor connected to the shield	78.74 pF / m at 1 kHz (nominal)
Capacitive unbalance	3.94 pF/1000 m at 1 kHz (max.)
Spec. DC resistor at 20°C	91.86 Ohm/1000 m (max.)
Attenuation	0.95 dB/100 m at 125 kHz (max.) 1.64 dB/100 m at 500 kHz (max.) 2.30 dB/100 m at 1.00MHz (max.)

Features of the voltage line	
Conductor pair	Copper 22 min.; min. 19 cores (individually tinned)
Insulation diameter	1.4 mm (nominal)
Colours	Red, black
Pair twist / m	Approx. 16
Tape shield over conductor pair	1000/1000, Al/Mylar, Al side out, w-shorting fold (pull-on applied)
Spec. DC resistor at 20°C	57.41 Ohm/1000 m (max.)

5.4.4 Communication connection

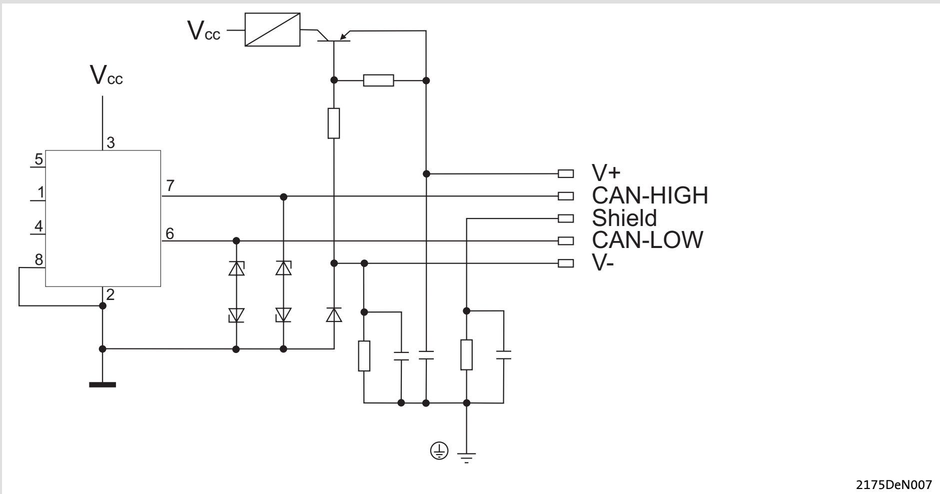
Use the 5-pole plug connector with double screw connection to connect the communication module to the bus.

The assignment of the plug connector and the cable colour used according to the DeviceNet Specification are listed in the table.



Designation	Cable colour	Explanation
V-	Black	Reference for external voltage supply
CAN_L	Blue	Data cable / input for terminating resistor of 120 Ohm
SHLD		Shield
CAN_H	White	Data cable / input for terminating resistor of 120 Ohm
V+	Red	External voltage supply ; see 🕮 5.4-11

Internal wiring of the bus terminals



Max. possible bus cable length

The following bus cable lengths are possible in dependence on the baud rate and the cable used:

Baud rate	Thin Cable	Thick Cable
125 kbits/s	100 m	500 m
250 kbits/s		250 m
500 kbits/s		100 m

When using both, “Thick” and “Thin” cables, the maximum cable lengths are to be selected according to the baud rate:

Baud rate	Bus cable length
125 kbits/s	L_{max} = 500 m = $L_{thick} + 5\ L_{thin}$
250 kbits/s	L_{max} = 250 m = $L_{thick} + 2.5\ L_{thin}$
500 kbits/s	L_{max} = 100 m = $L_{thick} + L_{thin}$

5.4.5 Voltage supply

External voltage supply

For DeviceNet always an external voltage supply is used.

Designation	Explanation
V+	External supply (exceeds the selection of the DeviceNet specification) V = 24 V DC (21.6 V - 0 % ... 26.4 V + 0 %) I = 100 mA
V-	Reference potential for external voltage supply

If the distance between the DeviceNet participants is larger than normal, you can use several voltage supplies.

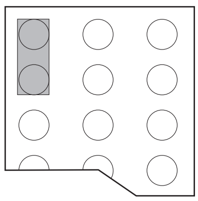
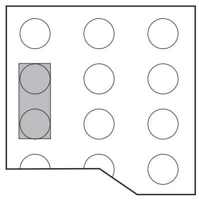
Controller	External voltage supply
820X	Always required
821X, 822X, 824X, 93XX and ECS	Only necessary, if the mains supplying the corresponding controllers is to be switched off but the communication must not be interrupted.
8200 vector	See “Internal DC voltage supply”

Internal DC voltage supply

Note!

