L-force *Communication*

PROFIBUS-DP

EMF2133IB

Communication module

Lenze

i   Contents

Lenze

i Contents

Lenze

1 About this documentation

Contents

This documentation only contains descriptions for the EMF2133IB communication module (PROFIBUS-DP).

> **Note!**
>
> This documentation supplements the **mounting instructions** supplied with the communication module and the **documentations for the standard devices used**.
>
> **The mounting instructions contain safety instructions which must be observed!**

► The features and functions of the communication module are described in detail.

► Examples illustrate typical applications.

► Furthermore this documentation contains the following:
  – Safety instructions that must be observed.
  – Key technical data relating to the communication module
  – Information on versions of Lenze standard devices to be used.
  – Notes on troubleshooting and fault elimination

The theoretical correlations are only explained in so far as they are necessary for comprehending the function of the function module.

This documentation does not describe the software of an original equipment manufacturer. No responsibility is taken for corresponding information given in this manual. Information on how to use the software can be obtained from the documents of the host system (master).

All brand names mentioned in this manual are trademarks of their respective companies.

Validity information

The information given in this documentation is valid for the following devices:

Communication module	Type designation	From hardware version	From software version
PROFIBUS-DP	EMF2133IB	V2	0x

Target group

This documentation is intended for all persons who plan, install, commission and maintain the networking and remote service of a machine.

> **Tip!**
>
> Information and auxiliary devices around the Lenze products can be found in the download area at
>
> **http://www.Lenze.com**

Lenze

1.1 Document history

Material no.	Version			Description
-	1.0	11/2001	TD06	First edition

-	2.0	06/2004	TD06	● As of software version 1.2: Code C1882 added
				● Complete revision:
				  – Layout change
				  – New German orthography

-	4.0	12/2006	TD17	● As of software version 1.3: code C1883 added ● Structural and editorial adjustments Extended descriptions for the use on the ECS servo system.
13389871	5.0	09/2011	TD14	● Extended information on PROFIBUS DP-V1. ● Structural and editorial adjustments

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

1.2   **Conventions used**

This documentation uses the following conventions to distinguish between different types of information:

Type of information	Identification	Examples/notes
Spelling of numbers		
Decimal separator	Point	In general, the decimal point is used. For instance: 1234.56
Decimal	Standard notation	For example: 1234
Hexadecimal	0x[0 ... 9, A ... F]	For example: 0x60F4
Binary • Nibble	In quotation marks Point	For example: '100' For example: '0110.0100'
Text		
Program name	» «	PC software For example: »Engineer«, »Global Drive Control« (GDC)
Icons		
Page reference	📖	Reference to another page with additional information For instance: 📖 16 = see page 16

1.3   **Terminology used**

Term	Meaning
PROFIBUS	The term stands for the **PROFIBUS-DP** variant according to IEC 61158 / IEC 61784. A different PROFIBUS variant is not described in these Instructions.
Standard device	Lenze controllers/frequency inverters with which the communication module can be used.
Controller	📖 12
Frequency inverter	
Master	PROFIBUS station which takes over the master function in the fieldbus system.
Slave	PROFIBUS station representing a slave in the fieldbus system.
Code	"Container" for one or several parameters used for parameter setting or monitoring of the controller.
Subcode	If a code contains several parameters, they are stored under "subcodes". The documentation uses a slash "/" as a separator between code and subcode (e.g. "C00118/3").
POW	Process output data word
PIW	Process input data word

Lenze

1.4 **Notes used**

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

Safety instructions

Structure of safety instructions:

⚠ **Danger!**

(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
ℹ **Note!**	Important note to ensure troublefree operation
💡 **Tip!**	Useful tip for simple handling
🔖	Reference to another documentation

2   Safety instructions

> 🛈 **Note!**
>
> It is absolutely vital that the stated safety measures are implemented in order to prevent serious injury to persons and damage to material assets.
>
> Always keep this documentation to hand in the vicinity of the product during operation.

2.1   General safety information

> ⚠ **Danger!**
>
> Disregarding the following basic safety measures may lead to severe personal injury and damage to material assets!

► Lenze drive and automation components ...

... must only be used for the intended purpose.

... must never be operated if damaged.

... must never be subjected to technical modifications.

... must never be operated unless completely assembled.

... must never be operated without the covers/guards.

... can - depending on their degree of protection - have live, movable or rotating parts during or after operation. Surfaces can be hot.

► All specifications of the corresponding enclosed documentation must be observed.

This is vital for a safe and trouble-free operation and for achieving the specified product features.

The procedural notes and circuit details provided in this document are proposals which the user must check for suitability for his application. The manufacturer does not accept any liability for the suitability of the specified procedures and circuit proposals.

► Only qualified skilled personnel are permitted to work with or on Lenze drive and automation components.

According to IEC 60364 or CENELEC HD 384, these are persons ...

... who are familiar with the installation, assembly, commissioning and operation of the product,

... possess the appropriate qualifications for their work,

... and are acquainted with and can apply all the accident prevent regulations, directives and laws applicable at the place of use.

2.2 Device- and application-specific safety instructions

▶ During operation, the communication module must be securely connected to the standard device.

▶ With external voltage supply, always use a separate power supply unit, safely separated in accordance with EN 61800-5-1 in every control cabinet ("SELV" / "PELV").

▶ Only use cables that comply with the given specifications (📖 25).

Documentation for the standard device, control system, system/machine

All the other measures prescribed in this documentation must also be implemented. Observe the safety instructions and application notes stated in this manual.

2.3 Residual hazards

Protection of persons

▶ If the controllers are used on a phase earthed mains with a rated mains voltage ≥ 400 V, protection against accidental contact is not ensured without implementing external measures. (See chapter "4.2", 📖 17)

Device protection

▶ The module contains electronic components that can be damaged or destroyed by electrostatic discharge.

3   Product description

3.1   Application as directed

The communication module ...

► is an accessory module which can be used in conjunction with the following Lenze standard devices:

Device type	Design	Version		Variant	Explanation
		HW	**SW**		
82EVxxxxxBxxxXX		Vx	1x		8200 vector
82CVxxxxxBxxxXX		Vx	1x		8200 vector, cold plate
82DVxxxKxBxxxXX		Vx	1x		8200 vector, thermally separated
EPL 10200	E	1x	8x		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	8x	Vxxx	9300 Servo PLC
33.93XX	CI / CT	2x	8x	Vxxx	9300 Servo PLC, Cold plate
ECSxSxxxx4xxxxXX [1]		1A	6.0		ECSxS (Speed and Torque)
ECSxPxxxx4xxxxXX [1]		1A	6.0		ECSxP (Posi and Shaft)
ECSxMxxxx4xxxxXX [1]		1A	6.0		ECSxM (Motion)
ECSxAxxxx4xxxxXX [1]		1A	2.3		ECSxA (Application)

[1] The standard device cannot be used with the DRIVECOM or PROFIdrive control.

► is a device intended for use in industrial power systems.

Any other use shall be deemed inappropriate!

3.2 Identification

9371BC019

		33.2133IB	Vx	0X
Series				
Hardware version				
Software version				
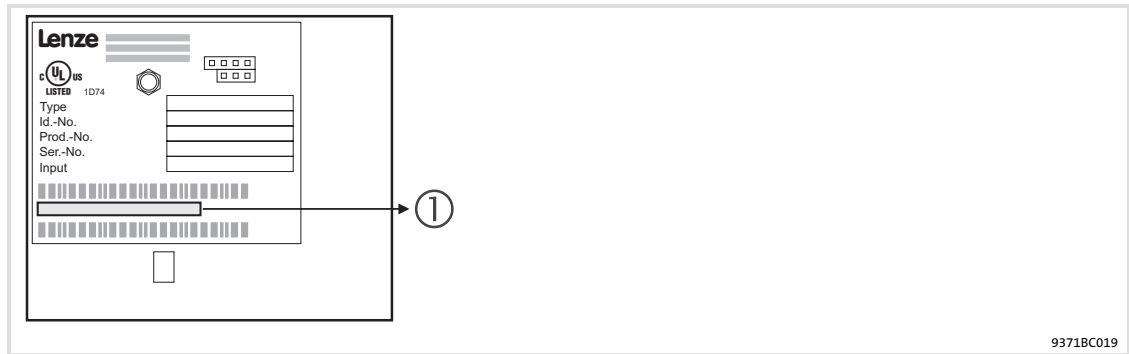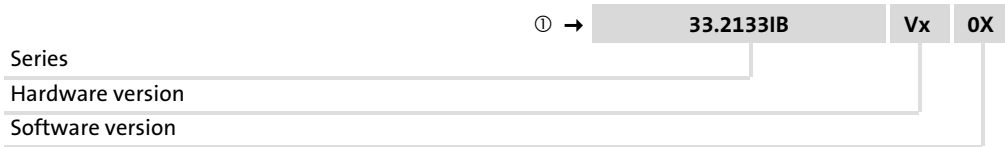
3.3 Product features

▶ Interface module for the PROFIBUS communication system with the PROFIBUS-DP-V0 (DRIVECOM profile) and PROFIBUS-DP-V1 (PROFIdrive) communication profiles

▶ Drive profiles:
– DRIVECOM profile "drive technology 20" (can be switched off)
– PROFIdrive (can be switched off, state machine and PROFIdrive parameter data channel)

▶ Support of the I&M0 functionality for identifying the standard device

▶ Automatic detection of the baud rate (9.6 kbps ... 12 Mbps)

▶ Optionally up to 12 process data words (depending on the basic device)

▶ Acyclic parameter access via DP-V1

▶ Access to all Lenze parameters

▶ External 24V supply for maintaining the PROFIBUS network if the standard device fails

▶ DIP switches for ...
– Setting of the node address
– Setting of the compatibility to the Lenze PROFIBUS communication module EMF2131IB

▶ LED status displays:
– Voltage supply of the communication module
– Connection from the communication module to the PROFIBUS network
– Connection from the communication module to the standard device
– Operating statuses of the standard device

3.4 Connections and interfaces
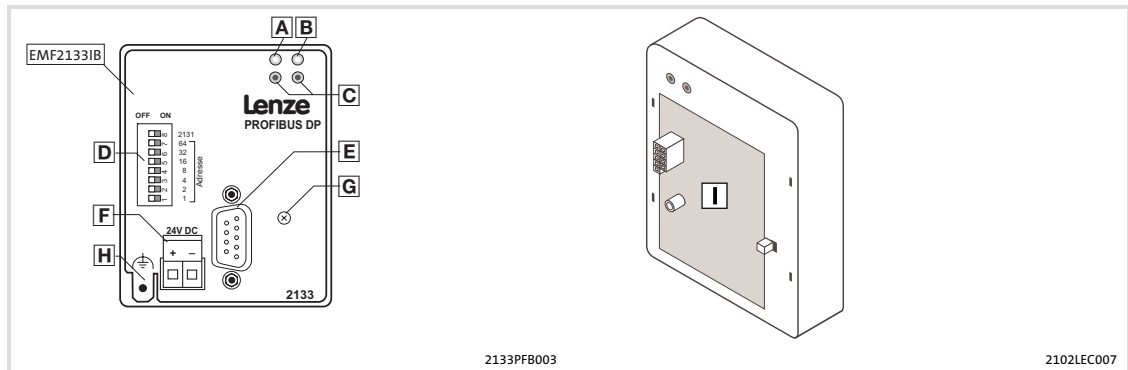
Fig. 3-1 EMF2133IB communication module (PROFIBUS-DP)

Pos.	Description	Detailed information
A	Status of the voltage supply (green LED)	
B	Status of the PROFIBUS communication (yellow LED)	📖 112
C	Operating status of the standard device (red/green LED)	
D	DIP switches for setting the ... ● Compatibility with the PROFIBUS communication module EMF2131IB ● Station address	📖 37 📖 42
E	PROFIBUS connection (Sub-D socket, 9-pole)	📖 24 📖 27
F	Connection for external voltage supply (Plug connector with screw connection, 2-pole)	📖 29
H	PE connection (only with 82XX)	
G	Fixing screw	
I	Nameplate	📖 13

> 🛈 **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.

4 Technical data

4.1 General data

Area	Values
Order designation	EMF2133IB
PNO ID number	2133_{hex}
Communication profile (DIN 19245 Part 1 and Part 3)	● PROFIBUS-DP-V0 ● PROFIBUS-DP-V1
Communication medium	RS485
Interface	9-pin Sub-D socket
Drive profile	● DRIVECOM profile "drive technology 20" (can be switched off) ● PROFIdrive profile (can be switched off, state machine and PROFIdrive parameter data channel)
Network topology	● without repeaters: Line ● with repeaters: Line or tree
PROFIBUS nodes	Slave
Baud rate [kbps]	9.6 ... 12000 (automatic detection)
Process data words	1 ... 12 words (16 bits/word)
DP user data length	1 ... 12 process data words + 4 parameter data words
Max. number of stations	● Standard: 32 (= 1 bus segment) ● with repeater: 125
Max. cable length per bus segment	1200 m (depending on the baud rate and cable type used)
External DC voltage supply	V = +24 V DC ±10 % I = 120 mA

Documentation for Lenze series of devices 8200 vector, 9300 and ECS

Here you can find the **ambient conditions** and the **electromagnetic compatibility (EMC)** specifications applying to the communication module.

4.2 Protective insulation

> ⚠ **Danger!**
>
> **Dangerous electrical voltage**
>
> If Lenze controllers are used on a phase earthed mains with a rated mains voltage ≥ 400 V, protection against accidental contact is not ensured without implementing external measures.
>
> **Possible consequences:**
>
> ► Death or serious injury
>
> **Protective measures:**
>
> ► If protection against accidental contact is required for the control terminals of the controller and the connections of the plugged device modules, ...
>   – a double isolating distance must exist.
>   – the components to be connected must be provided with the second isolating distance.

Insulation between bus and ...	Type of insulation (in accordance with EN 61800-5-1)
● Reference earth / PE	Functional insulation
● External supply	Functional insulation
● Power section	
– 820X / 821X	Basic insulation
– 822X / 8200 vector	Reinforced insulation
– Drive PLC	Reinforced insulation
– 93XX / 9300 Servo PLC	Reinforced insulation
– ECS servo system	Reinforced insulation
● Control terminals	
– 820X / 8200 vector	Functional insulation
– 821X	Functional insulation
– 822X	Basic insulation
– Drive PLC	Basic insulation
– 93XX / 9300 Servo PLC	Basic insulation
– ECS servo system	Reinforced insulation

4.3 Communication time

The communication time is the time between the start of a request and the arrival of the corresponding response.

The communication times depend on ...

► the processing time in the controller

► the transmission delay time
   – the baud rate
   – the telegram length

4.3.1 Processing time 820X

For the 820X series several processing steps are required in the controller, which are processed cyclically.

A processing cycle consists of:

► Writing of control word or setpoint if the value has changed;

► Alternating reading of status word and actual value;

► Processing of parameter accesses if there is a job.

If the processing time caused by cyclic reading of the status word/actual value is too large, the alternating reading of status word and actual value can be suppressed. This is controlled by bit 15 (process input data inhibit) of the DRIVECOM control word:

► Process input data inhibit = 0: Status and actual value update active

► Process input data inhibit = 1: Status and actual value update not active

A suppression of the processing of parameter accesses is not necessary, since this is controlled by the user.

In the following table the times for the processing steps are listed:

Processing step	Max. processing time in [ms]			
	Process input data inhibit = 0	Tolerance	Process input data inhibit = 1	Tolerance
Read parameter	55	+48	55	+8
Control word or setpoint	27	+48	27	+8
Control word and setpoint	54	+56	54	+16
Write parameter	108	+32	-	-
Status word and actual value	200	+40	200	-

Note!

A setpoint sign reversal also results in writing the control word.

Lenze

4.3.2 Processing time 821X / 822X / 824X / 8200 vector

Parameter data	Process data
30 ... 50 ms	2 ... 3 ms

4.3.3 Processing time 93XX / ECSxS

There are no interdependencies between parameter data and process data.

Parameter data	Process data
Approx. 30 ms + 20 ms tolerance (typical) For some codes, the processing time can be longer (see documentation for 9300 and ECS servo system).	2 ms + 1 ms tolerance

4.3.4 Processing time Drive PLC / 9300 Servo PLC / ECSxA

Parameter data	Process data
Approx. 30 ms + 20 ms tolerance (typical) For some codes, the processing time can be longer (see documentation for 9300 and ECS servo system).	Depending on the process image

Lenze

4.4 Dimensions
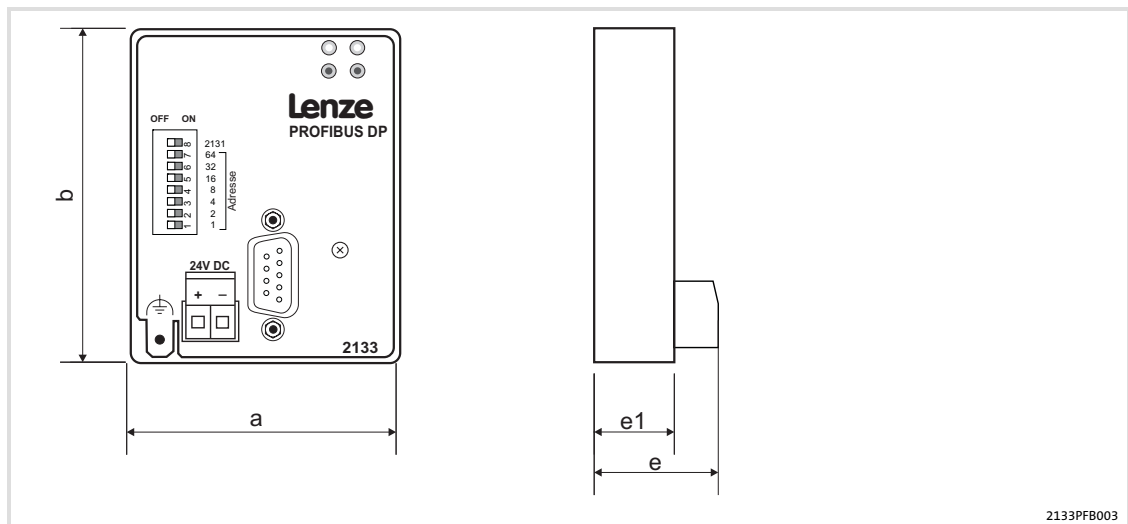
2133PFB003

a	61 mm
b	75 mm
e	28 mm
e1	18 mm

5   Installation

⚠️ **Danger!**

Inappropriate handling of the communication module and the standard device can cause serious personal injury and material damage.

Observe the safety instructions and residual hazards described in the documentation for the standard device.

🛑 **Stop!**

Electrostatic discharge

Electronic components of the communication module can be damaged or destroyed through electrostatic discharge.

Possible consequences:

▶ The communication module is damaged.

▶ Fieldbus communication is not possible or faulty.

Protective measures

▶ Discharge electrostatic charges before touching the module.

5.1 Mechanical installation
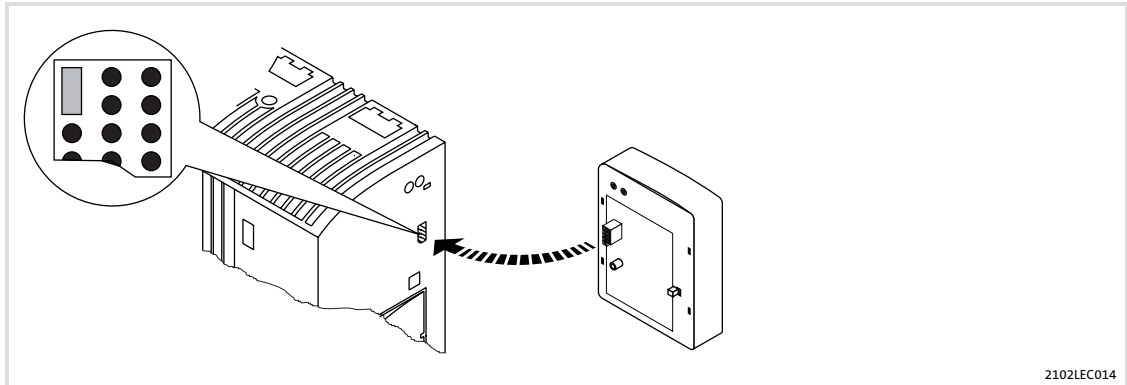
Fig. 5-1   Attaching the communication module

▶ Plug the communication module onto the standard device (here: 8200 vector).

▶ Tighten the communication module to the standard device using the fixing screw in order to ensure a good PE connection.

> **Note!**
>
> For the internal supply of the communication module by the 8200 vector frequency inverter the jumper has to be adjusted within the interface opening (see illustration above).
>
> Observe the notes (📖 28).

5.2 **Electrical installation**

5.2.1 **Wiring according to EMC (CE-typical drive system)**

For wiring according to EMC requirements observe the following points:

> **Note!**
> ► Separate control cables/data lines from motor cables.
> ► Connect the shields of control cables/data lines *at both ends* in the case of digital signals.
> ► Use an equalizing conductor with a cross-section of at least 16 mm^2 (reference: PE) to avoid potential differences between the bus nodes.
> ► Observe the other notes concerning EMC-compliant wiring given in the documentation for the standard device.

Wiring procedure

1. Comply with bus topology, thus do not use stubs.

2. Observe notes and wiring instructions in the documents for the control system.

3. Only use cables that comply with the given specifications (25).

4. Observe notes for the voltage supply of the module (28).

5. Activate the bus terminating resistors on the first and last physical bus device (24).

6. Adapt baud rate to the bus cable length.

5.2.2 Wiring with a host (master)

> ⚡ **Danger!**
>
> You have to provide additional electrical isolation if ...
> ► an 820X and 821X controller is connected to the host and
> ► a safe electrical isolation (reinforced insulation) according to EN 61800-5-1 is required.