The internal voltage supply option is available for basic devices with extended AIF interface opening (8200 vector front). The area marked in grey in the graphic representation indicates the jumper position.

- ► In the delivery status of the frequency inverters, these are *not* supplied internally.
- ► For internal voltage supply put the jumper on the position indicated below.

Lenze setting (only external voltage supply)	**Internal voltage supply**
	

5.5 Commissioning

5.5.1 Possible settings via DIP switch

Note!

The Lenze setting for all switches is OFF.

The device address and baud rate set via DIP switch will only be active after a renewed mains connection.

Switch S9 is ineffective.

The following settings can be easily carried out via the front DIP switches of the communication module:

- ► Software compatibility of 2175 DeviceNet / 2179 communication module with S10
- ► Device address with S1 - S6
- ► Baud rate with S7 / S8

Adjustment of software compatibility

Note!

If compatibility is active (S10 = ON), please observe the information in the instructions for the 2175 communication module (Mounting Instructions or part 2175 of the DeviceNet Communication Manual).

This particularly applies to DIP switch assignments changed with this setting.

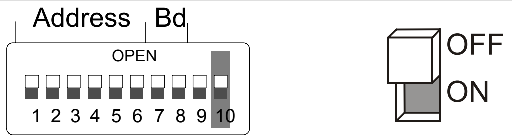


Fig. 5.5-1 Setting the software compatibility

Compatibility	S10
2179	OFF
2175 For the communication module description, please see e.g. 2175 Mounting Instructions, part DeviceNet	ON

Setting of the device address

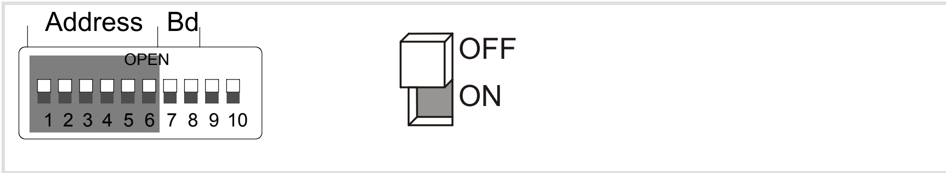


Fig. 5.5-2 Address assignment via DIP switch

The address (decimal number) is calculated by inserting the positions of switches S1 to S6 ('0' = OFF and '1' = ON) into the following equation.

$$\mathbf{Address_{dec} = S6 \cdot 2^0 + S5 \cdot 2^1 + S4 \cdot 2^2 + S3 \cdot 2^3 + S2 \cdot 2^4 + S1 \cdot 2^5}$$

The equation can also be used to calculate the valency of a switch. The sum of valencies results in the node address to be set:

Switch	Valency	Example	
		Switch position	Node address
S1	32	**ON**	32 + 16+ 8 = 56
S2	16	**ON**	
S3	8	**ON**	
S4	4	OFF	
S5	2	OFF	
S6	1	OFF	

Baud rate setting

Note!
The baud rate must be the same for all devices and the scanner.

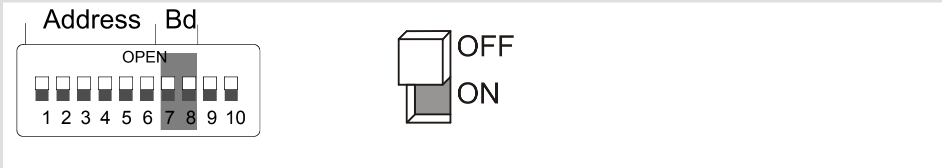


Fig. 5.5-3 Baud rate setting

Baud rate	S7	S8
125 kbits/s	OFF	OFF
250 kbits/s	OFF	ON
500 kbits/s	ON	OFF

5.5.2 Before switching on

Stop!

Before you switch on the standard device with the communication module for the first time, check

- ► the entire wiring for completeness, short circuit and earth fault.
- ► whether the bus system is terminated through the bus terminating resistor at the first and last physical bus station.

5.5.3 First switch-on

Note!
Follow the commissioning steps in the given order!

1. Switch on the basic device and the external supply of the communication module.
   – The status message display of the controller (”Drive-LED”) must light up green or be blinking.
   – The LED ”Connection status to basic device” must light up green.
   – The LED ”Connection status to bus” must be blinking green.
2. Use the configuration software (e.g. RSNetWorx) to integrate the communication module into the DeviceNet.
   – When the communication module has been configured, the status of the LED ”Connection status to bus” changes from “blinking” to “on”.
3. It is now possible to communicate with the drive, i.e.
   – Via “explicit messages” you can read and write all parameters from the drive and/or communication module.
   – You can read actual values (e.g. status words) or write setpoints (e.g. frequency setpoints).

5.5.4 Preparing the basic device for communication

Note!
During operation, the change of a communication module to another basic device can lead to undefined operating states.

82XX / 8200 vector frequency inverter

1. Set the Lenze parameter operating mode (C0001) to ”3” to enable the controller via the communication module. This can be made using the keypad or directly via DeviceNet.

   Lenze codes in the controller and communication module can be read and written via the vendor-specific class 110.

► Example: write C0001 = 3
   – Class: 0x6E
   – Instance: 0x1
   – Attribute: 0x1
   – Service code: set single attribute
   – Data sent: 0x7530 (30 000_{dec})

   For further examples, see (📖 5.7-17).

2. Terminal 28 (controller enable) is always active and must be set to HIGH level during DeviceNet operation (see Operating Instructions for the controller).
   – Otherwise, the controller cannot be enabled.
   – With 821X, 8200vector and 822X, the QSP (quick stop) function is always active. If QSP is assigned to an input terminal (Lenze setting: not assigned), this terminal must be at HIGH level during DeviceNet operation (see Operating Instructions for the controller).

93XX servo inverter

1. For drive control via DeviceNet set the Lenze parameter signal configuration (C0005) to a value xxx3 (e.g. “1013”, speed control via the communication module). This change can be carried out using the 9371BB keypad or the DeviceNet.

► Example: write C0005 = 1013
   – Class: 0x6E
   – Instance: 0x5
   – Attribute: 1
   – Service code: set single attribute
   – Data sent: 0x9A9250 (10 130 000_{dec})

2. Set the parameter C0142 to 0
3. Terminal 28 (controller enable) is always active and must be set to HIGH level during DeviceNet operation (see Operating Instructions for the controller). Otherwise, the controller cannot be enabled.
   – With the signal configuration C0005=1013, the QSP (quick stop) function and the CW/CCW changeover are assigned to the digital input terminals E1 and E2 and thus they are always active. For DeviceNet operation, E1 must be set to HIGH level (see Operating Instructions 93XX).

Note!

With the signal configuration C0005=xx13, terminal A1 is switched as voltage output. Connect the following terminals:

► X5.A1 with X5.28 (controller enable)
► X5.A1 with X5.E1 (CW/QSP)

ECS servo system, variants ”Speed & Torque” and ”Posi & Shaft”

1. For drive control via DeviceNet, use the EMZ9371BC keypad or the DeviceNet and set the Lenze parameter signal configuration (C0005) to a value implementing communication via the AIF interface:

► Variant ”Speed & Torque”: C0005 = 1013
(speed control via AIF interface)

► Variant ”Posi & Shaft”: C4010 = 2
(automation interface AIF X1)

► Example: write C0005 = 1013
– Class: 0x6E
– Instance: 0x5
– Attribute: 1
– Service code: set single attribute
– Data sent: 0x9A9250 (10 130 000$_{dec}$)

2. Terminal SI1 (controller enable) and SI2 (IMP = pulse inhibit) are always active and must be set to HIGH level during operation. Otherwise, the controller is inhibited.

Note!
The controller must always be externally supplied with 24 V DC.

3. The controller can now accept DeviceNet control and parameterisation data.
4. Select speed / position setpoint

► Variant ”Speed & Torque”: select the speed setpoint with a value unequal 0.

► Variant ”Posi & Shaft”: select the position setpoint with a value unequal 0.
Prior to this, the position profiles must be parameterised according to the Operating Instructions for the controller.

Protection against uncontrolled restart

Note!

In some cases the controller should not restart after a fault (e.g. after a short mains failure).