Basic wiring of the PROFIBUS

The connection of the PROFIBUS bus system is shown in the general layout drawing.

E82ZAFP005
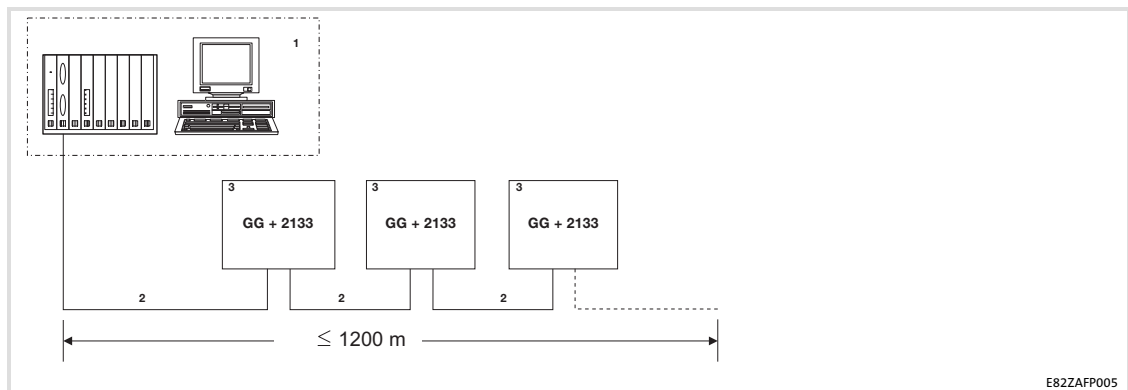
Fig. 5-2   Example: PROFIBUS with RS485 wiring (without repeater)

No.	Element	Comment
1	Host	e.g. PC or PLC with PROFIBUS master interface module
2	Bus cable	Connects the PROFIBUS master interface module to the communication modules. ● The baud rate depends on the bus cable length (□ 26).
3	PROFIBUS slave	Applicable standard device (GG, □ 12) with communication module ● Activate the bus terminating resistors on the first and last physical bus device (□ 24).

> ℹ️ **Note!**
>
> When using a repeater, max. 125 devices can communicate via the PROFIBUS.

Bus terminating resistor

The PROFIBUS must be terminated by a bus terminating resistor at the physically first and last station.

The bus terminating resistor is in the bus connector (□ 128)and is activated using a switch.

> ℹ️ **Note!**
>
> ► If you want to disconnect individual bus devices, ensure that the bus terminators at the cable ends remain active.
> ► Please note that the bus termination is no longer active if ...
> – the connector has been disconnected e.g. in service case;
> – the voltage supply of the communication module has been switched off.

Number of bus devices

2133PFB004
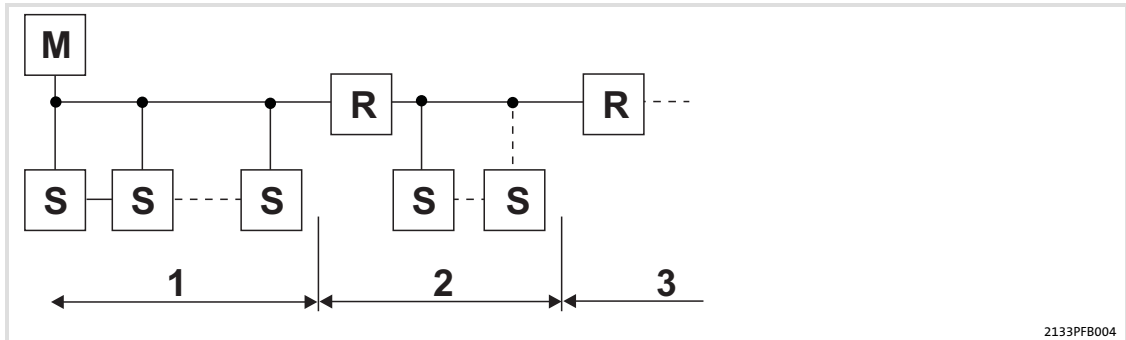
Segment	Master (M)	Slave (S)	Repeater (R)
1	1	31	-
	2	30	-
2	-	30	1
3	-	30	1

Tip!

Repeaters do not have a device address. When calculating the maximum number of bus devices, they reduce the number of devices by 1 on each side of the segment.

Repeaters can be used to build up line and tree topologies. The maximum total bus system expansion depends on ...

▶ the baud rate used;

▶ the number of repeaters used.

Specification of the transmission cable

Note!

Only use cables complying with the listed specifications of the PROFIBUS user organisation.

Field	Values
Specific resistance	135 ... 165 Ω/km, (f = 3 ... 20 MHz)
Capacitance per unit length	$\leq$ 30 nF/km
Loop resistance	< 110 Ω/km
Core diameter	> 0.64 mm
Core cross-section	> 0.34 mm^2
Cores	Twisted double, insulated and shielded

Bus cable length

The length of the bus cable depends on the baud rate used:

Baud rate [kbps]	Length [m]
9.6 ... 93.75	1200
187.5	1000
500	400
1500	200
3000 ... 12000	100

Note!

The baud rate depending on the data volume, cycle time, and number of nodes should only be selected as high as required for the application.

Tip!
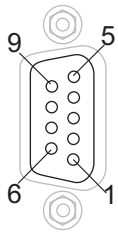
For high baud rates we recommend to consider the use of optical fibres.

Advantages of optical fibres:

► On the transmission path external electromagnetic interference remains ineffective.
► Bus lengths of several kilometres are also possible with higher baud rates. The bus length
   – is irrespective of the baud rate.
   – depends on the optical fibre used.

5.2.3   Connection of the PROFIBUS

The PROFIBUS network is connected via the 9-pole Sub-D socket.

View	Pin	Designation	Description
	1	-	-
	2	-	-
	3	RxD/TxD-P	Data cable B (receive / send data plus)
	4	RTS	Request To Send (receive / send data, no differential signal)
	5	M5V2	Data reference potential (ground to 5V)
	6	P5V2	5 V DC / 30 mA (bus termination)
	7	-	-
	8	RxD/TxD-N	Data cable A (receive / send data minus)
	9	-	-

5.2.4   Voltage supply

Internal voltage supply

> **ℹ️ Note!**
>
> Internal voltage supply has been selected in the case of standard devices with an extended AIF interface opening (e.g. front of 8200 vector). The area shown on a grey background in the graphic marks the jumper position.
>
> ► By default, this is **not** supplied internally in the standard device.
>
> ► For internal voltage supply place the jumper on the position indicated below.
>
> In the case of all other device series (9300, ECS), voltage is always supplied from the standard device.

Lenze setting (Only external voltage supply possible.)	Internal voltage supply
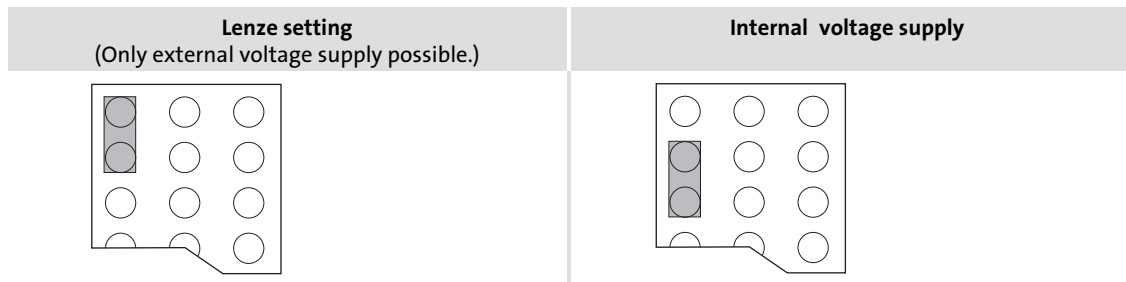
External voltage supply

> **🛈 Note!**
>
> Always use a separate power supply unit in every control cabinet and safely separate it according to EN 61800-5-1 ("SELV"/"PELV") in the case of external voltage supply and larger distances between the control cabinets.

External voltage supply of the communication module is required if communication via the fieldbus is to be maintained even when the power supply of the standard device fails.
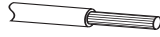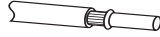
> **🛈 Note!**
>
> With external voltage supply of the communication module, the active bus terminating resistor is fed independently of the operation of the basic device. Thus the bus system remains active even if the basic device is switched off or fails.

Plug connector	Explanation
"+"	V = 24 V DC (21.6 V - 0% ... 26.4 V + 0 %) I = 120 mA
"-"	Reference potential for external voltage supply

Controller	External voltage supply
820X	Always required
821X / 822X / 824X / 93XX / 9300 Servo PLC / Drive PLC / ECSxS / ECSxP / ECSxA	Only required if the mains supplying the corresponding controller is to be switched off but communication must not be interrupted. For these basic devices the internal voltage supply can be used.
8200 vector	See notes given in "Internal voltage supply" 📖 28

5.2.5 **Cable cross-sections and screw-tightening torques**

Area	Values	
Electrical connection	Plug connector with 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)	
Stripping length	6 mm	

6 Commissioning

During commissioning, system-dependent data as e.g. motor parameters, operating parameters, responses and parameters for fieldbus communication are selected for the controller.

In Lenze devices, this is done via codes. The codes are stored in numerically ascending order in the Lenze controllers and in the plugged-in communication/function modules.

In addition to these configuration codes, there are codes for diagnosing and monitoring the bus devices.

The codes can be set e.g. via an operating module (keypad) or a PC with the Lenze parameter setting program »Global Drive Control« (GDC).

6.1 Before switching on

> (STOP) **Stop!**
>
> Prior to switching on the mains voltage, check the wiring for completeness, short-circuit and earth fault.

6.2 Initial switch-on

> **Note!**
>
> **ECS servo system**
> ECS devices cannot be used with the DRIVECOM or PROFIdrive control.

> **Note!**
>
> Manual settings are not required for the baud rate. The communication module is automatically adjusted to the baud rate of the master.

Step-by-step commissioning of the communication module with DRIVECOM device control is described below.

| Step	Procedure	Detailed information
1.	Select process data communication with DRIVECOM profile in the configuration software of the PROFIBUS master. Example: "Par(kons)+3PZD"	
2.	Configure host system for communication with the EMF2133IB communication module.	📖 34
3.	Inhibit standard device via terminal.	Documentation of the standard device
4.	Check bus termination. ● The PROFIBUS must be terminated by a bus terminating resistor at the physically first and last station. ● The bus terminating resistor is integrated into the bus connector and can be activated via a switch.	📖 37
5.	Provide software compatibility with the communication module. ● 2133: DIP switch S8 = OFF ● 2131: DIP switch S8 = ON (with this setting, continue commissioning for the EMF2131IB communication module.) **Lenze setting: S8 = OFF**	📖 37
6.	Drive-specific settings.	Documentation of the standard device
7.	Prepare controller for communication.	📖 38
8.	Switch on the mains voltage for the controller and, if available, the separate voltage supply for the communication module. **Response** The green bus LED on the front of the communication module comes on.	📖 44

9.	A Set station address via ...
	   – Standard device code C0009,
	   – DIP switch S1 ... S7 or
	   – define through a master (class 2).
	
	● In the PROFIBUS network, every station needs its own address.
	● Valid address range: 3 ... 126
	● If the settings via code apply (DIP switches S1 ... S7 = OFF), the address has to be newly assigned after a parameter set transfer.
	● The address modified via keypad becomes effective immediately.
	
	B Switch off the voltage supply of the function module and the standard device and then switch it on again to accept the changed settings.
		📖 42

10.	Manual settings are not required for the baud rate. The communication module is automatically adjusted to the baud rate of the master.	

Lenze

Step	Procedure	Detailed information
11.	It is now possible to communicate with the controller, i.e. • exchange process data (setpoints and actual values); • read all codes; • change all codes that can be written. See the attribute table or code description of the corresponding standard device. **Response** The yellow LED on the communication module is blinking when the PROFIBUS is active.	
12.	Enable standard device via terminal.	Documentation of the standard device

6.3 **Configuring the host system (master)**

The host must be configured before communication with the communication module is possible.

Master settings

For configuring the PROFIBUS, the device data base file (GSE file) of the communication module has to be imported into the configuring software of the master.

💡 **Tip!**

The GSE file can be downloaded in the "Services & Downloads" area at www.Lenze.com.

Device data base file (GSE)

The following configurations can be found in the device data base files **Lenz2133.GSD** (DP-V0) and **Len_2133.GSD** (DP-V1):

► Device control and DP-V0 parameter data channel

Selection text in Lenz2133.GSE	Parameter data		Process data		Assigned I/O memory
	without consistency	with consistency	without consistency	with consistency	
PAR(cons.)+PZD(nwordsI/O)AR		•	n words		4 + n words
PAR(cons.)+PZD(nwordscon)AR				n words	4 + n words
PAR + PZD(n words I/O) AR	•		n words		4 + n words
PAR + PZD(n words con) AR				n words	4 + n words
PZD(n words I/O) AR	Without parameter data channel		n words		n words
PZD(n words cons.) AR				n words	n words

n = 1 ... 12

► DRIVECOM control and DP-V0 parameter data channels

Selection text in Lenz2133.GSE	Parameter data		Process data		Assigned I/O memory
	without consistency	with consistency	without consistency	with consistency	
PAR(cons.) + PZD(n words I/O)		•	n words		4 + n words
PAR(cons.) + PZD(n words cons.)				n words	4 + n words
PAR + PZD(n words I/O)	•		n words		4 + n words
PAR + PZD(n words cons.)				n words	4 + n words
PZD(n words I/O)	Without parameter data channel		n words		n words
PZD(n words cons.)				n words	n words

n = 1 ... 12

▶ POFIdrive control and DP-V1 parameter data channel

Selection text in Len_2133.GSE	Parameter data		Process data		Assigned I/O memory
	without consistency	with consistency	without consistency	with consistency	
PPO1			2 words		6 words
PPO2		•	6 words		10 words
PPO5			10 words		14 words
PPO3	Without parameter data channel		2 words		2 words
PPO4			6 words		6 words
PPO1 (process data consistency)				2 words	6 words
PPO2 (process data consistency)		•		6 words	10 words
PPO5 (process data consistency)				10 words	14 words
PPO3 (process data consistency)	Without parameter data channel			2 words	2 words
PPO4 (process data consistency)				6 words	6 words

n = 1 ... 12

Example of the selection text of the device data base file

PAR (Cons) + PCD (7W) AR

Lenze device control

Process data words
(7 words)

Parameter data channel
(4 bytes consistent)

PAR (Cons) + PCD (8W)

Without "AR": Control with DRIVECOM-Profil

Process data words
(8 words)

Parameter data channel
(4 bytes consistent)
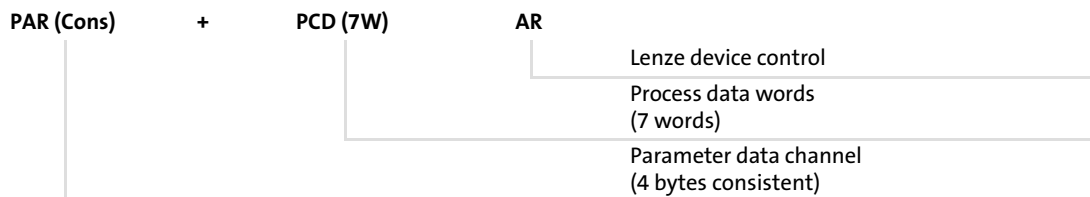
ⓘ **Note!**

Use overall consistency

▶ We recommend to exclusively use configurations with consistency for the parameter data channel to avoid data conflicts between the PROFIBUS master and the host CPU.

▶ Please note that the processing of consistent data varies between hosts. This must be considered in the PROFIBUS application program.

▶ Detailed information on consistency can be found on 📖 110.

Defining the user data length

The user data length is defined during the initialisation phase (configuration). Up to 12 process data words can be configured (depending on the basic device used).

Optionally you can activate the parameter data channel. If the parameter data channel is active, it additionally occupies 4 words of the process data inputs and outputs.

▶ PIW: Process data input word (process data from the controller to the master)

▶ POW: Process data output word (process data from the master to the controller)

The user data lengths for process input data and process output data are identical. The selection takes place via identification bytes in the configuration software for the PROFIBUS system.

Parameter data channel		Process data channel
Without / with	**Identification / user data length**	**Identification / user data length**
Without	-	• Identification – without consistency: 70_{hex} ... $7B_{hex}$ (112 ... 123) – with consistency: $F0_{hex}$ ... FB_{hex} (240 ... 251) • User data length: 1 ... 12 words (PAW1/PEW1 ... PAW12/PEW12)
With	• Identification – without consistency: 73_{hex} (115) – with consistency: $F3_{hex}$ (243) • User data length: 4 words (Word rt 1 ... word 4)	• Identification – without consistency: 70_{hex} ... $7B_{hex}$ (112 ... 123) – with consistency: $F0_{hex}$ ... FB_{hex} (240 ... 251) • User data length: 1 ... 12 words (PAW1/PEW1 ... PAW12/PEW12)

General structure of the identification byte

MSB							LSB
7	6	5	4	3	2	1	0

User data length
00 1 byte or 1 word
...
15 16 bytes or 16 words

Input/Output
00 Specific identification format
01 Input
10 Output
11 Input and output

Length/Format
0 Byte
1 Word

Consistency
0 Byte or word
1 Total length
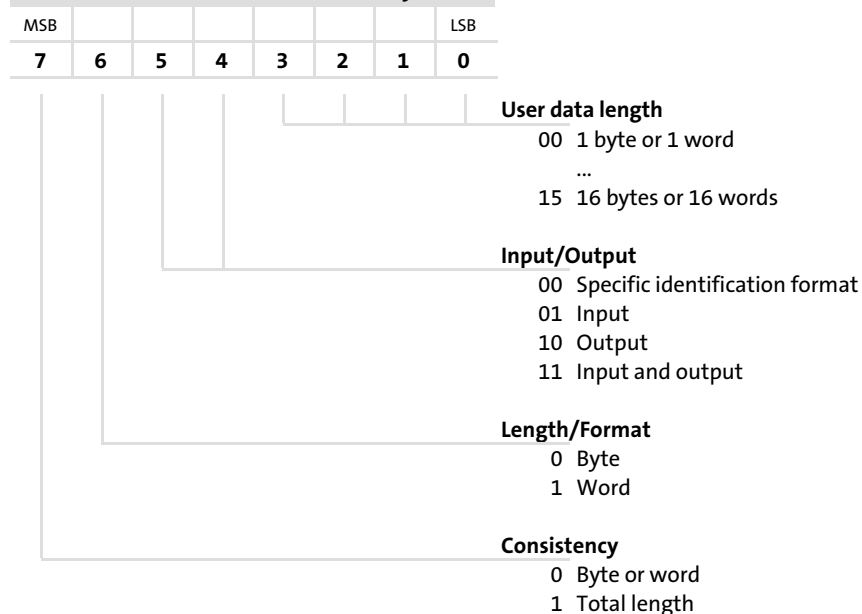
6.4   **Activating the bus terminating resistor**

Bus terminating resistor

The PROFIBUS must be terminated by a bus terminating resistor at the physically first and last station.

The bus terminating resistor is in the bus connector (📖 128)and is activated using a switch.

> 🛈 **Note!**
>
> ▶ If you want to disconnect individual bus devices, ensure that the bus terminators at the cable ends remain active.
> ▶ Please note that the bus termination is no longer active if ...
>   – the connector has been disconnected e.g. in service case;
>   – the voltage supply of the communication module has been switched off.

6.5   **Setting the software compatibility**

> 🛈 **Note!**
>
> If the EMF2131IB communication module is replaced by the EMF2133IB communication module, ...
> ▶ do not change any host settings;
> ▶ set the DIP switch **S8** to the "ON" position.

6   **Commissioning**
            Preparing the standard device for communication
            Frequency inverter 82XX / 8200 vector

6.6   Preparing the standard device for communication

6.6.1   Frequency inverter 82XX / 8200 vector

Step	Procedure	Detailed information
1.	In order that you can operate the controller via PROFIBUS, set the Lenze parameter "Operating mode" C0001 = 3.	📖 46
	Example of PROFIBUS Write: ● C0001=3 ● Index = 0x5FFE (resulting from 0x5FFF - C0001$_{hex}$) ● Subindex: 0 ● Value: 30000 (resulting from 3×10^4)	Documentation of the standard device
2.	Terminal 28 (RFR = controller enable) is always active and must be set to HIGH level during PROFIBUS operation. Otherwise the controller cannot be enabled by PROFIBUS (DRIVECOM device status "OPERATION ENABLED"). **Note** In case of 821X, 822X and 8200 vector, the quick stop function (QSP) is always active. If QSP is configured to an input terminal (Lenze setting: Not assigned), it has to be on HIGH level during PROFIBUS operation.	
3.	The controller can now accept control and parameter setting data via the PROFIBUS.	
4.	Select speed setpoint unequal to 0.	📖 47
5.	Change to status "READY TO SWITCH ON". Select value for DRIVECOM control word: 0b0000 0000 0111 1110 (0x007E).	
6.	Wait for status "READY TO START" to be reached. Value for DRIVECOM status word: 0bxxxx xxxx x01x 0001.	
7.	Change to the "OPERATION ENABLED" state. Select value for DRIVECOM control word: 0b0000 0000 0111 1111 (0x007F)	
8.	Wait for "OPERATION ENABLED".	📖 68

Commissioning 6
Preparing the standard device for communication
93XX servo inverter / 9300 Servo PLC

6.6.2 93XX servo inverter / 9300 Servo PLC

Step	Procedure		Detailed information
1.	**93XX**	In order that you can operate the controller via PROFIBUS, set the Lenze parameter "Signal configuration" C0005 = xxx3. ● When commissioning for the first time, we recommend to select the signal configuration "1013" (speed control). Example of PROFIBUS Write: ● C0005=1013 (speed control) ● Index = 0x5FFA (resulting from 0x5FFF - C0005$_{hex}$) ● Subindex: 0 ● Value: 10130000 (resulting from 1013 x 10^4)	📖 46 Documentation of the standard device
	9300 servo PLC	Implement the system blocks AIF-IN1 ... 3, AIF-OUT1 ... 3 and, if available, the AIF management into the control configuration of the IEC61131 project.	
2.		Terminal 28 (RFR = controller enable) is always active and must be set to HIGH level during PROFIBUS operation. Otherwise the controller cannot be enabled by PROFIBUS (DRIVECOM device status "OPERATION ENABLED").	