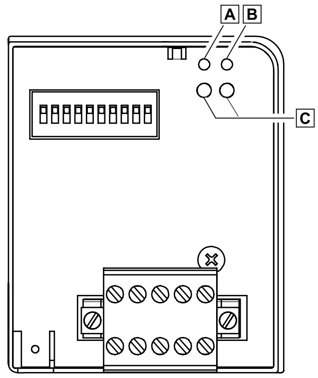
- ► The drive can be inhibited by setting C0142 = 0 if
  - – the corresponding controller sets a fault ”LU message” and
  - – the fault is active for more than 0.5 seconds.

Parameter function:

- ► C0142 = 0
  - – The controller remains inhibited even after the fault has been eliminated and
  - – the drive restarts in a controlled mode: LOW-HIGH transition at terminal 28 (CINH)
- ► C0142 = 1
  - – An uncontrolled restart of the controller is possible.

Lenze codes in the basic device and communication module can be read and written via the vendor-specific class 110. An example is included in the appendix of the DeviceNet Manual (Class Instance Editor).

5.5.5 Status display



Pos.	Colour	Condition	Description
A	Green	Blinking	Communication module is supplied with voltage but is not connected to the controller (controller is switched off, initialising, or not available).
		On	Communication module is supplied with voltage and is connected to the controller.
B	Off		• No communication with the communication module • Communication module is not supplied with voltage
	Green	Blinking	Dup_Mac_ID testing. Still no connection to the master.
	Green	On	DeviceNet connection built up.
	Red	Blinking	No communication because time limit exceeded
	Red	On	Internal fault of the communication module
C	Operating status of the basic device (see instructions for the basic device)		

5.6 Data transfer

Note!

When using the DeviceNet communication profile, the corresponding specified terminology must be considered. Note that translation into German is not always permissible.

In these instructions, the following terms are used with the same meaning:

- I/O data ↔ Process data
  - Input data is process data to the scanner
  - Output data is process data from the scanner
- Explicit Messages ↔ Parameter data
- Scanner ↔ DeviceNet master

Telegram types

Between the master computer and the controller(s), the ”I/O data” and ”Explicit Messages” telegram types are transmitted.

- I/O data

  I/O data (process data) is transmitted or received according to the producer/consumer principle, i. e. there is one transmitter and no or an arbitrary number of receivers.

  The following transmission modes are supported:

  - I/O polled messages (polled): the poll command sent by the master includes output data for the slave. The slave then sends its input data to the master. The poll response can also be used to confirm the received data.
  - Cyclic I/O: by means of cyclic I/O, master and slave create data independently of each other which is sent according to the settings of a timer. The user must set the timer.
  - This type of I/O message is a special cyclic message. COS nodes send their data when the data status is changed.

- Explicit messages

  Explicit messages are used for configuration and parameter setting of the devices connected to the DeviceNet. The relationship between two devices is a client-server relationship. The client sends the request and the server accepts the order and tries to carry it out. The server reacts as follows

  - the required data in case of a positive response or
  - a fault message in case of a negative response.

  The target address and the service (request <-> response) are data coded.

Dividing the telegram types into communication channels

Depending on their time-critical behaviour, the telegram types are divided into corresponding communication channels.

- The process data channel transmits ”I/O data”
  - By means of the I/O data you can control the controller.
  - The host can directly access the I/O data. The data is directly stored int the I/O area of the PLC.
  - I/O data is not stored in the controller; they are cyclically transferred between the host and the controllers (continuous exchange of input and output data).
- The parameter data channel transmits ”explicit messages”
  - Enables access to all Lenze codes.
  - Saving parameter changes:
    Automatic storage for the 82XX and 8200 vector frequency inverters
    Manually required storage for the 93XX servo inverters
    (Lenze code C0003).

Note!

- As default message type, ”Polled” is set in the communication module.
- The selection and configuration of the message types is effected by means of software via the scanner.

5.7 Lenze codes and DeviceNet objects

5.7.1 Diagnostics

C1810: software ID

For the component initialisation, it is determined by means of the ID (identification code) which device is connected as node.

The code contains a string which is 14 bytes long. The product code is output, e.g. 33F2179I_XXXXX.

C1811: Software creation date

The code includes a string with a length of 17 bytes. The creation date and time of the software are output, e.g. June 21 2003 12:31.

In the first place this information is important for the service.

5.7.2 Implemented DeviceNet objects

A device connected to the DeviceNet is a conglomeration of objects. Every individual object is described by its class, instances and attributes. These objects can be used with different services such as reading or writing.

Overview of the implemented objects:

Object	Class	Notes
Identity	01_{hex}	-
DeviceNet	03_{hex}	-
Assembly	04_{hex}	-
Connection	05_{hex}	-
Acknowledge handler	$2B_{hex}$	-
Lenze	65_{hex}	Response to idle mode, communication interruption and bus error
	66_{hex}	Changing the I/O data length
	67_{hex}	I/O image of the transmitted data
	68_{hex}	I/O image of the received data
	$6E_{hex}$	Access to Lenze codes

Identity class (01_{hex})

Instance 1:

Attribute	Service(s)	Description	Data type	Value
1	GET	Vendor ID	UINT	445 ($01BD_{hex}$)
2	GET	Device type	UINT	0 (generic)
3	GET	Product code	UINT	1024 (0400_{hex})
4	GET	Revision	Struct of	
		Major revision	USINT	1 (01_{hex})
		Minor revision	USINT	1 (01_{hex})
5	GET	STATUS	WORD	Depending on the current module status
6	GET	Serial number	UDINT	Individually for the respective module
7	GET	Product name	SHORT_STRING	e.g. "EMF2179IB"
9	GET	Configuration consistency value	UINT	If stored in the EEPROM, value is increased

Services:

Service code	Name	Description
$0E_{hex}$	Get_Attribute_Single	Reading of an attribute
05_{hex}	Reset	Reset of the communication module

DeviceNet Class (03_{hex})

Instance 0:

Attribute	Service(s)	Description	Data type	Value
1	GET	Revision	UINT	0002_{hex}

Instance 1:

Attribute	Service(s)	Description	Data type	Value
1	GET	MAC ID	USINT	0 - 63
2	GET	Baud rate	USINT	0 - 2
3	GET / SET	BOI	BOOL	0/1
4	GET	Bus-off counter	USINT	0 - 255
5	GET	Allocation information	Struct of	
		Allocation choice byte	BYTE	0 – 63
		Master's MAC ID	USINT	0 – 63

Services:

Service code	Name	Description
$0E_{hex}$	Get_Attribute_Single	Reading of an attribute
10_{hex}	Set_Attribute_Single	Writing of an attribute
$4B_{hex}$	Allocate_Master/Slave_Connection_Set	Demands the application of "Predefined Master/Slave Connection Set"
$4C_{hex}$	Release_Group_2_Identifier_Set	Connections via "Predefined Master/Slave Connection Set" are deleted

Assembly class (04_{hex})

Instance 101 ... 112:

Attribute	Service(s)	Description	Data type	Instance / value
3	GET / SET	Data	Array of BYTE	Instance 101: 1 word (= 2 byte) from master
				Instance 102: 2 words (= 4 byte) from master
				Instance 103: 3 words (= 6 byte) from master
				Instance 104: 4 words (= 8 byte) from master
				Instance 105: 5 words (= 10 byte) from master
				Instance 106: 6 words (= 12 byte) from master
				Instance 107: 7 words (= 14 byte) from master
				Instance 108: 8 words (= 16 byte) from master
				Instance 109: 9 words (= 18 byte) from master
				Instance 110: 10 words (= 20 byte) from master
				Instance 111: 11 words (= 22 byte) from master
				Instance 112: 12 words (= 24 byte) from master

Instance 114 ... 125:

Attribute	Service(s)	Description	Data type	Entity / value
3	GET / SET	Data	Array of BYTE	Instance 114: 1 word (= 2 byte) from master
				Instance 115: 2 words (= 4 byte) from master
				Instance 116: 3 words (= 6 byte) from master
				Instance 117: 4 words (= 8 byte) from master
				Instance 118: 5 words (= 10 byte) from master
				Instance 119: 6 words (= 12 byte) from master
				Instance 120: 7 words (= 14 byte) from master
				Instance 121: 8 words (= 16 byte) from master
				Instance 122: 9 words (= 18 byte) from master
				Instance 123: 10 words (= 20 byte) from master
				Instance 124: 11 words (= 22 byte) from master
				Instance 125: 12 words (= 24 byte) from master

Services:

Service code	Name	Description
$0E_{hex}$	Get_Attribute_Single	Reading an attribute
10_{hex}	Set_Attribute_Single	Writing an attribute

Connection Class (05_{hex})

Instance 1 (explicit messages):