		Note
		● For the signal configuration C0005 = 1013 (speed control), the quick stop function (QSP) in connection with the right/left change-over is assigned to the digital input terminals E1 and E2 and thus always active. For PROFIBUS operation, E1 must be assigned to HIGH level.
		● With the signal configuration C0005 = xx13, the terminal A1 is switched as voltage output. This means that only the following terminals should be connected:
		  − X5.A1 with X5.28 (RFR)
		  − X5.A1 with X5.E1 (CW/QSP)	

3.		The controller can now accept control and parameter setting data via the PROFIBUS.	
4.		Select speed setpoint unequal to 0.	📖 47
5.		Change to status "READY TO SWITCH ON". Select value for DRIVECOM control word: 0b0000 0000 0111 1110 (0x007E).	
6.		Wait for status "READY TO START" to be reached. Value for DRIVECOM status word: 0bxxxx xxxx x01x 0001.	
7.		Change to the "OPERATION ENABLED" state. Select value for DRIVECOM control word: 0b0000 0000 0111 1111 (0x007F)	
8.		Wait for "OPERATION ENABLED".	📖 68

6.6.3 Drive PLC

Step	Procedure	Detailed information
1.	Implement the system blocks AIF-IN1 ... 3, AIF-OUT1 ... 3 and, if available, the AIF management into the control configuration of the IEC61131 project.	📖 46
2.	The controller can now accept control and parameter setting data via the PROFIBUS.	Documentation of the standard device
3.	Select speed setpoint unequal to 0.	📖 47
4.	Change to status "READY TO SWITCH ON". Select value for DRIVECOM control word: 0b0000 0000 0111 1110 (0x007E).	
5.	Wait for status "READY TO START" to be reached. Value for DRIVECOM status word: 0bxxxx xxxx x01x 0001.	
6.	Change to the "OPERATION ENABLED" state. Select value for DRIVECOM control word: 0b0000 0000 0111 1111 (0x007F)	
7.	Wait for "OPERATION ENABLED".	📖 68

6.6.4 Axis modules ECSxS / ECSxA

Step	Procedure		Detailed information
1.	**ECSxS**	Set the Lenze parameter "Control mode": ● C3005 = 1003 (setpoint via AIF, speed-controlled) ● C3005 = 4003 (setpoint via AIF, torque-controlled) Example of PROFIBUS Write: ● C3005=1003 (speed control) ● Index = 0x5442 (resulting from 0x5FFF - C3005$_{hex}$) ● Subindex: 0 ● Value: 10030000 (resulting from 1003 x 10^4)	📖 46 Documentation of the standard device
	ECSxA	Implement the system blocks AIF-IN1 ... 3, AIF-OUT1 ... 3 and, if available, the AIF management into the control configuration of the IEC61131 project.	
2.	The terminals SI1 (controller enable) and SI2 (pulse inhibit) are always active and must be assigned to HIGH level during PROFIBUS operation. Otherwise the controller cannot be enabled by PROFIBUS.		
3.	The controller can now accept control and parameter setting data via the PROFIBUS.		

> **ℹ️ Note!**
>
> **ECS servo system**
>
> ECS devices cannot be used with the DRIVECOM or PROFIdrive control.

6 **Commissioning**
Setting the node address
Setting via code

6.7 Setting the node address

> **i** **Note!**
> ► The addresses of all controllers connected to the network must differ from each other.
> ► If the DIP switches **S1 ... S7** are in the OFF position, the code setting for the station address is active (Lenze setting).
> ► Switch off the voltage supply of the function module and the controller and then switch it on again to activate the changed settings.

The setting of the station address can be freely selected ...

► via the front DIP switches **S1 ... S7**;

► via the standard device code **C0009**;

► through a master (class 2).

Valid address range: 3 … 126
(Lenze setting: 126, provided that C0009 = 1)

6.7.1 Setting via code

► DIP switches **S1 ... S7** = OFF (Lenze setting)

► Set the node address via the standard device code **C0009** (e.g. via keypad or »Global Drive Control« (GDC)).

6.7.2 Settings via DIP switch

Set the node address with the DIP switches **S1 ... S7**.

The sum of valencies makes the station address to be set:

DIP switch	Valency	Example	
		Switch position	**Node address**
S1	1	**ON**	
S2	2	OFF	
S3	4	OFF	
S4	8	OFF	1 + 16 + 32 + 64 = 113
S5	16	**ON**	
S6	32	**ON**	
S7	64	**ON**	

Lenze

6.7.3   Settings by a master (class 2)

► With this method only one device must be connected to the bus. This can be achieved by a special switch-on sequence.

► In the "Power On" status, the master (class 2) can set a device address via the "Set_Slave_Address" telegram.

► Settings made through the master (class 2 only) have an effect on the setting in standard device code C0009.

PROFIBUS station address	Mapping to code C0009
1 ... 2	No (master addresses)
3 ... 99	Yes (3 ... 99)
100 ... 125	Yes (C0009 = 2)
126 (**LENZE setting**)	Yes (**C0009 = 1**)

Tab. 6-1   Assignment of station addresses to controllers

6   **Commissioning**
Connecting the mains voltage
Settings by a master (class 2)

6.8   **Connecting the mains voltage**

> 🛈 **Note!**
>
> If you use the external voltage supply for the communication module, please switch it on.

The following LEDs at the front of the communication module must be on:

► The top green LED (Status display of voltage supply)

► The bottom green LED (status display of standard device)

Protection against uncontrolled restart

> 🛈 **Note!**
>
> **Establishing communication**
>
> If communication is to be established via an externally supplied communication module, initially the standard device must also be switched on.
>
> After communication has been established, the externally supplied module is independent of the power on/off state of the standard device.
>
> **Protection against uncontrolled restart**
>
> After a fault (e.g. short-term mains failure), a restart of the drive is not always wanted and - in some cases - even not allowed.
>
> The restart behaviour of the controller can be set in C0142:
>
> ► C0142 = 0 (Lenze setting)
>   – The controller remains inhibited (even if the fault is no longer active).
>   – The drive starts up in a controlled manner by explicit controller enable:
>   93XX: Set terminal 28 to HIGH level.
>   ECSXX: Set terminals X6/SI1 and X6/SI2 to HIGH level.
>
> ► C0142 = 1
>   – An uncontrolled restart of the drive is possible.

7 Process data transfer
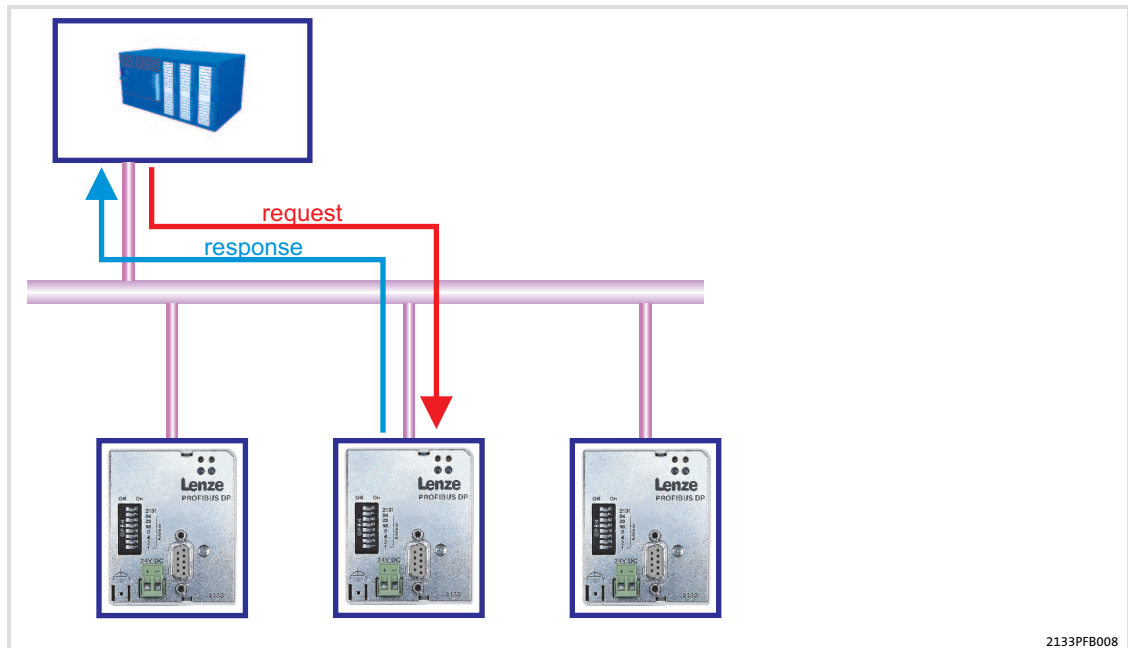
Fig. 7-1   PROFIBUS process data transfer

PROFIBUS transmits parameter data and process data between the host (master) and the controllers connected to the bus (slaves). Depending on their time-critical nature, the data are transmitted via different communication channels.

► Process data are transmitted via the process data channel.

► Process data serve to control the drive controller.

► The transmission of process data is time-critical.

► Process data are cyclically transferred between the host and the controllers (continuous exchange of current input and output data).

► The host can directly access the process data. In the PLC, for instance, the data are directly assigned to the I/O area.

► With the function module a maximum of 10 process data words (16 bits/word) can be exchanged in each direction.

► Process data are not stored in the controller.

► Process data are, for instance, setpoints, actual values, control words and status words.

> ⓘ **Note!**
>
> Observe the direction of the information flow!
> ► Process input data (Rx data):
>   – Process data from controller (slave) to host (master)
> ► Process output data (Tx data):
>   – Process data from host (master) to controller (slave)

7   **Process data transfer**
Lenze device control
Setpoint source selection

7.1   Lenze device control

7.1.1   Setpoint source selection

> ### 🛈   Note!
>
> Note that the selection of the setpoint source must be set the same in all parameter sets.

82XX / 8200 vector frequency inverters

For these controllers the setpoint source selection is determined under code C0001. An evaluation of process data is only possible if code C0001 is set to "3" when the controller is operated together with the communication module (selection: Process data channel of a communication module). The process data channel which defines the frequency setpoint (mapping to C0046) is the *setpoint source* and the control word (C0135).

In case of the 8200 vector, the assignment of the setpoint source to the corresponding analog signal can be checked or changed in code C0412.

93XX controller

For operation via PROFIBUS, code C0005 must be set to the value "xxx3" (x = wildcard for selected preconfiguration).

Example: C0005 = 1013: "Speed control" preconfiguration

ECSxS axis module

For operation via the PROFIBUS, code C3005 "Control mode" must be set:

► C3005 = 1003 (setpoint via AIF, speed-controlled)

► C3005 = 4003 (setpoint via AIF, torque-controlled)

Servo PLC 9300 / Drive PLC / ECSxA

Operation via the PROFIBUS requires that the system blocks AIF-IN1 ... 3, AIF-OUT1 ... 3 and, if available, the AIF management are part of the control configuration of the IEC61131 project.

► For cyclic process data telegrams *to* the drive , the AIF-IN1 ... 3 system blocks are used. The control word (byte 1 and byte 2) contained in a process data telegram is further processed via these system blocks in the standard device.

► For cyclic process data telegrams *from* the drive , the system blocks AIF-OUT1 ... 3 are used. The status word (byte 1 and byte 2) contained in the process data telegram is transmitted to the master via these system blocks .

7.1.2 Process data signals for 82XX frequency inverters

Process data telegram from drive

Byte 1	Byte 2	Byte 3	Byte 4
Status word		Actual value	
High byte	Low byte	High byte	Low byte

ℹ️ **Note!**

► Frequency and speed values are scaled with $\pm 24000 \equiv \pm 480$ Hz.

► Torque values are scaled with $16384 \equiv 100\%$.

7 **Process data transfer**
Lenze device control
Process data signals for 82XX frequency inverters

Device status word AIF-STAT for 82XX (C0150, I-5F69)

820X		821X / 822X / 824X	
Bit	**Assignment**	**Bit**	**Assignment**
0	**Current parameter set**	**0**	**Current parameter set**
0	Parameter set 1 or 3 active	0	Parameter set 1 or 3 active
1	Parameter set 2 or 4 active	1	Parameter set 2 or 4 active
1	**Pulse inhibit (IMP)**	**1**	**Pulse inhibit (IMP)**
0	Pulses for power stage enabled	0	Pulses for power stage enabled
1	Pulses for power stage inhibited	1	Pulses for power stage inhibited
2	I_{max} (current limit reached)	**2**	I_{max} (current limit reached)
0	Current limit not reached	0	Current limit not reached
1	Current limit reached	1	Current limit reached
3	**Not assigned**	**3**	$f_d = f_{dset}$
		0	$f_d \neq f_{dset}$
		1	$f_d = f_{dset}$
4	$f_d = f_{dset}$	**4**	**Ramp function generator (RFG) on/off**
0	$f_d \neq f_{dset}$	0	RFG-On $\neq$ RFG-Off
1	$f_d = f_{dset}$	1	RFG on = RFG off
5	**Qmin** ($f_d \leq f_{dQmin}$)	**5**	**Qmin** ($f_d \leq f_{dQmin}$)
0	Qmin not active	0	Qmin not active
1	Qmin active	1	Qmin active
6	$f_d = 0$ (actual frequency value = 0)	**6**	$f_d = 0$ (actual frequency value = 0)
0	$f_d \neq 0$	0	$f_d \neq 0$
1	$f_d = 0$	1	$f_d = 0$
7	**Controller inhibit (CINH))**	**7**	**Controller inhibit (CINH))**
0	No controller inhibit	0	No controller inhibit
1	Controller inhibit active	1	Controller inhibit active

820X					821X / 822X / 824X					
8 ... 11	**Device status**				**8 ... 11**	**Device status**				
Bit	11	10	9	8	Bit	11	10	9	8	
	0	0	0	0	Device initialisation		0	0	0	0	Device initialisation
	1	0	0	0	Active fault		0	0	1	0	Switch-on inhibit
						0	0	1	1	Operation inhibited
						0	1	0	0	Flying restart circuit active
						0	1	0	1	DC injection brake active
						0	1	1	0	Operation enabled
						0	1	1	1	Message active
						1	0	0	0	Active fault
						1	1	1	1	Communication with standard device not possible

820X		821X / 822X / 824X	
12	**Overtemperature warning**	**12**	**Overtemperature warning**
0	No warning	0	No warning
1	Warning	1	Warning
13	U_{Gmax} (DC bus overvoltage)	**13**	U_{Gmax} (DC bus overvoltage)
0	No overvoltage	0	No overvoltage
1	Overvoltage	1	Overvoltage
14	**Direction of rotation**	**14**	**Direction of rotation**
0	CW rotation	0	CW rotation
1	CCW rotation	1	CCW rotation
15	**Ready for operation**	**15**	**Ready for operation**
0	Not ready for operation	0	Not ready for operation
1	Ready for operation	1	Ready for operation

Lenze
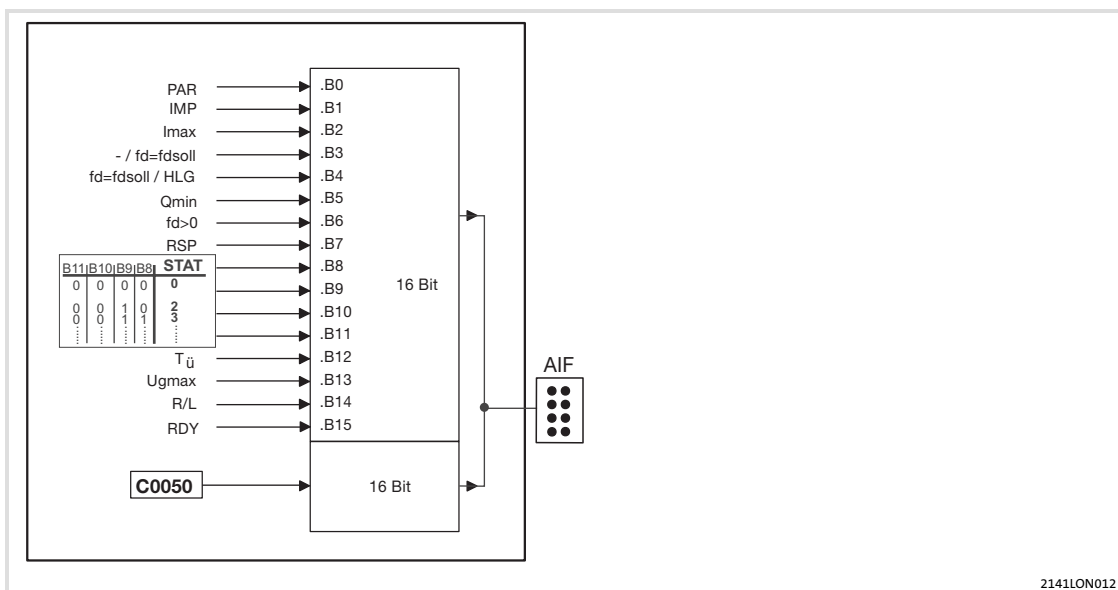
Fig. 7-2   Read access to status word and actual frequency in 82XX (fixed assignment)

Process data telegram to drive

Byte 1	Byte 2	Byte 3	Byte 4
Control word		Setpoint	
High byte	Low byte	High byte	Low byte

Note!

► Frequency and speed values are scaled with
  $\pm 24000 \equiv \pm 480$ Hz.

► Torque values are scaled with $16384 \equiv 100\%$.

Device control word AIF-CTRL for 82XX (C0135, index 5F78$_{hex}$)

820X			821X / 822X / 824X		
Bit	**Assignment**		**Bit**	**Assignment**	
0 / 1	**JOG values**		**0 / 1**	**JOG values**	
	Bit 1	0		Bit 1	0
	0	0	C0046 active		0	0	C0046 active
	0	1	JOG1 in C0037 active		0	1	JOG1 in C0037 active
	1	0	JOG2 in C0038 active		1	0	JOG2 in C0038 active
	1	1	JOG3 in C0039 active		1	1	JOG3 in C0039 active
2	**CW/CCW rotation**		**2**	**CW/CCW rotation**	
	0	CW rotation		0	CW rotation
	1	CCW rotation		1	CCW rotation
3	**Quick stop (QSP)**		**3**	**Quick stop (QSP)**	
	0	QSP not active		0	QSP not active
	1	QSP active		1	QSP active
4 ... 8	**Reserved**		**4**	**Ramp function generator (RFG) stop**	
				0	RFG stop not active
				1	RFG stop active
			5	**Ramp function generator (RFG) zero** (Deceleration on the T$_{if}$ ramp C0013)	
				0	RFG zero not active
				1	RFG zero active
			6	**UP function for motor potentiometer**	
				0	UP not active
				1	UP active
			7	**DOWN function for motor potentiometer**	
				0	DOWN not active
				1	DOWN active
			8	**Reserved**	
9	**Controller inhibit (CINH))**		**9**	**Controller inhibit (CINH))**	
	0	Not active		0	Not active
	1	Active		1	Active
10	**Reserved**		**10**	**Reserved**	
11	**Reserved**		**11**	**TRIP reset**	
				0 -> 1: Edge from 0 to 1	
12	**PAR1** (parameter set changeover)		**12**	**PAR1** (parameter set changeover)	
	0 -> 1: Parameter set 1 -> 0: Parameter set			0 -> 1: Parameter set 1 -> 0: Parameter set	
13	**Reserved**		**13**	**Reserved**	
14	**DC injection brake**		**14**	**DC injection brake**	
	0	DC brake not active		0	DC brake not active
	1	DC brake active		1	DC brake active
15	**Reserved**		**15**	**Reserved**	
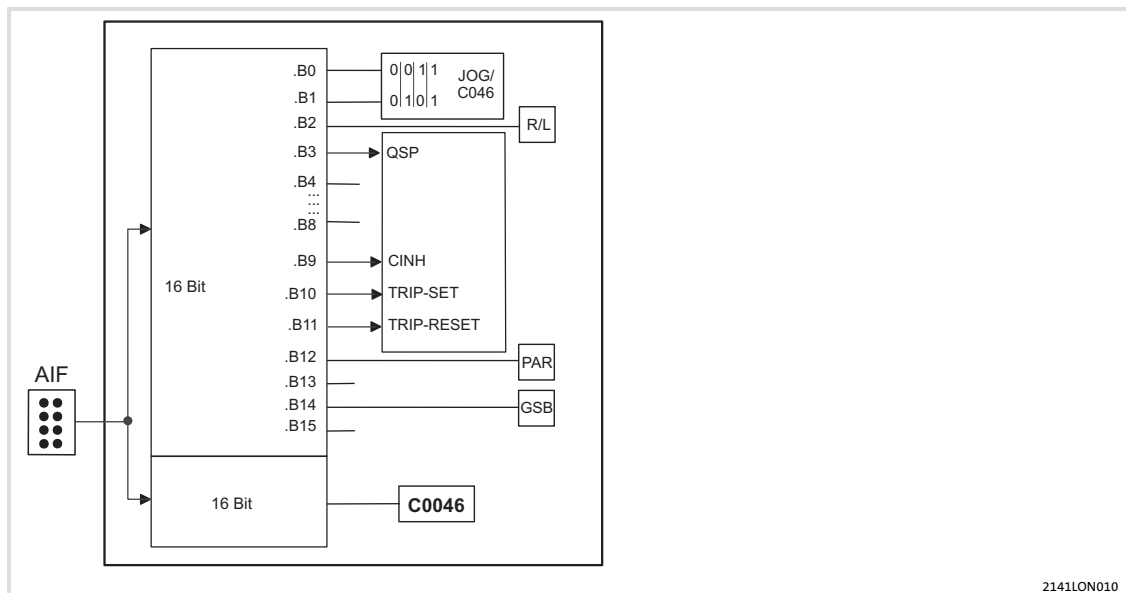
Fig. 7-3   Access to control word and frequency setpoint in 82XX (fixed assignment)

Special features

> ⬛ **Stop!**
>
> ▶ Only carry out a TRIP reset via the fieldbus!
>
> The drive might start running for a short period if a fault is reset via terminal 28 while the controller is being operated with fieldbus control (C0001 = 3) and has assumed the device status "FAULT".
>
> ▶ If the setpoint and the direction of rotation are changed simultaneously via the DRIVECOM speed setpoint, a speed change in the wrong direction of rotation may occur for a short time.
>
> For this reason always send a low rotation direction setpoint first, followed by the new setpoint if the direction of rotation is changed.
>
> This is because first the setpoint is sent to the controller as a unipolar value, followed by the information on the change of the rotation direction.