Attribute	Service(s)	Description	Data type	Value
1	GET	state	USINT	Status of the object
2	GET	instance_type	USINT	0
3	GET	transportClass_trigger	BYTE	131 (83_{hex})
4	GET	produced_connection_id	UINT	Send CAN identifier
5	GET	consumed_connection_id	UINT	Reception of CAN identifier
6	GET	initial_comm_characteristics	BYTE	33 (21_{hex})
7	GET	produced_connection_size	UINT	64 (40_{hex})
8	GET	consumed_connection_size	UINT	64 (40_{hex})
9	GET / SET	expected_packet_rate	UINT	Connection-dependent
10/11		Not used		No longer defined
12	GET / SET	watchdog_timeout_action	USINT	Defined reaction to timeout ● 1 = Auto Delete ● 3 = Deferred Delete
13	GET	produced_connection_path_length	UINT	0
14	GET	produced_connection_path	EPATH	---
15	GET	consumed_connection_path_length	UINT	0
16	GET	consumed_connection_path	EPATH	---
17	GET	production_inhibit_time	UINT	0

Instance 2 (polled I/O data):

Attribute	Service(s)	Description	Data type	Value
1	GET	state	USINT	Status of the object
2	GET	instance_type	USINT	1
3	GET	transportClass_trigger	BYTE	128/130 (80_{hex}/82_{hex})
4	GET	produced_connection_id	UINT	Send CAN identifier
5	GET	consumed_connection_id	UINT	Reception of CAN identifier
6	GET	initial_comm_characteristics	BYTE	1 (01_{hex})
7	GET	produced_connection_size	UINT	Dependent on the number of I/O data words
8	GET	consumed_connection_size	UINT	Dependent on the number of I/O data words
9	GET / SET	expected_packet_rate	UINT	Connection-dependent
10/11		Not used		No longer defined
12	GET	watchdog_time-out_action	USINT	Defines time-outs
13	GET	produced_connection_path_length	UINT	4
14	GET	produced_connection_path	EPATH	[20_{hex}, 67_{hex}, 24_{hex}, 01]
15	GET	consumed_connection_path_length	UINT	4
16	GET	consumed_connection_path	EPATH	[20_{hex}, 68_{hex}, 24_{hex}, 01_{hex}]
17	GET	production_inhibit_time	UINT	0

Instance 4 (COS I/O):

Attribute	Service(s)	Description	Data type	Value
1	GET	state	USINT	Status of the object
2	GET	instance_type	USINT	1
3	GET	transportClass_trigger	BYTE	128/130 (80_{hex}/82_{hex})
4	GET	produced_connection_id	UINT	Send CAN identifier
5	GET	consumed_connection_id	UINT	Reception of CAN identifier
6	GET	initial_comm_characteristics	BYTE	1 (01_{hex})
7	GET	produced_connection_size	UINT	Dependent on the number of I/O data words
8	GET	consumed_connection_size	UINT	Dependent on the number of I/O data words
9	GET / SET	expected_packet_rate	UINT	Connection-dependent
10/11		Not used		No longer defined
12	GET	watchdog_timeout_action	USINT	Defines time-outs
13	GET	produced_connection_path_length	UINT	4
14	GET	produced_connection_path	EPATH	[20_{hex}, 67_{hex}, 24_{hex}, 01_{hex}]
15	GET	consumed_connection_path_length	UINT	4
16	GET	consumed_connection_path	EPATH	[20_{hex}, 68_{hex}, 24_{hex}, 01_{hex}]
17	GET / SET	production_inhibit_time	UINT	0

Services:

Service code	Name	Description
05_{hex}	Reset_Request	Reset effect: ● Reset of the watchdog timer ● Communication between scanner and slave in established status.
$0E_{hex}$	Get_Attribute_Single	Reading an attribute
10_{hex}	Set_Attribute_Single	Writing an attribute

Acknowledge Handler Class ($2B_{hex}$)

Instance 1:

Attribute	Service(s)	Description	Data type	Value
1	GET / SET	Acknowledge Timer	UINT	2 – 65534 ms (0002_{hex} –$FFFE_{hex}$), default 16 ms (0010_{hex})
2	GET	Retry Limit	USINT	0 – 255 ms (00_{hex} – FF_{hex}), default 1 ms
3	GET	COS Producing Connection Instance	UINT	4 (0004_{hex})

Services:

Service code	Name	Description
$0E_{hex}$	Get_Attribute_Single	Reading an attribute
10_{hex}	Set_Attribute_Single	Writing an attribute

Manufacturer-specific class 101 (65_{hex}): response to IdleMode, response to communication errors, response to BusOff

Instance 0:

Attribute	Service(s)	Description	Data type	Value
1	GET	Revision	UINT	0001_{hex}

Instance 1:

Attribute	Service(s)	Description	Data type	Value
1	GET / SET	Response to IdleMode	UINT	0 = no response 1 = CINH 2= QSP
2	GET / SET	Response to communicatio n errors	UINT	0 = no response 1 = CINH 2= QSP
3	GET / SET	Response to BusOff	UINT	0 = no response 1 = CINH 2= QSP

Services:

Service code	Name	Description
$0E_{hex}$	Get_Attribute_Single	Reading of an attribute
10_{hex}	Set_Attribute_Single	Writing of an attribute

Note!

The parameters of 65_{hex} are described in the EDS file and therefore can be directly set in the feature dialog of the DeviceNet nodes under ”Parameters” via the »RSNetWorx« Rockwell software.

Manufacturer-specific class 102 (66_{hex})

Instance 0:

Attribute	Service(s)	Description	Data type	Value
1	GET	Revision	UINT	0001_{hex}

Instance 1:

Attribute	Service(s)	Description	Data type	Value
1	GET/SET	I/O data length in words	UINT	1 – 12 (0000_{hex} – $000C_{hex}$), Default: 2, (saved in EEPROM)

Services:

Service code	Name	Description
$0E_{hex}$	Get_Attribute_Single	Reading of an attribute
10_{hex}	Set_Attribute_Single	Writing of an attribute

Note!

- If the I/O data length is changed, the DeviceNet scanner will be informed about this change (produced/consumed data size).
- If the I/O data length is reduced, it must be checked before, if the intended data length is sufficient for the application.
- The parameters of 66_{hex} are described in the EDS file and therefore can be directly set in the feature dialog of the DeviceNet nodes under "Parameters" via the »RSNetWorx« Rockwell software.
- In case of a number of process data words deviating from the Lenze setting (2 words), the "Automap on Add" function in the scanner configuration dialog of the »RSNetWorx« software has to be deactivated, and the number of process data words has to be adjusted via the "Edit I/O parameters" button.

Manufacturer-specific class 103 (67_{hex})

Instance 0:

Attribute	Service(s)	Description	Data type	Value
1	GET	Revision	UINT	0001_{hex}

Instance 1:

Attribute	Service(s)	Description	Data type	Value
1	GET	I/O image of the transmitted data (input data of the scanner)	Array of UINT	Value according to the words set

Services:

Service code	Name	Description
$0E_{hex}$	Get_Attribute_Single	Reading of an attribute

Manufacturer-specific class 104 (68_{hex})

Instance 0:

Attribute	Service(s)	Description	Data type	Value
1	GET	Revision	UINT	0001_{hex}

Instance 1:

Attribute	Service(s)	Description	Data type	Value
1	GET/SET	I/O image of the received data (output data of the scanner)	Array of UINT	Value according to the words set

Services:

Service code	Name	Description
$0E_{hex}$	Get_Attribute_Single	Reading of an attribute
10_{hex}	Set_Attribute_Single	Writing of an attribute

Manufacturer-specific class 110 ($6E_{hex}$): access to Lenze codes

Instance (Lenze code):

Attribute	Service(s)	Description	Data type	Value
Lenze subcode	GET / SET	Access to Lenze code ($6E_{hex}$)	Data type of the Lenze code	Value of the Lenze code/subcode

Note!

- If the corresponding Lenze code does not have a subcode, the value "1" must be entered into the attribute.
- The display code cannot be configured by the "SET" service.

5.7.2.1 Examples for reading / writing with the »Class Instance Editor«

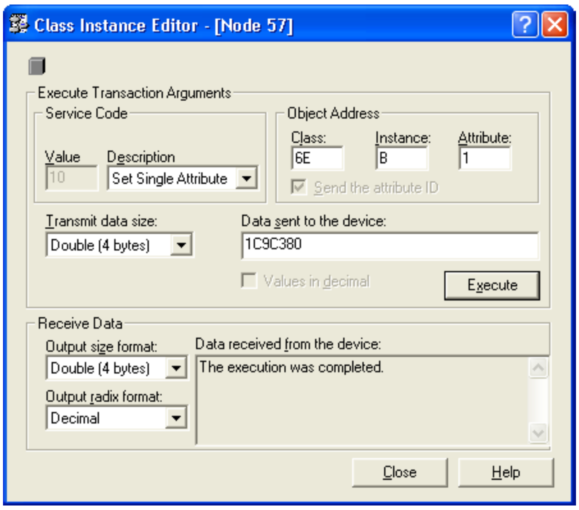
Tip!
The »Class Instance Editor« is a function of the »RSNetworx«Rockwell software

Example 1: writing to the code

The maximum speed (n_{max}) is to be set to 3000 min^{-1} via code C0011 .