The X controller is initialised after the "fault reset" command. During this time the controller does not accept any other commands.

7 **Process data transfer**
Lenze device control
Process data signals for 8200 vector frequency inverters

7.1.3   **Process data signals for 8200 vector frequency inverters**

General

Digital and analog input and output signals can be configured freely (see "8200 vector" documentation: codes C0410, C0412, C0417 and C0421).

The change of code C0001 to 3 starts the preconfiguration of the process data words in the controller (📖 46).

Process data telegram from drive

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
Status word		AIF-OUT.W1		AIF-OUT.W2	
High byte	Low byte	High byte	Low byte	High byte	Low byte

AIF-OUT.Wx see C0421.

Lenze

Device status word AIF-STAT for 8200 vector (C0150, index 5F69$_{hex}$)

Bit	Assignment (Lenze setting)	Set under C0417/...
0	**Current parameter set** (DCTRL-PAR-B0)	1
1	**Pulse inhibit** (DCTRL1-IMP)	2
2	**I$_{max}$ limit** (MCTRL1-IMAX)	3
3	**Output frequency = frequency setpoint** (MCTRL1-RFG1=NOUT)	4
4	**Ramp function generator input = ramp function generator output 1** (NSET1-RFG1-I=0)	5
5	**Q$_{min}$ threshold** (PCTRL1-QMIN)	6
6	**Output frequency = 0** (DCTRL1-NOUT=0)	7
7	**Controller inhibit** (DCTRL1-CINH)	8
8 ... 11	**Device status** (DCTRL1-Stat*1 ... STAT*8)	Reserved
12	**Overtemperature warning** (DCTRL1-OH-WARN)	13
13	**DC-bus overvoltage** (DCTRL1-OV)	14
14	**Direction of rotation** (DCTRL1-CCW)	15
15	**Ready for operation** (DCTRL1-RDY)	16

Bit	11	10	9	8	
	0	0	0	0	Device initialisation
	0	0	1	0	Switch-on inhibit
	0	0	1	1	Operation inhibited
	0	1	0	0	Flying restart circuit active
	0	1	0	1	DC injection brake active
	0	1	1	0	Operation enabled
	0	1	1	1	Message active
	1	0	0	0	Fault active
	1	1	1	1	Communication with basic device not possible
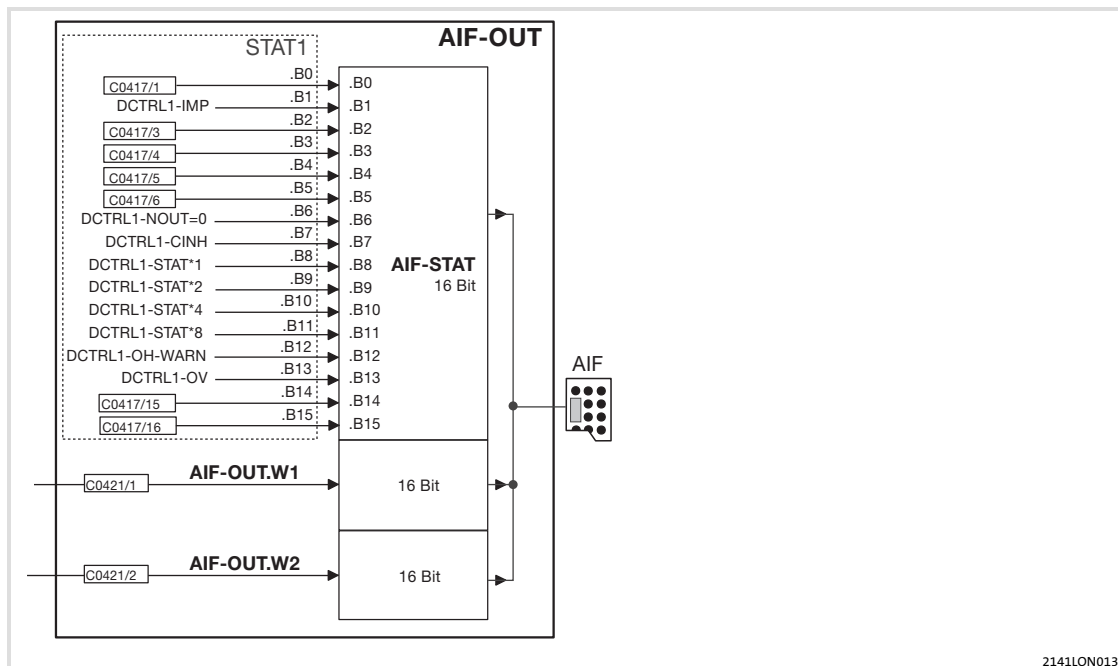
Fig. 7-4   System block AIF-OUT in 8200 vector (freely programmable assignment)

7   **Process data transfer**
Lenze device control
Process data signals for 8200 vector frequency inverters

Process data telegram to drive

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
Control word		AIF-IN.W1		AIF-IN.W2	
High byte	Low byte	High byte	Low byte	High byte	Low byte

AIF-IN.Wx see C0412.

> **Note!**
>
> ► Frequency and speed values are scaled with
> $\pm 24000 \equiv \pm 480$ Hz.
>
> ► Torque values are scaled with $16384 \equiv 100\%$.

Lenze

Device control word AIF-CTRL for 8200 vector (C0135, index 5F78$_{hex}$)

Bit	Assignment (Lenze setting)		Set under C0410/...
	C0001 = 3 with C0007 ≤ 51	C0001 = 3 with C0007 > 51	
0 / 1	**JOG values**	**Freely configurable**	1
			2
Bit	1 0		
	0 0 00 = C0046 active		
	0 1 01 = NSET1-JOG1 (C0037) active		
	1 0 10 = NSET1-JOG2 (C0038) active		
	1 1 11 = NSET1-JOG3 (C0039) active		
2	**Current direction of rotation** (DCTRL1-CW/CCW)	**Freely configurable**	3
	0 Not active		
	1 Active		
3	**Quick stop (QSP)** (AIF-CTRL-QSP)	**Quick stop (QSP)** (AIF-CTRL-QSP)	4
	0 Not active	0 Not active	
	1 Active	1 Active	
4	**Stop ramp function generator** (NSET1-RFG1-STOP)	**Freely configurable**	5
	0 Not active		
	1 Active		
5	**Ramp function generator input = 0** (NSET1-RFG1-0)	**Freely configurable**	6
	0 Not active		
	1 Active		
6	**UP function motor potentiometer** (MPOT1-UP)	**Freely configurable**	7
	0 Not active		
	1 Active		
7	**DOWN function motor potentiometer** (MPOT1-DOWN)	**Freely configurable**	8
8	**Freely configurable**	**Freely configurable**	9
9	**Controller inhibit** (AIF-CTRL-CINH)	**Controller inhibit** (AIF-CTRL-CINH)	10
	0 Not active	0 Not active	
	1 Active	1 Active	
10	**External fault** (AIF-CTRL-TRIP-SET)	**External fault** (AIF-CTRL-TRIP-SET)	11
	0 Not active	0 Not active	
	1 Active	1 Active	
11	**Reset fault** (AIF-CTRL-TRIP-RESET)	**Reset fault** (AIF-CTRL-TRIP-RESET)	12
	0 -> 1 Edge from 0 to 1	0 -> 1 Edge from 0 to 1	
12	**Change over parameter set** (DCTRL1-PAR2/4)	**Freely configurable**	13
	0 Not active		
	1 Active		
13	**Change over parameter set** (DCTRL1-PAR3/4)	**Freely configurable**	14
	0 Not active		
	1 Active		
14	**DC injection brake** (MCTRL1-DCB)	**Freely configurable**	15
	0 Not active		
	1 Active		
15	**Freely configurable**	**Freely configurable**	16

7 **Process data transfer**
Lenze device control
Process data signals for 8200 vector frequency inverters
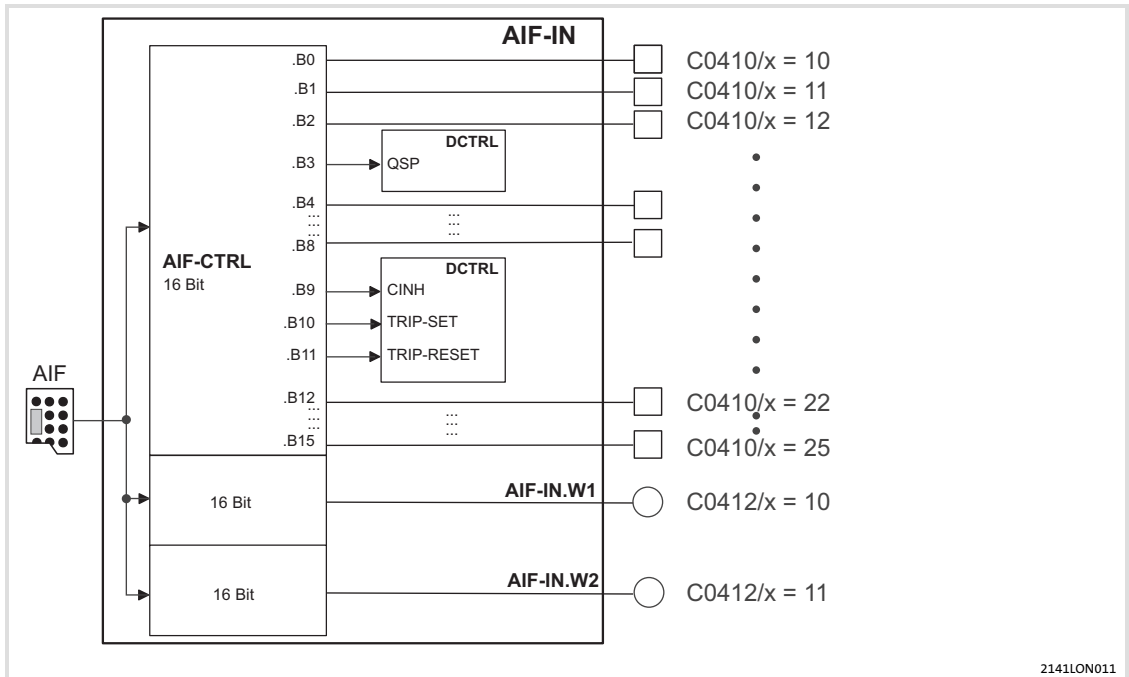
Fig. 7-5   System block AIF-IN in 8200 vector (freely configurable assignment)

7.1.4 Process data signal for 9300 servo inverters

The assignment of the process data for the 93XX controller can be changed by configuring the system blocks AIF-IN and AIF-OUT.

Process data telegram from drive

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
DRIVECOM status word		AIF-OUT.W1		AIF-OUT.W2		AIF-OUT.W3	
High byte	Low byte	High byte	Low byte	High byte	Low byte	High byte	Low byte

Assignment of AIF-OUT.W1 ... W3 depending on the signal configuration selected under C0005:

Signal configuration (C0005)	AIF-OUT.W1	AIF-OUT.W2	AIF-OUT.W3	AIF-OUT. D1
Speed control				
1003 1013 1113	MCTRL-NACT Actual speed value ±100%= ±16383	MCTRL-MSET2 Torque display ±100%= ±16383	MCTRL-NSET2 Speed controller input ±100%= ±16383	Not used
Torque control				
4003 4013 4113	MCTRL-MSET2 Torque display ±100%= ±16383	MCTRL-NACT Actual speed in % ±100%= ±16383	MCTRL-NSET2 Speed controller input ±100%= ±16383	Not used
LF master				
5003 5013 5113	MCTRL-NACT Actual speed value ±100%= ±16383	MCTRL-MSET2 Torque display ±100%= ±16383	MCTRL-NSET2 Speed controller input ±100%= ±16383	Not used
LF slave rail				
6003 6013 6113	MCTRL-NACT Actual speed value ±100%= ±16383	MCTRL-PHI-ACT Angle actual value	MCTRL-MSET2 Torque setpoint in % ±100%= ±16383	Not used
LF slave cascade				
7003 7013 7113	MCTRL-NACT Actual speed value ±100%= ±16383	MCTRL-PHI-ACT Angle actual value	MCTRL-MSET2 Torque setpoint in % ±100%= ±16383	Not used
Not equal to xxx3 (except for own configurations)	MCTRL-NACT Actual speed value ±100%= ±16383	MCTRL-MSET2 Torque display ±100%= ±16383	MCTRL-PHI-ACT Angle actual value	Not used

In the controller, the signals AIF-OUT.W1 ... W3 can be assigned to other signals. For this, the system block configuration is used. The AIF-OUT system block defines the output data of the controller as data interface to the communication module.

Detailed information on system block configuration, 93XX signal configurations (main configurations 1000, 4000, 5000 etc.) and the AIF-OUT system block can be found in the documentation of the 93XX controller.

Device status word AIF-STAT for 93XX

Bit	9300 servo inverter			
	C0005 = 1xx3	C0005 = 4xx3	C0005 = 5xx3	C0005 = 6xx3, 7xx3
0	DCTRL-PAR1-0	DCTRL-PAR1-0	DCTRL-PAR1-0	DCTRL-PAR1-0
1	DCTRL-IMP	DCTRL-IMP	DCTRL-IMP	DCTRL-IMP
2	MCTRL-IMAX	MCTRL-IMAX	REF-OK	REF-OK
3	MCTRL-MMAX	Not used	MCTRL-MMAX	Not used
4	NSET-RFG-I=0	MCTRL-IMAX negated	NSET-RFG-I=0	MCTRL-IMAX negated
5	QMIN	QMIN	REF-BUSY	REF-BUSY
6	DCTRL-NACT=0	DCTRL-NACT=0	DCTRL-NACT=0	DCTRL-NACT=0
7	DCTRL-CINH	DCTRL-CINH	DCTRL-CINH	DCTRL-CINH

8 ... 11	Device status			

Bit	11	10	9	8	
	0	0	0	0	Device initialisation
	0	0	1	0	Switch-on inhibit
	0	0	1	1	Operation inhibited
	0	1	0	0	Flying restart circuit active
	0	1	0	1	DC injection brake active
	0	1	1	0	Operation enabled
	0	1	1	1	Message active
	1	0	0	0	Active fault
	1	0	1	0	Fail-QSP (only 9300 servo position controller)
	1	1	1	1	Communication not possible

Bit	C0005 = 1xx3	C0005 = 4xx3	C0005 = 5xx3	C0005 = 6xx3, 7xx3
12	DCTRL-WARN	DCTRL-WARN	DCTRL-WARN	DCTRL-WARN
13	DCTRL-MESS	DCTRL-MESS	DCTRL-MESS	DCTRL-MESS
14	DCTRL-CW/CCW	DCTRL-CW/CCW	DCTRL-CW/CCW	Not used
15	DCTRL-RDY	DCTRL-RDY	DCTRL-RDY	DCTRL-RDY

Bit	9300 POS	9300 CRV	9300 vector		
	C0005 = 2xxx3	C0005 = xxx3	C0005 = 1xxx, 2xxx, 3xxx, 5xxx, 10xxx, 11xxx	C0005 = 4xx3	C0005 = 6xx3, 7xx3
0	Not used	CERR1-ERR	DCTRL-PAR1-0	DCTRL-PAR1-0	DCTRL-PAR1-0
1	DCTRL-IMP	DCTRL-IMP	DCTRL-IMP	DCTRL-IMP	DCTRL-IMP
2	POS-REF-OK	MCTRL-IMAX	MCTRL-IMAX	MCTRL-IMAX	MCTRL-IMAX
3	Not used	MCTRL-MMAX	MCTRL-MMAX	MCTRL-IMAX negated	MCTRL-MMAX
4	MCTRL-MMAX negated	DCTRL-TRIP	NSET-RFG-I=0	NSET-RFG-I=0	NSET-QSP-OUT
5	POS-IN-TARGET	CDATA-X0	QMIN	QMIN	QMIN
6	DCTRL-NACT=0	DCTRL-NACT=0	DCTRL-NACT=0	DCTRL-NACT=0	DCTRL-NACT=0
7	DCTRL-CINH	DCTRL-CINH	DCTRL-CINH	DCTRL-CINH	DCTRL-CINH
8 ... 11	Device status				

Bit	11	10	9	8	
	0	0	0	0	Device initialisation
	0	0	1	0	Switch-on inhibit
	0	0	1	1	Operation inhibited
	0	1	0	0	Flying restart circuit active
	0	1	0	1	DC injection brake active
	0	1	1	0	Operation enabled
	0	1	1	1	Message active
	1	0	0	0	Active fault
	1	0	1	0	Fail-QSP (only 9300 servo position controller)
	1	1	1	1	Communication not possible

Bit	9300 POS	9300 CRV	9300 vector		
12	DCTRL-WARN	DCTRL-WARN	DCTRL-WARN	DCTRL-WARN	DCTRL-WARN
13	DCTRL-MESS	DCTRL-MESS	DCTRL-MESS	DCTRL-MESS	DCTRL-MESS
14	DCTRL-AIF-QSP	DCTRL-CW/CCW	DCTRL-CW/CCW	DCTRL-CW/CCW	DCTRL-CW/CCW
15	DCTRL-RDY	DCTRL-RDY	DCTRL-RDY	DCTRL-RDY	DCTRL-RDY
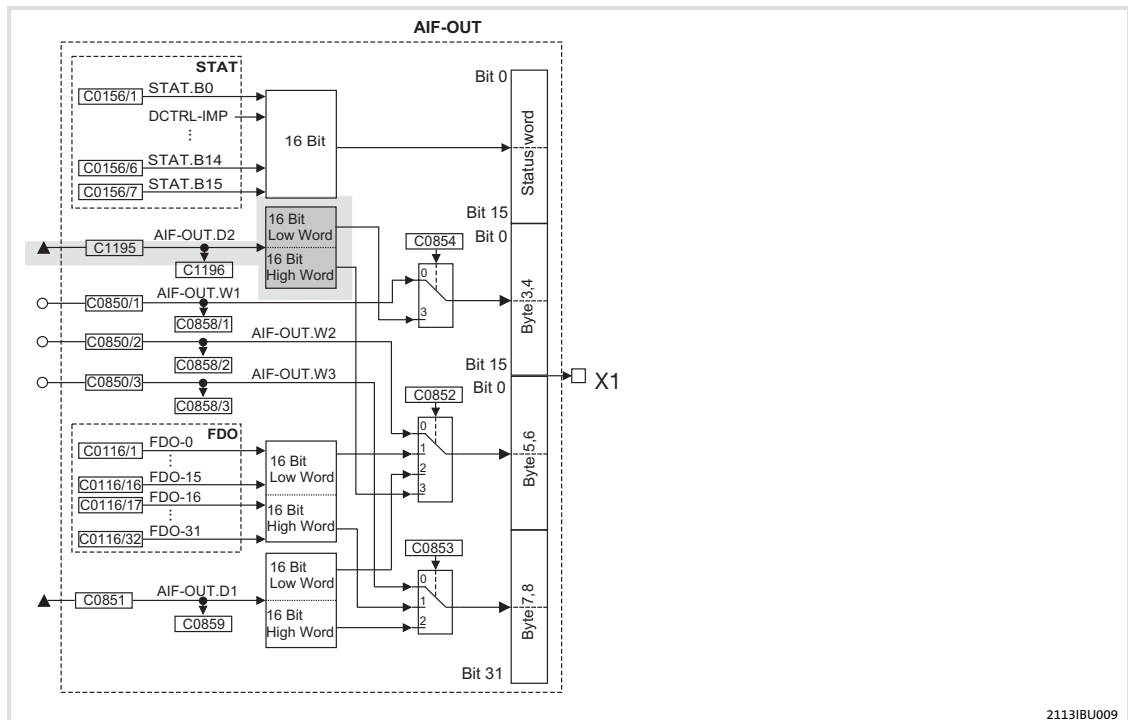
Fig. 7-6   AIF-OUT system block (system block extension on grey background: available from software version 2.0)

7   **Process data transfer**
Lenze device control
Process data signal for 9300 servo inverters

Process data telegram to drive

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Control word		AIF-IN.W1		AIF-IN.W2		AIF-IN.W3	
High byte	Low byte	High byte	Low byte	High byte	Low byte	High byte	Low byte

Assignment of AIF-IN.W1 ... W3 depending on the signal configuration selected under C0005:

Signal configuration (C0005)	AIF-IN.W1	AIF-IN.W2	AIF-IN.W3	AIF-IN.D1
Speed control 1003 / 1013 / 1113	NSET-N Speed setpoint	Not used		
Torque control 4003 / 4013 / 4113	MCTRL-MADD Torque setpoint	Not used		
LF master 5003 / 5013 / 5113	NSET-N Speed setpoint	Not used	Not used	Not used
LF slave rail 6003 / 6013 / 6113	DFSET-A-TRIM Phase trimming	DFSET-N-TRIM Speed trimming		
LF slave cascade 7003 / 7013 / 7113	DFSET-VP-DIV LF factor	DFSET-A-TRIM phase trimming		
Not equal to xxx3	Not used	Not used		

In the controller, the signals AIF-IN.W1 ... W3 can be assigned to other signals. For this, the system block configuration is used. The AIF-IN system block defines the output data of the controller as data interface from the communication module.

Detailed information on system block configuration, 93XX signal configurations (main configurations 1000, 4000, 5000 etc.) and the AIF-IN system block can be found in the documentation of the 93XX controller.

Device control word AIF-CTRL for 93XX

Bit	9300 servo inverter			
	C0005 = 1xx3	C0005 = 4xx3	C0005 = 5xx3	C0005 = 6xx3, 7xx3
0	NSET-JOG*1	Not used	NSET-JOG*1	Not used
1	NSET-JOG*2	Not used	NSET-JOG*2	Not used
2	NSET-N-INV	NSET-N-INV	NSET-N-INV	NSET-N-INV
3	AIF-CTRL.QSP	AIF-CTRL.QSP	AIF-CTRL.QSP	AIF-CTRL.QSP
4	NSET-RFG-STOP	NSET-RFG-STOP	NSET-RFG-STOP	NSET-RFG-STOP
5	NSET-RFG-0	NSET-RFG-0	NSET-RFG-0	NSET-RFG-0
6	Not used	Not used	Not used	Not used
7	Not used	Not used	Not used	Not used
8	Not used	Not used	Not used	Not used
9	AIF-CTRL.CINH	AIF-CTRL.CINH	AIF-CTRL.CINH	AIF-CTRL.CINH
10	AIF-CTRL.TRIP-SET	AIF-CTRL.TRIP-SET	AIF-CTRL.TRIP-SET	AIF-CTRL.TRIP-SET
11	AIF-CTRL.TRIP-RESET	AIF-CTRL.TRIP-RESET	AIF-CTRL.TRIP-RESET	AIF-CTRL.TRIP-RESET
12	DCTRL-PAR*1	DCTRL-PAR*1	DCTRL-PAR*1	DCTRL-PAR*1
13	DCTLR-PAR-LOAD	DCTLR-PAR-LOAD	DCTLR-PAR-LOAD	DCTLR-PAR-LOAD
14	NSET-Ti*1	NSET-JOG*1	REF-ON	REF-ON
15	NSET-Ti*2	NSET-JOG*2	NSET-Ti*1	Not used

Bit	9300 POS	9300 CRV	9300 vector		
	C0005 = 2xxx3	C0005 = xxx3	C0005 = 1xxx, 2xxx, 3xxx, 5xxx, 10xxx, 11xxx	C0005 = 4xx3	C0005 = 6xx3, 7xx3
0	Not used	CSEL1-CAM*1	NSET-JOG*1	Not used	Not used
1	Not used	CSEL1-CAM*2	NSET-JOG*2	Not used	Not used
2	Not used	CSEL1-CAM*4	NSET-N-INV	Not used	Not used
3	AIF-CTRL.QSP	AIF-CTRL.QSP	AIF-CTRL.QSP	AIF-CTRL.QSP	AIF-CTRL.QSP
4	POS-PRG-START	CSEL1-EVENT	NSET-RFG-STOP	NSET-RFG-STOP	Not used
5	POS-PRG-STOP	CDATA-CYCLE	NSET-RFG-0	NSET-RFG-0	Not used
6	Not used	CSEL1-LOAD	Not used	Not used	Not used
7	POS-PRG-RESET	CSEL1-LOAD	Not used	Not used	Not used
8	Not used	Not used	Not used	Not used	Not used
9	AIF-CTRL.CINH	AIF-CTRL.CINH	AIF-CTRL.CINH	AIF-CTRL.CINH	AIF-CTRL.CINH
10	AIF-CTRL.TRIP-SET	AIF-CTRL.TRIP-SET	AIF-CTRL.TRIP-SET	AIF-CTRL.TRIP-SET	AIF-CTRL.TRIP-SET
11	AIF-CTRL.TRIP-RESET	AIF-CTRL.TRIP-RESET	AIF-CTRL.TRIP-RESET	AIF-CTRL.TRIP-RESET	AIF-CTRL.TRIP-RESET
12	POS-PS-CANCEL	Not used	DCTRL-PAR*1	DCTRL-PAR*1	DCTRL-PAR*1
13	POS-PARAM-RD	Not used	DCTLR-PAR-LOAD	DCTLR-PAR-LOAD	DCTLR-PAR-LOAD
14	POS-LOOP-ONH	Not used	NSET-Ti*1	NSET-JOG*1	Not used
15	POS-STBY-STP	Not used	NSET-Ti*2	NSET-JOG*2	Not used

7 **Process data transfer**
Lenze device control
Process data signal for 9300 servo inverters
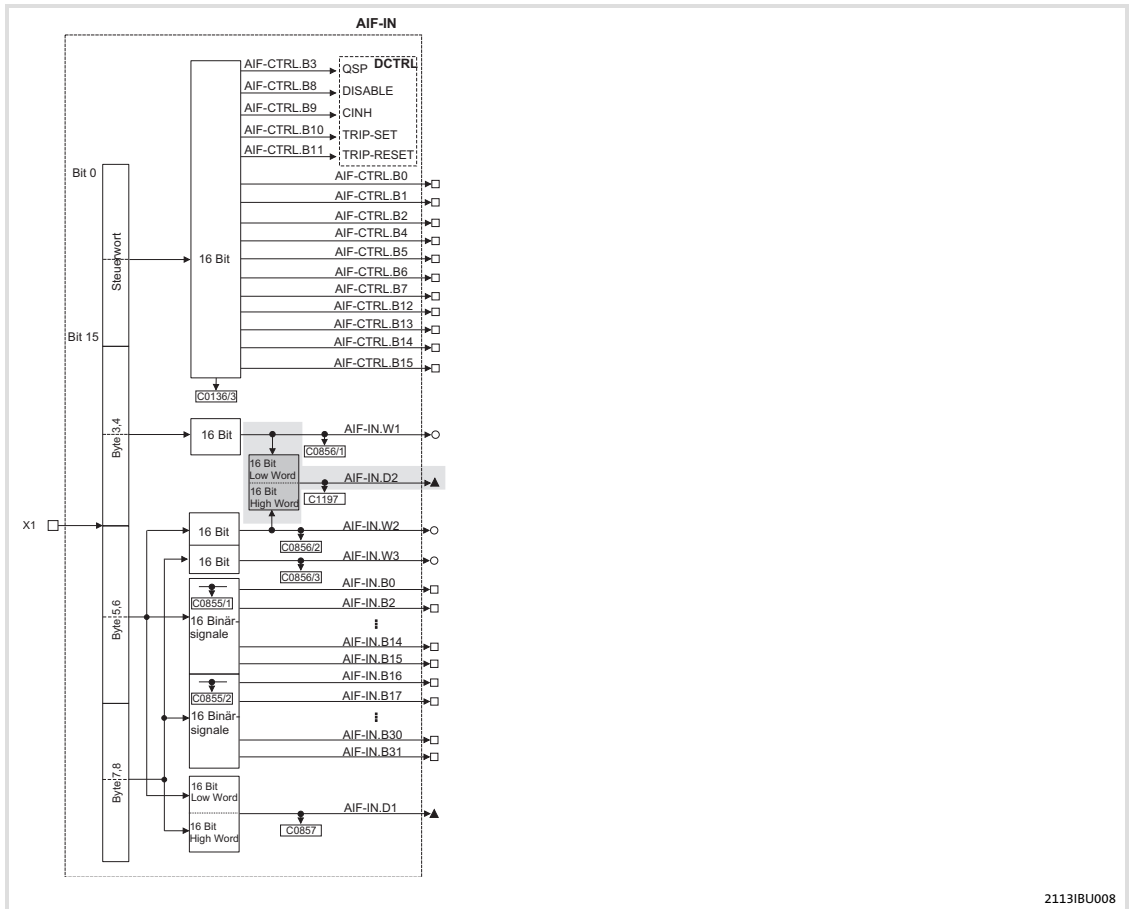
Fig. 7-7   System block AIF-IN (grey: Extension available from software version 2.0)

7.1.5 **Process data signals for 9300 Servo PLC and Drive PLC**

Process data telegram from the drive

Name / variable name	Meaning
AIF1_wDctrlStat	Device status word
AIF_nOutW1_a	AIF word 1
AIF_nOutW2_a	AIF word 2
AIF_nOutW3_a	AIF word 3
AIF2_nOutW1_a	AIF word 4
AIF2_nOutW2_a	AIF word 5
AIF2_nOutW3_a	AIF word 6
AIF2_nOutW4_a	AIF word 7
AIF3_nOutW1_a	AIF word 8
AIF3_nOutW2_a	AIF word 9
AIF3_nOutW3_a	AIF word 10
AIF3_nOutW4_a	AIF word 11
AIF1_dnOutD1_p	AIF double word 1

$\boxed{i}$ **Note!**

9300 Servo PLC

Link the following in the PLC program of the controller:

AIF1_wDctrlCtrl → DCTRL_wAIF1Ctrl

DCTRL_wStat → AIF1_wDctrlStat

Drive PLC

For the Drive PLC the *device control must* be used.

7   **Process data transfer**
Lenze device control
Process data signals for 9300 Servo PLC and Drive PLC

Fig. 7-8   System blocks AIF-OUT1, AIF-OUT2 and AIF-OUT3

Process data telegram to the drive

Name / variable name	Meaning
AIF1_wDctrlCtrl	Device control word
AIF1_nInW1_a	AIF word 1
AIF1_nInW2_a	AIF word 2
AIF1_nInW3_a	AIF word 3
AIF2_nInW1_a	AIF word 4
AIF2_nInW2_a	AIF word 5
AIF2_nInW3_a	AIF word 6
AIF2_nInW4_a	AIF word 7
AIF3_nInW1_a	AIF word 8
AIF3_nInW2_a	AIF word 9
AIF3_nInW3_a	AIF word 10
AIF3_nInW4_a	AIF word 11
AIF1_dnInD1_p	AIF double word 1

Note!

9300 Servo PLC

Link the following in the PLC program of the controller:

► AIF1_wDctrlCtrl → DCTRL_wAIF1Ctrl

► DCTRL_wStat → AIF1_wDctrlStat

Drive PLC

For the Drive PLC the *device control* must be used.
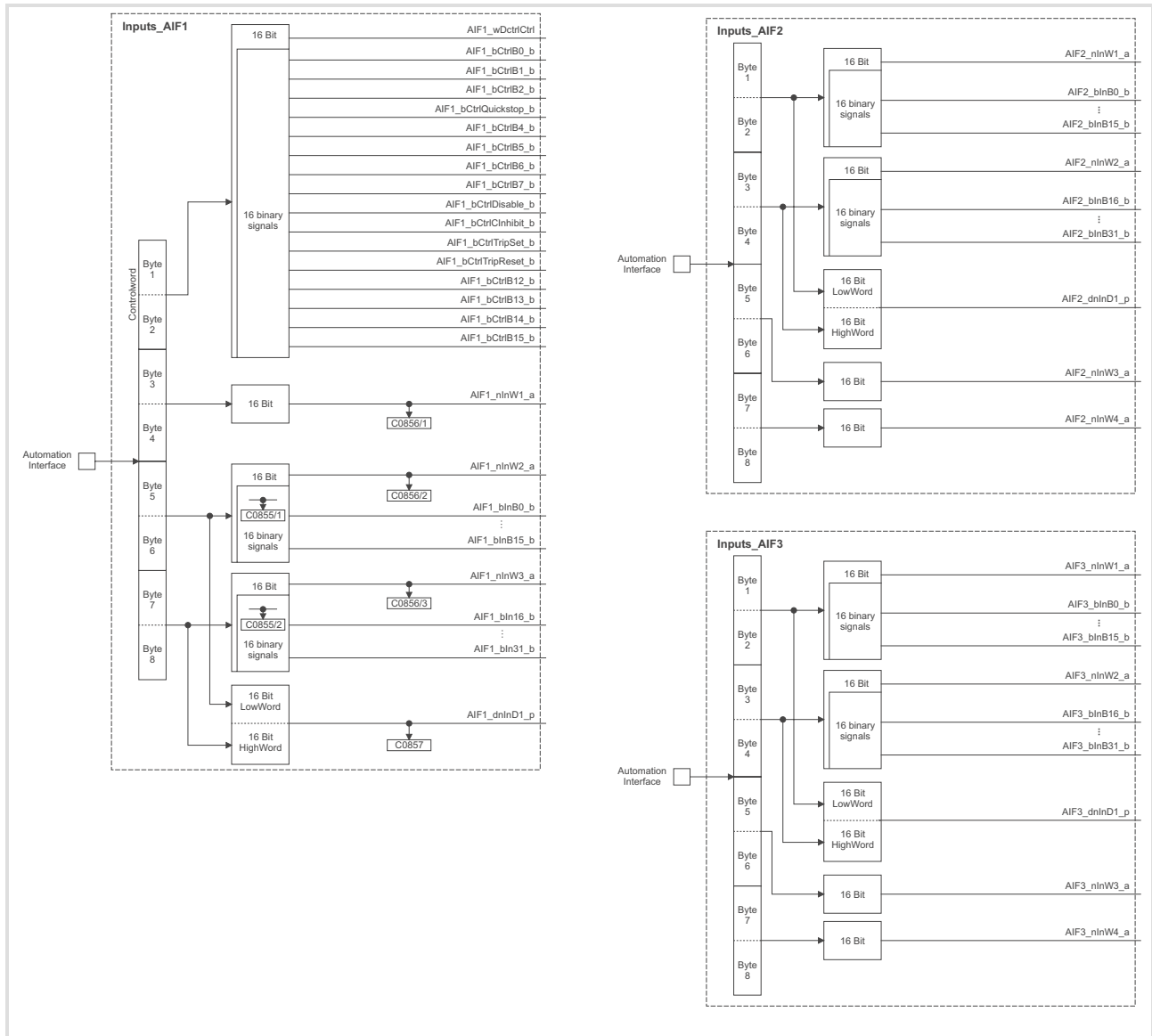
Fig. 7-9　　　System blocks AIF-IN1, AIF-IN2 and AIF-IN3

7.1.6 Process data signals for axis modules ECSxS / ECSxA

Detailed information ....

on the process data transfer for the ECS servo system and the required system blocks and settings can be found in the documentation of the following devices:

► ECSxS axis module (speed and torque)

► ECSxA axis module (application)

7.2 DRIVECOM control

> ℹ️ **Note!**
>
> **ECS servo system**
>
> ECS devices cannot be used with the DRIVECOM or PROFIdrive control.

7.2.1 Provide DRIVECOM compatibility

Drivecom Profile 20 is a non-proprietary definition of important parameters and the controller behaviour.

Apart from DRIVECOM definitions, there are other Lenze specific functionalities which must be adapted to achieve full compatibility to the DRIVECOM Profile.

The device-specific changes are listed in the table below:

Controller	Device-specific changes
820X	For the 820X controller, parameters can only be set during controller inhibit. Controller inhibit is active for the DRIVECOM device statuses ● "SWITCH-ON INHIBIT" ● "READY TO START" ● "SWITCHED ON" ● "FAULT"
821X / 822X / 8200 vector	The automatic DC injection brake must be deactivated in all parameter sets, i.e. ● C0106=0 ● C2106=0 ● C4106=0 (8200 vector only) ● C6106=0 (8200 vector only) If the automatic DC injection brake is not deactivated (holding time of DC injection brake C0106 unequal to 0), the controller automatically switches from the device status "OPERATION ENABLED" to the status "SWITCHED ON" when the speed is 0 and the holding time of the DC injection brake has elapsed. If the setpoint is greater than 0, the controller is automatically reset to the status "ENABLE OPERATION".
93XX	Set a DRIVECOM speed signal configuration under code C0005 (e. g.: C0005=1013). This configuration corresponds to the signal configuration 1000 with the following changes: ● Setpoint selection via PROFIBUS ● DRIVECOM control via PROFIBUS ● Output X5.A1 is switched as a voltage output for the internal supply of the digital inputs. ● Actual values and status signals for PROFIBUS Detailed descriptions of the signal configuration can be found in the documentation for the controller.

> ℹ️ **Note!**
>
> The states FAIL-QSP and QSP are not considered in the status messages of the controller, compare C0150.
>
> Communication modules with drive profiles such as DRIVECOM or PROFIdrive therefore cannot evaluate this information in the Lenze setting.
>
> FAIL-QSP and QSP will only be evaluated after the outputs MCTRL-QSP-OUT or DCTRL-FAIL-QSP have been set to a free bit of the AIF status word (STAT system block) in a user-defined setting.

Process data transfer **7**
DRIVECOM control
DRIVECOM state machine

7.2.2 DRIVECOM state machine

For the PROFIBUS control and the use of the EMF2133IB communication module the Lenze controllers feature the standardised device states in accordance with the "Drive technology 20" DRIVECOM profile. Associated preconfiguration of the controllers:

► 82XX: C0001 = 3

► 93XX: C0005 = xxx3

Information on the current device state (marked by rectangles in the diagram) is stored in the DRIVECOM parameter "status word". Commands in the DRIVECOM parameter "control word" can change the device state. These commands are represented by arrows in the following diagram.
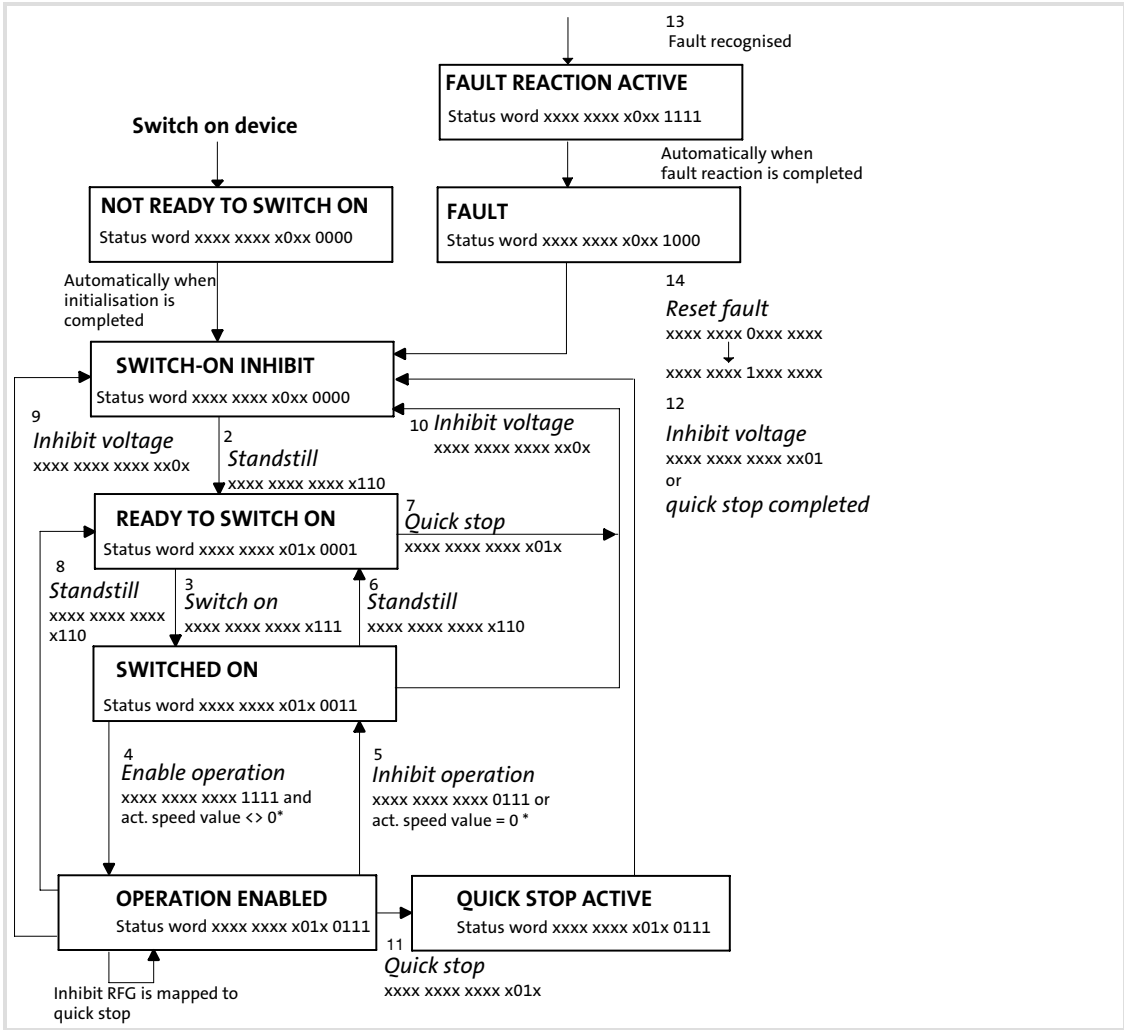
Fig. 7-10   Status diagram of DRIVECOM device control

*   only effective for 821X, 8200 vector when the automatic DC injection brake is active (C0106, C2106 <> 0)