Please enter the following in the »Class Instance Editor« (see graphics):

- ”Description”: set single attribute (write to code).
- ”Class”: 6E (access to Lenze code)
- ”Instance”: B (hex value of the code)
- ”Attribute”: 1 (hex value of the subcode)
- ”Data sent to the device”
  3000 [min^{-1}] ==> x 10 000 (only in this example) =
  30000000_{dez} = $1C9C380_{hex}$ (enter speed as hex value)
- Further information on the data format



2179DEN011

Example 2: reading the code

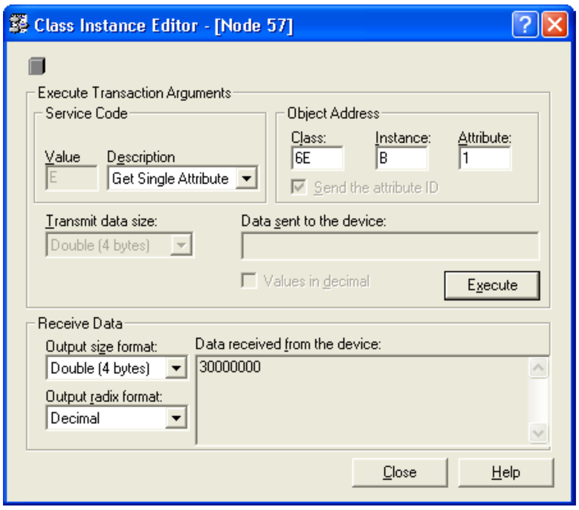
The maximum speed (n_{max}) is to be read from code C0011.

Please enter the following in the »Class Instance Editor« (see graphics):

- "Description": get single attribute (read code).
- "Class": 6E (access to Lenze code)
- "Instance": B (hex value of the code)
- "Attribute": 1 (hex value of the subcode)
- Further information on the data format

Read the returned decimal data from the »Class Instance Editor« and divide the value by the factor 10 000 (see graphics):

- "Data received from the device":
  30000000_{dez} ==> / 10 000 (only in this example) = 3000 [min^{-1}]



2179DEN012

5.7.3 Specific features for parameterising the controller

8200 drive controller

For the 8200 inverter series, the following specific features apply:

> **Danger!**
>
> Parameter setting (codes except C046, C0135) can only be effected if the controller is inhibited. Though parameters are accepted if the controller is enabled, they are discarded afterwards.
> After parameterisation of a value, the controller may not be activated via "Explicit Message" for approx. 50 ms, as otherwise these commands are ignored!
>
> After completion of the entire paramterisation process, a period of approx. 70 ms can pass until controller enable (terminal, C040, C0135) is accepted.
>
> The TRIP reset function (reset of a fault) is carried out by setting controller inhibit and then controller enable via C040 or C0135.

5.8 Troubleshooting

No communication with the controller.

Possible causes	Diagnostics	Remedy
Is the controller switched on?	One of the operating status LEDs of the basic device must be on.	Supply controller with voltage.
Is the communication module supplied with voltage?	The LED ”Connection status to the basic device” must be lit or blinking green.	Check the external voltage supply. The measured voltage value at the terminals for external voltage supply of the communication module must be in the range of 24 V ± 10 %.
		The communication module has not yet been initialised with the controller. Possibility 1: controller not switched on Possibility 2: check the connection to the controller
Does the controller receive telegrams?	The LED ”Connection status to the bus” at the communication module must be blinking green when communicating with the master computer.	Check your wiring (see 🕮 5.4-1). Check whether your master computer sends telegrams. Check the data assignment in the scan list (I/0 mapping).
		Is the available device address already assigned? Check the setting of the other nodes on the DeviceNet.

5.9 Appendix

5.9.1 Program examples

Tip!

Current program examples for this Lenze product can be found in the download area of the "Application Knowledge Base" under **http://www.Lenze.com**

5.10 Index