7

Process data transfer
DRIVECOM control
DRIVECOM state machine

82XX, 8200 vector (C0001 ╪3)

For standard device control you enter the control information via the corresponding control inputs (terminal):

Information on the current device status (see Fig. 7-11, marked by rectangles) is stored in the DRIVECOM parameter "status word". Commands in the DRIVECOM parameter "control word" are switched off and cannot change the device status. The commands to change the device status are specified by the respective control input. These commands are represented by arrows in the following diagram.
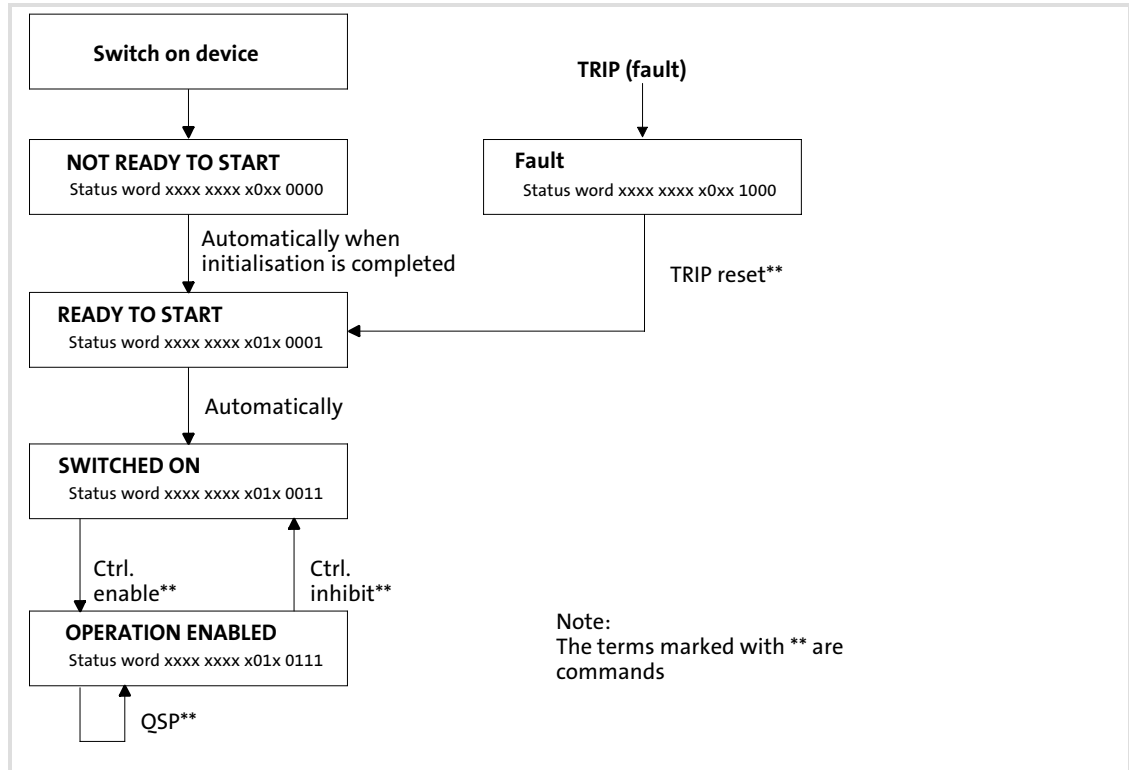
Fig. 7-11   Status diagram for standard device control

Status	Meaning
NOT READY TO START	Controller is being initialised and not yet ready for operation. Afterwards the device status will automatically change to "READY TO START".
READY TO START	Controller is inhibited (CINH) and waits until the power stage has been charged. Afterwards the device status automatically changes to "SWITCHED ON".
SWITCHED ON	The controller is inhibited (CINH) and waits for controller enable (RFR).
OPERATION ENABLED	The controller is enabled RFR). In this device status the pulse inhibit (IMP) can be set automatically.
FAULT	The controller is in the "FAULT" (TRIP) device status.

Process data transfer **7**
DRIVECOM control
DRIVECOM control word

7.2.3 DRIVECOM control word

Bit	Name	Meaning	
0	Switch on	0 = command 2, 6, 8 (controller inhibit) 1 = command 3 (controller inhibit)	
1	Voltage inhibit	0 = Voltage inhibit active 1 = Voltage inhibit not active	
2	Quick stop	0 = quick stop (QSP) active 1 = quick stop (QSP) inactive	
3	Enable operation	0 = Inhibit operation 1 = Enable operation	
4	RFG inhibit	Inhibit of the ramp function generator (RFG): The quick stop function (QSP) is activated, the device status of the drive does not change. 0 = RFG inhibit (QSP) 1 = RFG inhibit not active	
5	RFG stop	820X	Not used
		821X / 822X	Output of the RFG (speed setpoint integrator) is "frozen". 0 = RFG stop 1 = RFG stop not active
		8200 vector / 93XX	Free (mapping to bit AIF-CTRL.B4 negated)
6	RFG zero	820X	Not used
		821X / 822X	Ramp function generator input (speed setpoint integrator) is set to 0. This results in a controlled deceleration along the set ramp. 0 = RFG zero 1 = RFG zero not active
		8200 vector / 93XX	Free (mapping to bit AIF-CTRL.B5 negated)
7	Error reset	Reset of a fault (TRIP): ● A bit change from 0 to 1 is required. ● For 82XX, the controller is initialised. During this time, the controller does not accept any commands.	
8 ... 10	Reserved		
11	Manufacturer	820X / 821x / 822x	Not used
		8200 vector / 93XX	Free (mapping to bit AIF-CTRL.B7)
12	Manufacturer	820X / 821x / 822x	Parameter set changeover: 0 − 1 = Parameter set 2 1 − 0 = Parameter set 1
		8200 vector / 93XX	Free (mapping to bit AIF-CTRL.B12)
13	Manufacturer	820X / 821x / 822x	DC-injection brake: 0 = DCB not active 1 = DCB active
		8200 vector / 93XX	Free (mapping to bit AIF-CTRL.B13)
14	Manufacturer	820X / 821x / 822x	Not used
		8200 vector / 93XX	Free (mapping to bit AIF-CTRL.B14)

Lenze

7

Process data transfer
DRIVECOM control
DRIVECOM control word

Bit	Name	Meaning	
15	Manufacturer	820X	Process input data inhibit: Inhibit update of the process output data of the controller (input data for the master). The updating of status and actual information of the process data channel can be inhibited in order to transmit control information more precisely in time. 0 = Read status and actual value 1 = Do not read status and actual value
		821X / 822X	Not used
		8200 vector / 93XX	Free (mapping to bit AIF-CTRL.B15)

Lenze

Process data transfer **7**
DRIVECOM control
DRIVECOM status word

7.2.4 DRIVECOM status word

Bit	Name	Description		
0	Ready to start	Device status information 0 = Status lower than "READY TO START" 1 = Status at least "READY TO START"		
1	Switched on	Device status information 0 = Status lower than "SWITCHED ON" 1 = Status at least "SWITCHED ON"		
2	Operation enabled	Device status information 0 = Status lower than "OPERATION ENABLED" 1 = Status "OPERATION ENABLED"		
3	Fault	Device status information 0 = No fault (TRIP) 1 = Fault (TRIP) occurred		
4	Voltage inhibited	Information about command "Voltage inhibit" 0 = Command active 1 = Command not active		
5	Quick stop	Information about command "Quick stop" (QSP) 0 = Command (QSP) active 1 = Command (QSP) not active		
6	Switch-on inhibit	Device status information 0 = Status not "SWITCH-ON INHIBIT" 1 = Status "SWITCH-ON INHIBIT"		
7	Warning	Collective warning 0 = No warning 1 = Warning		
8	Message	Group signal: Collective message. Automatic setting and resetting of pulse inhibit (IMP) in the device status "OPERATION ENABLED". Reasons for this can be undervoltage or overvoltage as well as overcurrent. 0 = No message 1 = Message (IMP)		
9	Remote	82xx / 821x / 822x / 8200 vector	Bus access right, depending on Lenze parameter "Operating mode" (C0001) 0 = C0001 ≠ 3 1 = C0001 = 3	
		93XX	1	
10	Setpoint reached	Status of the speed/frequency deviation 0 = RFG on ≠ RFG off 1 = RFG on = RFG off		
11	Limit value	Status of the DRIVECOM speed limitation is always "0".		
12	Reserved	82xx / 821x / 822x	Not used	
		8200 vector / 93XX	Mapping to bit C0150.B14	
13	Reserved	82xx / 821x / 822x	Not used	
		8200 vector	Mapping to bit C0150.B15	
		93XX	Mapping to bit C0150.B3	
14	Manufacturer	82xx / 821x / 822x	I_{max} (current limit reached) 0 = Current limit not reached 1 = Current limit exceeded	
		8200 vector / 93XX	Mapping to bit C0150.B2	

7 **Process data transfer**
DRIVECOM control
DRIVECOM status word

Bit	Name	Description	
15	Manufacturer	82xx / 821x / 822x	Q_{min} ($f_d < d_{Qmin}$) 0 = Q_{min} not active 1 = Q_{min} active
		8200 vector / 93XX	Mapping to bit C0150.B5

Lenze

Process data transfer **7**
DRIVECOM control
Bit control commands

7.2.5 Bit control commands

The bit control commands of the control word depend on other bit settings. The command is only executed with the following bit patterns:

Device status commands	Bits of the control word							
	7	6	5	4	3	2	1	0
1 Standstill						1	1	**0**
2 Switch on						1	1	**1**
3 Enable operation					**1**	1	1	1
4 Inhibit operation					**0**	1	1	1
5 Voltage inhibit							**0**	
6 Quick stop						**0**	1	
8 Error reset	0→1							

Error reset
RFG zero
RFG stop
RFG inhibit
Enable operation
Quick stop
Voltage inhibit
Switch on
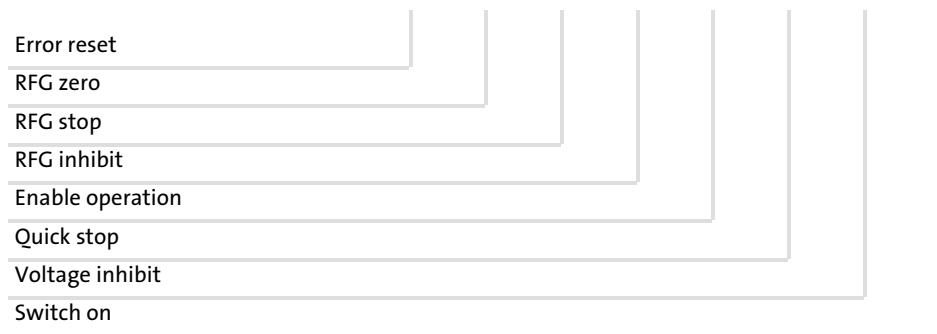
0: Bit status is 0
1: Bit status is 1
Empty: Any bit status, no influence

Command	Description
COMMAND 2, 6, 8 (standstill) Control word: bit 0 = 0	Command to change from different device status to the READY TO START status.
COMMAND 3 (switch on)	Command to change to the SWITCHED ON device status.
COMMAND 4 (enable operation)	Command to change to the OPERATION ENABLED device status. The controller inhibit is deactivated.
COMMAND 5 (inhibit operation)	Command to change to the SWITCHED ON device status. Controller inhibit is activated.
COMMAND 7, 9, 10, 12 (voltage inhibit) Control word: bit 1 = 0	Command to change to the SWITCH-ON INHIBIT device status. Controller inhibit is activated.
COMMAND 7, 10, 11 (quick stop) Control word: bit 2 = 0	Command to change to the SWITCH-ON INHIBIT device status. If the drive was enabled, a controlled deceleration (quick stop deceleration ramp) is executed.
COMMAND 13 (fault/TRIP)	The controller has detected a fault. Some faults require a controlled deceleration (device-dependent). When it is completed, the device status changes to FAULT.
COMMAND 14 (reset fault/TRIP) Control word: Bit 7 = 0 ⇒ 1	In the case of the 821X, 8200 vector device series, this command resets a fault. The controller changes to the SWITCH-ON INHIBIT device status when a fault has been eliminated.

7.2.6 Status bits

The current device status is clearly coded in bits 0 ... 6 of the status word:

Device status	Bits of the status word						
	6	5	4	3	2	1	0
NOT READY TO START	0			0	0	0	0
SWITCH-ON INHIBIT	1			0	0	0	0
READY TO START	0	1		0	0	0	1
SWITCHED ON	0	1		0	0	1	1
OPERATION ENABLED	0	1		0	1	1	1
FAULT	0			1	0	0	0
FAULT REACTION ACTIVE	0			1	1	1	1
QUICK STOP ACTIVE	0	0		0	1	1	1

Switch-on inhibit
Quick stop
Voltage inhibit
Fault
Operation enabled
Switched on
Ready to start
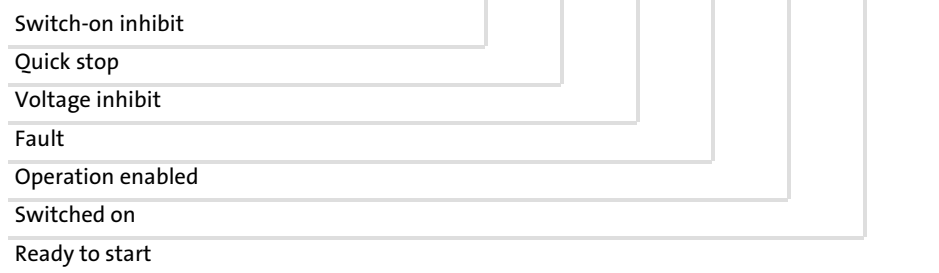
0: Bit status is 0
1: Bit status is 1
Empty: Any bit status, no influence

Status	Description
NOT READY TO START	During initialisation, the controller is not yet ready for operation. Afterwards it automatically switches to the "READY TO START" device status.
SWITCH-ON INHIBIT	The controller is inhibited and waits for command 2 (standstill).
READY TO START	The controller is inhibited and waits for command 3 (switch on).
SWITCHED ON	The controller is inhibited and waits for command 4 (enable operation).
OPERATION ENABLED	The controller is enabled. In this device status the pulse inhibit can be set automatically.
FAULT REACTION ACTIVE	A fault (TRIP) was recognised and a time-based fault-dependent response has been initiated.
FAULT	The controller is in the "FAULT" (TRIP) device status.
QUICK STOP ACTIVE	The "quick stop" (QSP) command was given in the device status "OPERATION ENABLED". A controlled deceleration (quick stop ramp) is executed. After deceleration the device changes automatically to the state "SWITCH-ON INHIBIT".

7.3 PROFIdrive control

> **ℹ️ Note!**
>
> **ECS servo system**
>
> ECS devices cannot be used with the DRIVECOM or PROFIdrive control.

7.3.1 Establishing PROFIdrive compatibility

The PROFIdrive profile, version 2, is a non-proprietary definition of important parameters and the controller behaviour.

Apart from PROFIdrive definitions, there are other Lenze specific functionalities which must be adapted to achieve full compatibility to the PROFIdrive profile.

The device-specific changes are listed in the table below:

Controller	Device-specific changes
820X	For the 820X controller, parameters can only be set during controller inhibit. Controller inhibit is active for the DRIVECOM device statuses • "NOT READY TO START" • "READY TO START" • "READY FOR OPERATION" • "FAULT"
821X / 822X / 8200 vector	The automatic DC injection brake must be deactivated in all parameter sets, i.e. • C0106=0 • C2106=0 • C4106=0 (8200 vector only) • C6106=0 (8200 vector only) If the automatic DC injection brake is not deactivated (holding time of DC injection brake C0106 unequal to 0), the controller automatically switches from the device status "OPERATION ENABLED" to the status "READY FOR OPERATION" when the speed is 0 and the holding time of the DC injection brake has elapsed. If the setpoint is greater than 0, the controller is automatically reset to the status "ENABLE OPERATION".
93XX	Set a PROFIdrive speed signal configuration under code C0005 (e.g.: C0005=1013). This configuration corresponds to the signal configuration 1000 with the following changes: • Setpoint selection via PROFIBUS • PROFIdrive control via PROFIBUS • Output X5.A1 is switched as a voltage output for the internal supply of the digital inputs. • Actual values and status signals for PROFIBUS Detailed descriptions of the signal configuration can be found in the documentation for the controller.

> **ℹ️ Note!**
>
> The states FAIL-QSP and QSP are not considered in the status messages of the controller, compare C0150.
>
> Communication modules with drive profiles such as DRIVECOM or PROFIdrive therefore cannot evaluate this information in the Lenze setting.
>
> FAIL-QSP and QSP will only be evaluated after the outputs MCTRL-QSP-OUT or DCTRL-FAIL-QSP have been set to a free bit of the AIF status word (STAT system block) in a user-defined setting.

7 **Process data transfer**
PROFIdrive control
PROFIdrive state machine

7.3.2 PROFIdrive state machine

Example:
Status information by parameter "status word"
bit 15 ... bit 0 (binary representation)

Command / control word:
OFF1 / xxxx x1xx xxxx x110
OFF2 / xxxx x1xx xxxx xx0x
OFF3 / xxxx x1xx xxxx x01x

Commands / priority:
OFF2 / highest priority
OFF3 /
OFF1 / lowest priority

Fault recognised

Fault reaction active
Status word xxxx xxxx x0xx 1111

Automatically when fault reaction is completed

Fault
Status word xxxx xxxx x0xx 1xxx

Reset fault
8200 automatically when the fault reaction is completed

Reset fault 8210
xxxx xxxx 0xxx xxx
↓
xxxx x1xx 1xxx xxx

SWITCH-ON INHIBIT
Status word xxxx xxxx x0xx 0000

OFF1

Switch on devices

Automatically when initialisation is completed

OFF2 / OFF3 ← **Not ready to start**
Status word xxxx xxxx x0xx x000

OFF1

OFF2 / OFF3 ← **Ready to start**
Status word xxxx xxxx x0xx x001

Switch-on warning
xxxx x1xx xxxx x111

OFF2 / OFF3 ← **Ready for operation**
Status word xxxx xxxx xxxx x011 → OFF1

Enable operation
xxxx x1xx xxxx 1111
and
$v_{act.} <> 0$

Inhibit operation
xxxx x1xx xxxx 0111
or
$v_{act.} = 0$

$v_{act.} = 0$

OFF2 ← **Operation enabled**
Status word xxxx xxxx xxxx x111

OFF1-ACTIVE
Status word xxxx xxxx xx11 x001

OFF3   OFF1

$v_{act.} = 0$ **OFF3-ACTIVE**
Status word xxxx xxxx xx0x x111

$v_{act.}$:
Actual speed value

Fig. 7-12   Status diagram of PROFIDrive device control

7.3.3 PROFIdrive control word

Bit	Name	Meaning		
0	OFF1	0 = OFF1 active; RFG zero, controller inhibit at zero speed 1 = OFF1 inactive		
1	OFF2	0 = OFF2 active 1 = OFF2 inactive		
2	OFF3	0 = OFF3 active 1 =OFF3 inactive		
3	Enable operation	0 = Inhibit operation 1 = Enable operation		
4	RFG inhibit	Inhibit of the ramp function generator (RFG): The quick stop function (QSP) is activated, the device status of the drive does not change. 0 = RFG inhibit (QSP) 1 = RFG inhibit not active		
5	RFG stop	820X	Not used	
		821X / 822X	Output of the RFG (speed setpoint integrator) is "frozen". 0 = RFG stop 1 = RFG stop not active	
		8200 vector / 93XX	Free (mapping to bit AIF-CTRL.B4 negated)	
6	Inhibit setpoint	820X	Not used	
		821X / 822X	Ramp function generator input (speed setpoint integrator) is set to 0. This results in a controlled deceleration along the set ramp. 0 = RFG zero 1 = RFG zero not active	
		8200 vector / 93XX	Free (mapping to bit AIF-CTRL.B5 negated)	
7	Error reset	Reset of a fault (TRIP): ● A bit change from 0 to 1 is required. ● For 82XX, the controller is initialised. During this time, the controller does not accept any commands.		
8	Jogging 1	Not used		
9	Jogging 2	Not used		
10	Master function from automation device	0 = No master function by automation device 1 = Master function by automation device		
11	Manufacturer	820X / 821x / 822x	Not used	
		8200 vector / 93XX	Free (mapping to bit AIF-CTRL.B7)	
12	Manufacturer	820X / 821x / 822x	Parameter set changeover: 0 − 1 = Parameter set 2 1 − 0 = Parameter set 1	
		88200 vector / 93XX	Free (mapping to bit AIF-CTRL.B12)	
13	Manufacturer	820X / 821x / 822x	DC injection brake 0 = DCB not active 1 = DCB active	
		8200 vector / 93XX	Free (mapping to bit AIF-CTRL.B13)	
14	Manufacturer	820X / 821x / 822x	Not used	
		8200 vector / 93XX	Free (mapping to bit AIF-CTRL.B14)	

7 **Process data transfer**
PROFIdrive control
PROFIdrive control word

Bit	Name	Meaning	
15	Manufacturer	820X	Process input data inhibit Inhibit update of the process output data of the controller (input data for the master). The updating of status and actual information of the process data channel can be inhibited in order to transmit control information more precisely in time. 0 = Read status and actual value 1 = Do not read status and actual value
		821X /822X	Not used
		8200 vector / 93XX	Free (mapping to bit AIF-CTRL.B15)

Lenze

7.3.4 PROFIdrive status word

Bit	Name	Meaning	
0	Ready to start	Device status information 0 = Status lower than "READY TO START" 1 = Status at least "READY TO START"	
1	Ready for operation	Device status information 0 = Status lower than "READY FOR OPERATION" 1 = Status at least "READY FOR OPERATION"	
2	Operation enabled	Device status information 0 = Status lower than "OPERATION ENABLED" 1 = Status "OPERATION ENABLED"	
3	Fault	Device status information 0 = No fault (TRIP) 1 = Fault (TRIP) occurred	
4	OFF2	Information via command OFF2 0 = Command active 1 = Command not active	
5	OFF3	Information via command OFF3 0 = Command active 1 = Command not active	
6	Switch-on inhibit	Device status information 0 = Status not "SWITCH-ON INHIBIT" 1 = Status "SWITCH-ON INHIBIT"	
7	Warning	Collective warning 0 = no warning 1 = warning	
8	Reserved	Always 1	
9	Master function requested	82xx / 821x / 822x / 8200 vector	Bus access right, depending on Lenze parameter "Operating mode" (C0001) 0 = C0001 <> 3 1 = C0001 = 3
		93XX	1
10	Setpoint reached	Status of the speed/frequency deviation 0 = HLG on <> HLG off 1 = RFG on = RFG off	
11	Reserved	0	
12	Manufacturer	82xx / 821x / 822x	Not used
		8200 vector / 93XX	Mapping to bit C0150.B14
13	Manufacturer	82xx / 821x / 822x	Not used
		8200 vector	Mapping to bit C0150.B15
		93XX	Mapping to bit C0150.B3
14	Manufacturer	82xx / 821x / 822x	I_{max} (current limit reached) 0 = Current limit not reached 1 = Current limit exceeded
		8200 vector / 93XX	Mapping to bit C0150.B2
15	Manufacturer	82xx / 821x / 822x	Q min (f d < dQmin) 0 = Q min not active 1 = Q min active
		8200 vector / 93XX	Mapping to bit C0150.B5

8   Parameter data transfer

PROFIBUS transmits parameter data and process data between the host (master) and the drives connected to the bus (slaves). The data is transmitted via corresponding communication channels depending on their time-critical behaviour.

▶ Parameter data are transferred via the parameter data channel.
 – DRIVECOM parameter data channel (DP-V0)
 – PROFIdrive parameter data channel (DP-V1)

▶ Via the parameter data channel, access to all Lenze codes is enabled.

▶ The transmission of parameter data usually is not time-critical.

▶ Parameter data for example are operating parameters, diagnostics information, and motor data.

▶ The cyclic DP-V0 parameter data are based on the DRIVECOM profile. When the parameter data channel based on DP-V0 is active, it additionally assigns 4 words of the process input data and the process output data.

▶ The acyclic DP-V1 parameter data are based on the PROFIdrive profile.

8.1   Lenze parameter sets

Stop!

82XX / 8200 vector / Drive PLC and communication module:

► The cyclic writing of parameter data into the EEPROM is not permissible. It may damage or destroy the EEPROM.

Only 8200 vector:

► After every mains switching, set code C0003 to the value "0" if you want to change the parameter data cyclically.

8.1.1   Parameter sets for 82XX controllers

The 82XX controller has two parameter sets, the parameters of which can be directly addressed.

Addressing

Addressing is effected by means of a code offset:

● Offset 0 addresses parameter set 1 with the Lenze codes C0000 to C1999.
● Offset "2000" addresses parameter set 2 with the Lenze codes C2000 to C3999.

If a parameter is only available once (see Operating Instructions for the controller), use code offset "0".

Example

Addressing of code C0011 (maximum field frequency) in different parameter sets:

● C0011 in parameter set 1: Lenze code number = 11
● C0011 in parameter set 2: Lenze code number = 2011

If a parameter is only available once (see Operating Instructions for the controller), use code offset "0".

Note!

► Automatic storage of the parameter data changed

► No automatic storage of process data

8.1.2 Parameter sets for 8200 vector controller

The 8200 vector controller has four parameter sets the parameters of which can be directly addressed via the bus.

Addressing

Addressing is carried out with a code offset:

► Offset "0" addresses parameter set 1 with codes C0000 ... C1999.

► Offset "2000" addresses parameter set 2 with codes C2000 ... C3999.

► Offset "4000" addresses parameter set 3 with codes C4000 ... C5999.

► Offset "6000" addresses parameter set 4 with codes C6000 ... C7999.

If a parameter is only available once (see documentation of the controller), use code offset "0".

Example

Addressing of code C0011 (maximum field frequency) in different parameter sets:

► C0011 in parameter set 1: code number = 11

► C0011 in parameter set 2: code number = 2011

► C0011 in parameter set 3: code number = 4011

► C0011 in parameter set 4: code number = 6011

> **Note!**
>
> Automatic storage of new parameter data is activated (basic Lenze setting, can be deactivated via C0003).

8.1.3   **Parameter sets for controller 93XX**

The drive controllers 93XX feature up to four parameter sets for storage in the EEPROM for each technology variant. An additional parameter set is located in the main memory of the drive controller. This parameter set is referred to as "current" parameter set.

Addressing

▶   Only the current parameter set can be directly addressed via the bus.

▶   After power-on, parameter set 1 is automatically loaded into the current parameter set.

▶   Before changing the parameters of parameter sets 2 ... 4, the parameter sets must be activated.

> 🛈   **Note!**
>
> Changes in the current parameter set are lost after switching off the drive controller.
>
> Code C0003 is used to store the current parameter set (see documentation of the controller).

8   **Parameter data transfer**
Lenze parameter sets
Parameter sets for Drive PLC and ECSxS / ECSxA axis modules

8.1.4   Parameter sets for Drive PLC and ECSxS / ECSxA axis modules

The Drive PLC and ECSxS / ECSxA axis modules each have a parameter set for storage in the EEPROM. Another parameter set is stored in the main memory. This parameter set is called "current" parameter set.

Addressing

- Only the current parameter set can be directly addressed via the bus.
- After switch-on, parameter set 1 is automatically loaded in the current parameter set.

Note!

Changes within the current parameter set are lost after the controller is switched off.

Code C0003 serves to save the current parameter set (see documentation of the controller).

8.2   **DRIVECOM parameter data channel**

The DRIVECOM parameter data channel ...

► enables parameter setting and diagnostics of the controller.

► allows access to all Lenze parameters (codes).

► additionally occupies 4 words of the input and output data words in the master.

► has an identical structure for both directions of transmission.

8.2.1   **Addressing of the parameter data**

The parameter data is accessed via codes listed in the code table included in this documentation of the function module and the corresponding documentation of your controller.

8.2.2   **Addressing of the Lenze parameters**

In the case of the DRIVECOM parameter data channel the parameters of a device are not directly addressed via Lenze code numbers, but via indexes (byte 3, byte 4) and subindexes (byte 2).

The Lenze code numbers are converted into indexes via an offset (24575_{dec} / $5FFF_{hex}$):

Addressing of Lenze codes	Example for C0001 (operating mode)
● PROFIBUS index = 24575 - Lenze code	● PROFIBUS index = 24575 - 1 = 24574
● PROFIBUS-DP-Index$_{hex}$ = $5FFF_{hex}$ - Lenze code$_{hex}$	● PROFIBUS-DP-Index$_{hex}$ = $5FFF_{hex}$ - 1_{hex} = $5FFE_{hex}$

Lenze parameters are mainly represented in the fixed point format (data type integer32 with four decimal digits). For this reason, the value of the parameter/code must be multiplied by 10000 in order to obtain integer values.

The parameter value is entered in the user data (bytes 5 ... 8) of the telegram.

Example:

Set C0039 (JOG) = 150.4 Hz.

► 150.4 x 10000 = 1504000 ($0016F300_{hex}$)

► The resulting parameter value is entered in the user data.

8.2.3   **Telegram structure**
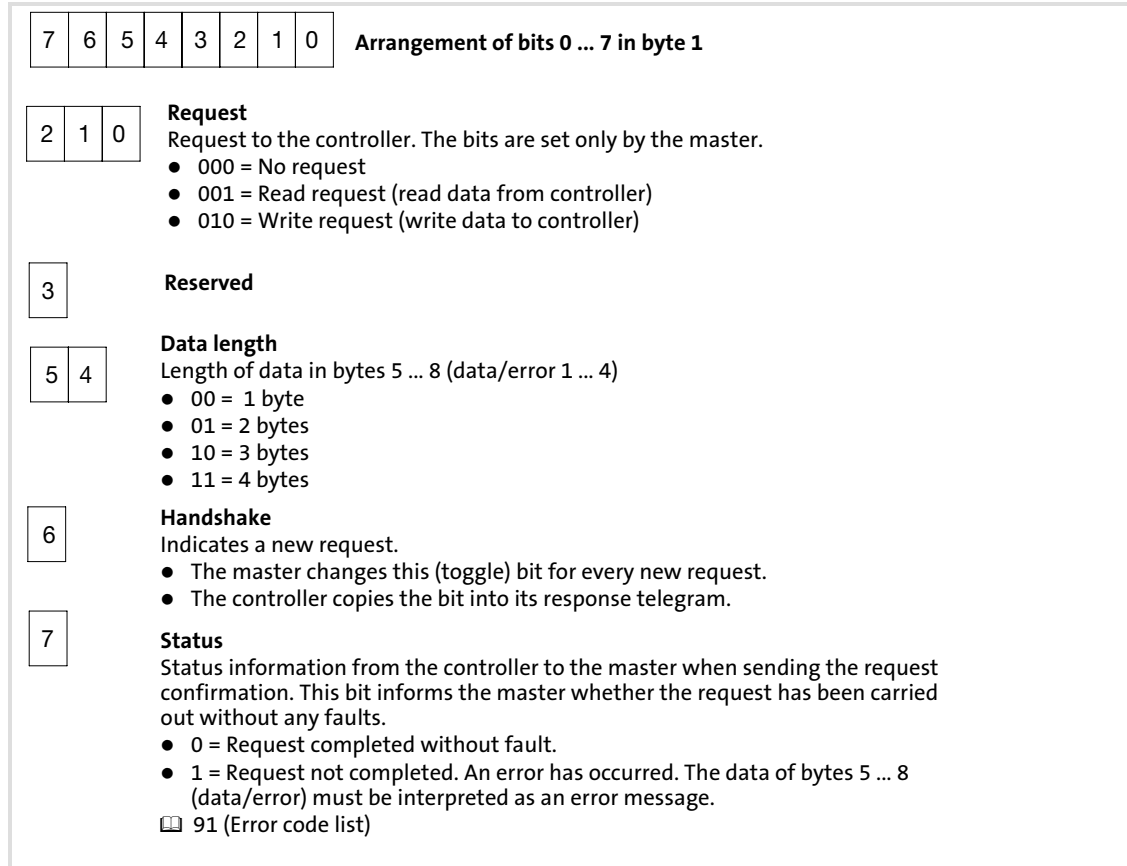
The telegram of the DRIVECOM parameter data channel consists of a total of 8 bytes. The individual bytes are described in detail on the following pages.

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	Index High byte	Index Low byte	Data 4 / Error 4	Data 3 / Error 3	Data 2 / Error 2	Data 1 / Error 1

8

Parameter data transfer
DRIVECOM parameter data channel
Telegram structure

Byte 1: Service, request and response control for the parameter data channel

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	Index High byte	Index Low byte	Data 4 / Error 4	Data 3 / Error 3	Data 2 / Error 2	Data 1 / Error 1

7	6	5	4	3	2	1	0	**Arrangement of bits 0 ... 7 in byte 1**

2	1	0

Request
Request to the controller. The bits are set only by the master.
- 000 = No request
- 001 = Read request (read data from controller)
- 010 = Write request (write data to controller)

3

Reserved

5	4

Data length
Length of data in bytes 5 ... 8 (data/error 1 ... 4)
- 00 = 1 byte
- 01 = 2 bytes
- 10 = 3 bytes
- 11 = 4 bytes

6

Handshake
Indicates a new request.
- The master changes this (toggle) bit for every new request.
- The controller copies the bit into its response telegram.

7

Status
Status information from the controller to the master when sending the request confirmation. This bit informs the master whether the request has been carried out without any faults.
- 0 = Request completed without fault.
- 1 = Request not completed. An error has occurred. The data of bytes 5 ... 8 (data/error) must be interpreted as an error message.

 91 (Error code list)

Examples of byte 1:

► Read request

Bit 7			...				Bit 0
0	X	1	1	0	0	0	1

"1" (read)
Reserved
"3" (data length 4 bytes)
Handshake
Status (only relevant for response telegram)

► Write request

Bit 7			...				Bit 0
0	X	0	1	0	0	1	0

"2" (write)
Reserved
"1" (data length 2 bytes)
Handshake
Status (only relevant for response telegram)
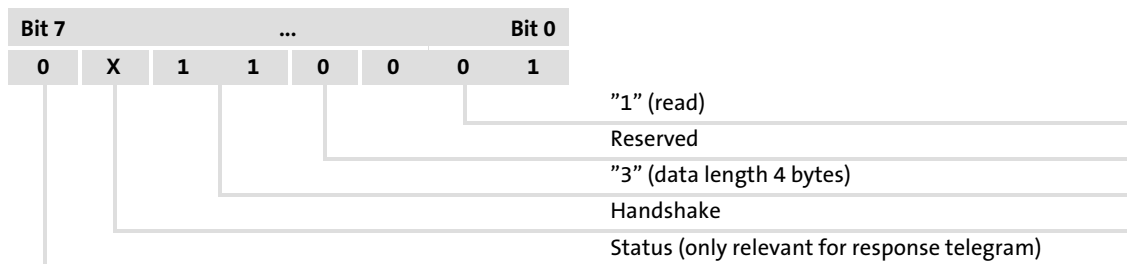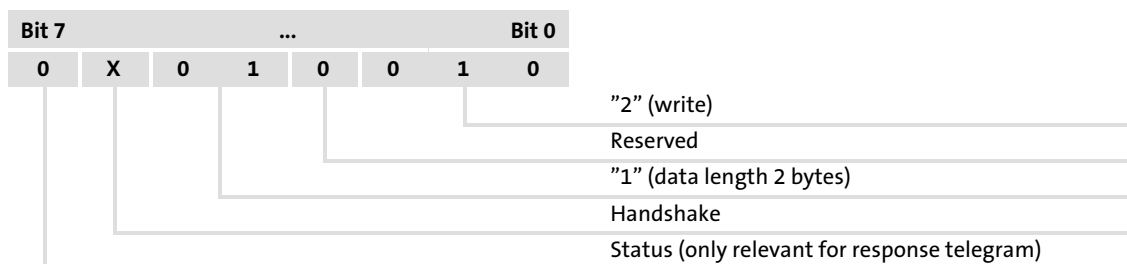
Byte 2: Subindex

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	**Subindex**	Index High byte	Index Low byte	Data 4 / Error 4	Data 3 / Error 3	Data 2 / Error 2	Data 1 / Error 1

► The 82XX series do not have codes with subindex, the value is always 0.

► For the device series 93XX and 8200 vector, an additional addressing via the subindex is required for those codes that have a subcode (see code table).

Example:

Code C0039 / subcode 3 addresses "NSET JOG" (50 % = Lenze setting)

Byte 3 / 4: index

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	**Index High byte**	**Index Low byte**	Data 4 / Error 4	Data 3 / Error 3	Data 2 / Error 2	Data 1 / Error 1

The parameter or the Lenze code is selected with these two bytes according to the formula:

Index = 24575 - Lenze code number

Example:

The parameter C0012 (acceleration time) is to be addressed:

► 24575 - 12 = 24563 = $5FF3_{hex}$

► Entry in byte 3 (high byte): $5F_{hex}$

► Entry in byte 4 (low byte): $F3_{hex}$

8 **Parameter data transfer**
DRIVECOM parameter data channel
Telegram structure

Bytes 5 ... 8: Parameter value (data) / error information (error)

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	Index High byte	Index Low byte	**Data 4 / Error 4**	**Data 3 / Error 3**	**Data 2 / Error 2**	**Data 1 / Error 1**

The status of the (status) bit 7 in byte 1 (job) determines the meaning of this data field:

Meaning of the bytes 5 ... 8 if ...	
Bit 7 = 0	**Bit 7 = 1**
Parameter value (data 1 ... 4)	Error information (error 1 ... 4) for an invalid access. 📖 91 (Error code list)

Parameter value (data)

Depending on the data format, the length of the parameter value is between 1 to 4 bytes. Data are saved in the Motorola format, i. e. first the high byte or high word, then the low byte or low word.

Byte 5	Byte 6	Byte 7	Byte 8
High byte	Low byte	High byte	Low byte
High word		Low word	
Double word			

Assignment of bytes 5 .. 8 with parameter values of different lengths

Byte 5	Byte 6	Byte 7	Byte 8
Parameter value (Length 1)	00	00	00
Parameter value (length 2)		00	00
Parameter value (length 4)			

> ℹ️ **Note!**
>
> Strings or data blocks cannot be transmitted.

Lenze

8.2.4 Error codes (DRIVECOM)

Data 1	Data 2	Data 3	Data 4	Meaning
0x06	0x03	0x00	0x00	No right to access
0x06	0x05		0x10	Impermissible job parameter
0x06	0x05		0x11	Invalid subindex
0x06	0x05		0x12	Data length too large
0x06	0x05		0x13	Data length too small
0x06	0x06		0x00	Object is no parameter
0x06	0x07		0x00	Object does not exist
0x06	0x08		0x00	Data types do not correspond
0x08	0x00		0x00	Job cannot be executed
0x08	0x00		0x20	Job cannot be executed at the moment
0x08	0x00		0x21	Not executable because of local control
0x08	0x00		0x22	Not executable because of device status
0x08	0x00		0x30	Out of value range/parameter can only be changed with inhibited controller
0x08	0x00		0x31	Parameter value too large
0x08	0x00		0x32	Parameter value too small
0x08	0x00		0x33	Subparameter out of value range
0x08	0x00		0x34	Subparameter value too large
0x08	0x00		0x35	Subparameter value too small
0x08	0x00		0x36	Maximum value smaller than minimum value
0x08	0x00		0x41	Communication object cannot be mapped on process data
0x08	0x00		0x42	Process data length exceeded
0x08	0x00		0x43	General collision with other values
0x08	0x00	0xFE	0x01	Invalid service (no read or write request)

8 **Parameter data transfer**
DRIVECOM parameter data channel
Reading parameters
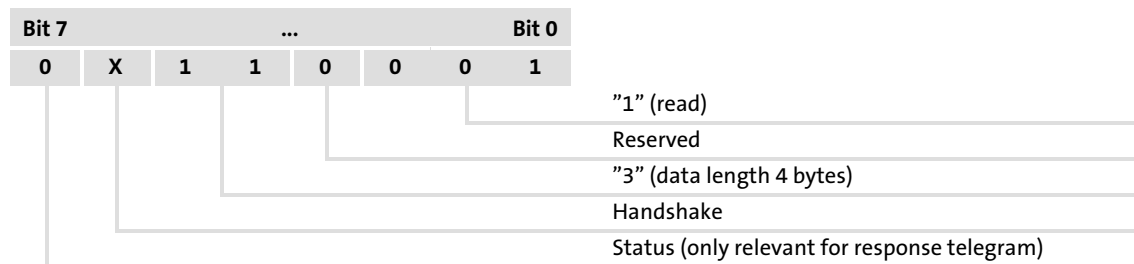
8.2.5 Reading parameters

General procedure

1. Define the user data range of the controller. (Where are the user data located in the host system?)
   Observe manufacturer-specific information.

2. Enter the address of the required parameter into the "Index" and "Subindex" fields (DP output data).

3. Request in the service byte = read request
   The status of the handshake bit in the service byte must be changed (DP output data).

4. Check whether the handshake bit in the service byte is the same for the DP input data and the DP output data.
   If the handshake bit is the same, the response has been received.
   It is useful to implement a time monitoring tool.

5. Check whether the status bit in the service byte is set.
   Status bit is not set: The "Data/Error" field contains the required parameter value.
   Status bit is set: The read request has not been executed correctly. The "Data/Error" field contains the error information.

Example:

The heatsink temperature (43 °C) of the controller is to be read (C0061).

▶ Byte 1: Request

Bit 7			...				Bit 0
0	X	1	1	0	0	0	1

"1" (read)
Reserved
"3" (data length 4 bytes)
Handshake
Status (only relevant for response telegram)

▶ Byte 2: Subindex
   Subindex = 0, as there is no subindex under code C0061.

▶ Byte 3 / 4: Index
   Index = 24575 - code number
   Index = 24575 - 61 = 24514 = $5FC2_{hex}$ ($5F_{hex}$ = high byte, $C2_{hex}$ = low byte)

▶ Bytes 5 ... 8: Data (contained in the response telegram)
   Data 1 ... 4 = 43 °C x 10000 = 430000 (FIX32) = $00068FB0_{hex}$

Result:

► Request telegram from master to drive:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	Index (High byte)	Index (Low byte)	Data 4	Data 3	Data 2	Data 1
01_{hex}	00_{hex}	$5F_{hex}$	$C2_{hex}$	00_{hex}	00_{hex}	00_{hex}	00_{hex}
00000001_{bin}	00000000_{bin}	01011111_{bin}	11000010_{bin}	00000000_{bin}	00000000_{bin}	00000000_{bin}	00000000_{bin}

Waiting for change of handshake bit in the response (bit 6 here: 0 → 1)

► Response telegram from drive to master (for error-free execution):

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	Index (High byte)	Index (Low byte)	Data 4	Data 3	Data 2	Data 1
30_{hex}	00_{hex}	$5F_{hex}$	$C2_{hex}$	00_{hex}	06_{hex}	$8F_{hex}$	$B0_{hex}$
00110000_{bin}	00000000_{bin}	01011111_{bin}	11000010_{bin}	00000000_{bin}	00000110_{bin}	10001111_{bin}	10110000_{bin}

8 **Parameter data transfer**
DRIVECOM parameter data channel
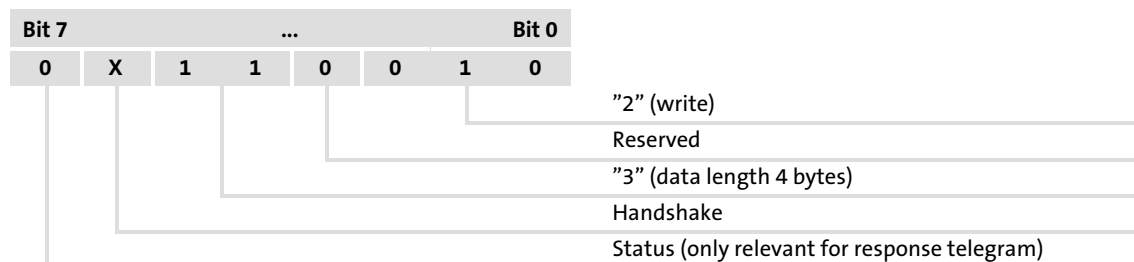Writing parameters

8.2.6 **Writing parameters**

General procedure

1. Define the user data range of the controller. (Where are the user data located in the host system?)

   Observe manufacturer-specific information.

2. Enter the address of the required parameter into the "Index" and "Subindex" fields (DP output data).

3. Enter the parameter value into the "Data/Error" field.

4. Request in the service byte = write request

   The status of the handshake bit in the service byte must be changed (DP output data).

5. Check whether the handshake bit in the service byte is the same for the DP input data and the DP output data.

   If the handshake bit is the same, the response has been received.

   It is useful to implement a time monitoring tool.

6. Check whether the status bit in the service byte is set.

   Status bit is not set: The write request has been executed correctly.

   Status bit is set: The write request has not been executed correctly. The "Data/Error" field contains the error information.

Example:

The acceleration time (C0012) of the controller is to be set to 20 s.

▶ Byte 1: Request

Bit 7			...				Bit 0
0	X	1	1	0	0	1	0

- "2" (write)
- Reserved
- "3" (data length 4 bytes)
- Handshake
- Status (only relevant for response telegram)

▶ Byte 2: Subindex

   Subindex = 0, as there is no subindex under code C0012.

▶ Byte 3 / 4: Index

   Index = 24575 - code number

   Index = 24575 - 12 = 24563 = $5FF3_{hex}$ ($5F_{hex}$ = high byte, $F3_{hex}$ = low byte)

▶ Bytes 5 ... 8: data

   Data 1 ... 4 = 20 s x 10000 = 200000 (FIX32) = $00030D40_{hex}$

Result:

► Request telegram from master to drive:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	Index (High byte)	Index (Low byte)	Data 4	Data 3	Data 2	Data 1
72_{hex}	00_{hex}	$5F_{hex}$	$F3_{hex}$	00_{hex}	03_{hex}	$0D_{hex}$	40_{hex}
01110010_{bin}	00000000_{bin}	01011111_{bin}	11110011_{bin}	00000000_{bin}	00000011_{bin}	00001101_{bin}	01000000_{bin}

Waiting for change of handshake bit (bit 6 here: 0 → 1)

► Response telegram from drive to master (for error-free execution):

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	Index (High byte)	Index (Low byte)	Data 4	Data 3	Data 2	Data 1
40_{hex}	00_{hex}	$5F_{hex}$	$F3_{hex}$	00_{hex}	00_{hex}	00_{hex}	00_{hex}
01000110_{bin}	00000000_{bin}	01011111_{bin}	11110011_{bin}	00000000_{bin}	00000000_{bin}	00000000_{bin}	00000000_{bin}

Waiting for change of handshake bit (bit 6 here: 1 → 0)

8.3   PROFIdrive parameter data channel

Data communication with PROFIBUS-DP-V0 is characterised by cyclic diagnostics and cyclic process data and parameter data transfer.

An optional service extension is the acyclic parameter data transfer of PROFIBUS-DP-V1. This service does not impair the functionality of the standard services under PROFIBUS-DP-V0.

PROFIBUS-DP-V0 and PROFIBUS-DP-V1 can be operated simultaneously in the same network. This enables the step-by-step expansion or modification of a system.

The services of PROFIBUS-DP-V1 can be used by the master class 1 (PLC) and the master class 2 (diagnostics master etc.).

The integration of the acyclic service into the fixed bus cycle depends on the corresponding configuration of the master class 1:

▶ For an existing configuration a *time slot is reserved*.

▶ When there is no configuration, the acyclic service is *appended* when a master class 2 acyclically accesses a DP-V1 slave.

Access to the Lenze codes of the controller

The codes of the first parameter set (C0000 ... C1999) can be accessed directly. A conversion is not required.

Entering a parameter value

The required parameter value is mapped in the data range.

Lenze parameters are mainly represented in the fixed point format with four places after the decimal point (data type FIX32, transmission as double word). These parameters are multiplied by 10000 to obtain integer values.

Example:

Set C0039 (JOG) = 150.4 Hz.

▶ 150.4 x 10000 = 1504000 (0016F300$_{hex}$)

8.3.1 PROFIdrive DP-V1

Features

► Parameter number and subindex addresses with a width of 16 bits each.

► Several parameter requests can be combined to one request (multi-parameter requests).

► Processing of one parameter request at a time (no pipelining).

► A parameter request or a parameter response must fit into one data block (max. 240 bytes). Requests/responses cannot be split over several data blocks.

► Spontaneous messages are not transmitted.

► There are only acyclic parameter requests.

► Profile-specific parameters can be read independently of the slave state.

8.3.1.1 Establishing a connection between master and slave

A class 1 master can always be used to request parameters from a slave if the slave is in the "Data_Exchange" state.

In addition to the class 1 master connection, a class 2 master can establish a communication connection to the slave:
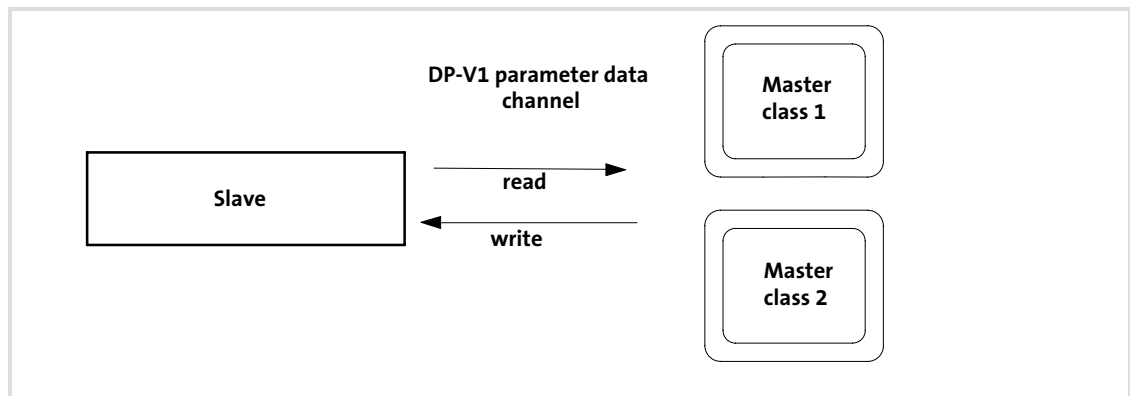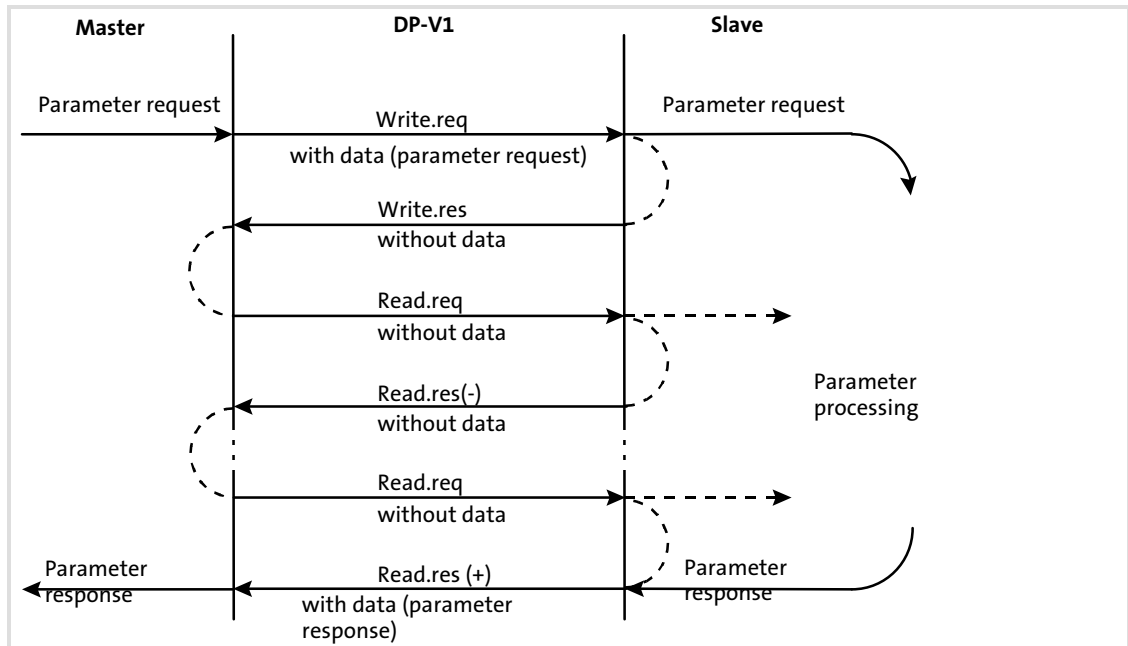

Fig. 8-1   Data communication via the DP-V1 parameter data channel

8.3.1.2 Acyclic data transfer

> **Note!**
>
> A parameter request refers to one or several parameter(s) (multi-parameter request).

Sequence:

► A "Write.req" is used to pass the data set (DB47) to the slave in the form of a parameter request.

► With "Write.res" the master receives the confirmation for the receipt of the message.

► The master requests the response of the slave with "Read.req".

► The slave responds with "Read.res (-)" if processing has not yet been completed.

► After parameter processing, the parameter request is completed by transmitting the parameter response to the master with "Read.res (+)".

Lenze

8.3.1.3 Telegram structure

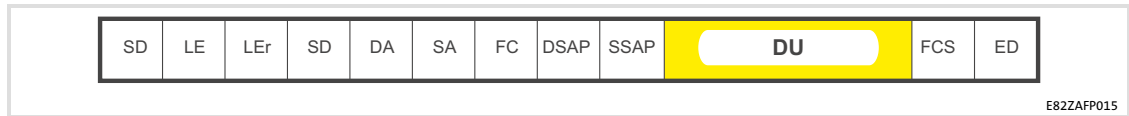| SD	LE	LEr	SD	DA	SA	FC	DSAP	SSAP	DU	FCS	ED

E82ZAFP015

Fig. 8-2   PROFIBUS data telegram with DP-V1

The data unit (DU) contains the DP-V1 header and the parameter request or the parameter response.

In the following subchapters, the parameter request and the parameter response are described in detail.

> **Note!**
>
> The DP-V1 header consists of:
> ► Function identifier
> ► Slot number
> ► Data set
> ► Length of the user data
>
> Please refer to the corresponding PROFIBUS specification for further information on the DP-V1 header.

8   **Parameter data transfer**
PROFIdrive parameter data channel
PROFIdrive DP-V1

8.3.1.4   Reading parameters

> ### 👤 Note!
>
> ► When a read request is processed, no parameter value is written to the slave.
> ► A response to a read request does not contain the parameter attribute, index and subindex.
> ► When a multi-parameter read request is transferred, the parameter attribute, index and subindex are repeated according to the number "n" of the parameters requested.
> ► A read request must not exceed the maximum data length of 240 bytes.

Request header

Byte 1	Byte 2	Byte 3	Byte 4
U8	U8	U8	U8
Request reference	Request identification	Axis	Number of indexes

Request reference:	This value is specified by the master
Request identification:	0x01 (request parameter for reading)
Axis:	0x00 or 0x01
Number of indexes:	0x"n" (number of parameters requested)

Parameter attribute

Byte 5	Byte 6
U8	U8
Attribute	Number of subindexes

Attribute:	0x10 (value)
Number of subindexes:	0x00
	● For array parameters enter the number of array parameters requested.

Index and subindex

Byte 7	Byte 8	Byte 9	Byte 10
	U16		U16
Index		Subindex	

Index:	0x0001 ... 0xFFFF (1 ... 65535)
Subindex:	0x0001 ... 0xFFFF (1 ... 65535)
	● 0x0000 for all non-array parameters

Parameter data transfer **8**
PROFIdrive parameter data channel
PROFIdrive DP-V1

8.3.1.5 **Response to a correctly executed read request**

> 🛈 **Note!**
> ► When a read request is processed, no parameter value is written to the slave.
> ► A response to a read request does not contain the parameter attribute, index and subindex.
> ► When a multi-parameter read request is transferred, the parameter format and parameter value are repeated according to the number "n" of parameters requested.
> ► A read request must not exceed the maximum data length of 240 bytes.

Response header

Byte 1	Byte 2	Byte 3	Byte 4
U8	U8	U8	U8
Request reference (mirrored)	Response identification	Axis (mirrored)	Number of indexes

Request reference:	Mirrored value of parameter request
Response identification:	0x01 (parameter has been read)
Axis:	0x00 or 0x01
Number of indexes:	0x"n" (number of parameters requested)

Parameter format

Byte 5	Byte 6
U8	U8
Format	Number of values

Format:	0x01 ... 0x36, data types
	0x41, byte
	0x42, word
	0x43, double word
Number of values:	0x01 or
	number of subindexes requested
	● If there is more than one subindex, only the parameter value is repeated.

8   **Parameter data transfer**
PROFIdrive parameter data channel
PROFIdrive DP-V1

Parameter value

Depending on the data type, the user data are assigned as follows:

Data type	Length	Assignment of the user data				
		Byte 7	Byte 8	Byte 9	Byte 10	Byte ...
String	x bytes					
U8	1 byte		00			
U16	2 bytes	High byte	Low byte			
U32	4 bytes	High word		Low word		
		High byte	Low byte	High byte	Low byte	

(This representation applies to one parameter value.)

Parameter data transfer 8
PROFIdrive parameter data channel
PROFIdrive DP-V1

8.3.1.6 Response to a read request error

Response header

Byte 1	Byte 2	Byte 3	Byte 4
U8	U8	U8	U8
Request reference (mirrored)	Response identification	Axis (mirrored)	Number of indexes

Request reference:	Mirrored value of parameter request
Response identification:	0x81 (read error)
	An error code is transmitted (see below).
Axis:	0x00 or 0x01
Number of indexes:	0x"n" (number of parameters requested)

Parameter format

Byte 5	Byte 6
U8	U8
Format	Number of values

Format:	0x44 (error)
Number of values:	0x01 (error code without additional information)
	0x02 (error code with additional information)

Error code

Byte 7	Byte 8	Byte 9	Byte 10
	U16		U16
Error code		Additional information (if available)	

Error code:	0x0000 ... 0x00FF
	📖 109 (Error code list)
(Additional information)	

8   **Parameter data transfer**
PROFIdrive parameter data channel
PROFIdrive DP-V1

8.3.1.7   Writing parameters

> **i**   **Note!**
>
> ▶ When a multi-parameter write request is processed, the ...
> – parameter attribute
> – index and subindex
> and then the
> – parameter format and
> – parameter value
> are repeated according to the number "n" of parameters requested.
>
> ▶ A write request must not exceed the maximum data length of 240 bytes.

Request header

Byte 1	Byte 2	Byte 3	Byte 4
U8	U8	U8	U8
Request reference	Request identification	Axis	Number of indexes

Request reference:	This value is specified by the master
Request identification:	0x02 (write parameter)
Axis:	0x00 or 0x01
Number of indexes:	0x"n" (number of parameters requested)

Parameter attribute

Byte 5	Byte 6
U8	U8
Attribute	Number of subindexes

Attribute:	0x10, value
Number of subindexes:	0x00
	● For array parameters enter the number of array parameters requested.

Index and subindex

Byte 7	Byte 8	Byte 9	Byte 10
	U16		U16
Index		Subindex	

Index:	0x0001 ... 0xFFFF (1 ... 65535)
Subindex:	0x0001 ... 0xFFFF (1 ... 65535)
	● 0x0000 for all non-array parameters

Parameter format

Byte 11	Byte 12
U8	U8
Format	Number of values

Format: 0x01 ... 0x36, data types

0x41, byte

0x42, word

0x43, double word

Number of values: 0x01 or

number of subindexes requested

- If there is more than one subindex, only the parameter value is repeated.

Parameter value

Depending on the data type, the user data are assigned as follows:

		Assignment of the user data				
Data type	Length	Byte 13	Byte 14	Byte 15	Byte 16	Byte ...
String	x bytes					
U8	1 byte		00			
U16	2 bytes	High byte	Low byte			
U32	4 bytes	High word: High byte	High word: Low byte	Low word: High byte	Low word: Low byte	

(This representation applies to one parameter value.)

8.3.1.8 Response to a correctly executed write request

Response header

Byte 1	Byte 2	Byte 3	Byte 4
U8	U8	U8	U8
Request reference (mirrored)	Response identification	Axis (mirrored)	Number of indexes

Request reference: Mirrored value of parameter request

Response identification: 0x02 (parameter has been written)

Axis: 0x00 or 0x01

Number of indexes: 0x"n" (number of parameters requested)

8 **Parameter data transfer**
PROFIdrive parameter data channel
PROFIdrive DP-V1

8.3.1.9 Response to a write request error

ℹ️ **Note!**

For a multi-parameter request, the correct and possibly faulty messages are combined in one telegram. The individual messages have the following data contents:

► Correct message
  – Format: 0x40 (zero)
  – Number of values: 0x00
► Faulty message
  – Format: 0x44
  – Number of values: 0x01 or 0x02
  – Error code without additional information (for number of values = 0x01) or
  – error code with additional information (for number of values = 0x02)

A faulty access to a parameter "n" is indicated at the nth position in the response telegram of a multi-parameter request.

Response header

Byte 1	Byte 2	Byte 3	Byte 4
U8	U8	U8	U8
Request reference (mirrored)	Response identification	Axis (mirrored)	Number of indexes

Request reference:	Mirrored value of parameter request
Response identification:	0x82 (write error)
	An error code is transmitted, see below
Axis:	0x00 or 0x01
Number of indexes:	0x"n" (number of parameters requested)

Parameter format

Byte 5	Byte 6
U8	U8
Format	Number of values

Format:	0x44, error
Number of values:	0x01 (error code without additional information)
	0x02 (error code with additional information)

Error code

Byte 7	Byte 8	Byte 9	Byte 10
	U16		U16
Error code		Additional information if available	

Error code:	0x0000 ... 0x00FF
	📖 109 (Error code list)
(Additional information)	

8.3.1.10 Programming of read requests

Procedure

1. Define the user data range of the controller (define the location of the user data in the host system).

   Observe manufacturer-specific data.

2. Enter the code of the desired parameter into the "code" field (output data).

3. Job identification / service = read request

4. Check whether index and subindex correspond with the job and whether the job identification is $\neq 0$:

   – If the criteria are fulfilled, the desired controller data from the field "Parameter value" are transmitted to the master.

   – If these criteria are not fulfilled, the response identifier is negative (high nibble of byte 1 = 7_{hex}). In this case, the error information can be read out from the entry in the low word.

Example:

The heatsink temperature (43 °C) of the controller is to be read (C0061).

► Job identification (high nibble in byte 1)
   – Read simple parameter: "1"

► Code: (low nibble in byte 1 and byte 2)
   – C0061: 61 = $3D_{hex}$

► Lenze subcode (byte 3):
   – Subindex = 0, as there is not subindex under code C0061.

► Bytes 5 ... 8: Data (not contained in the request telegram)
   – Data 1 ... 4 = 43° C x 10000 = 430000 = $00068FB0_{hex}$

Result:

► Request telegram from master to drive:

Byte 1*	Byte 1* +2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
AK	Code	Subcode	Reserved	Parameter value			
1_{hex}	$03D_{hex}$	00_{hex}	00_{hex}	00_{hex}	00_{hex}	00_{hex}	00_{hex}
0001_{bin}	000000111101_{bin}	00000000_{bin}	00000000_{bin}	00000000_{bin}	00000000_{bin}	00000000_{bin}	00000000_{bin}
			Wait for response identification with code = $03D_{hex}$ and subcode 0				

► Response telegram from drive to master (for faultless execution):

Byte 1*	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
AK	Code	Subcode	Reserved	Parameter value			
2_{hex}	$03D_{hex}$	00_{hex}	00_{hex}	00_{hex}	06_{hex}	$8F_{hex}$	$B0_{hex}$
0010_{bin}	000000111101_{bin}	00000000_{bin}	00000000_{bin}	00000000_{bin}	00000110_{bin}	10001111_{bin}	10110000_{bin}

8   **Parameter data transfer**
PROFIdrive parameter data channel
PROFIdrive DP-V1

8.3.1.11   Programming of write requests

Procedure

1. Define the user data range of the controller (define the location of the user data in the host system).

   Observe manufacturer-specific data.

2. Enter the code of the desired parameter into the "code" field (output data).

3. Enter parameter value into the "Data/Error" field.

4. Job identification / service = write request

5. Check whether index and subindex correspond with the job and whether the job identification is $\neq 0$:

   – If the criteria are fulfilled, the desired master data from the field "Parameter value" are accepted by the controller.

   – If these criteria are not fulfilled, the response identifier is negative (high nibble of byte 1 = 7_{hex}). In this case, the error information can be read out from the entry in the low word.

Example:

The controller acceleration time (C0012) is to be set to 20 s.

► Job identification (high nibble in byte 1)

   Transmit simple parameter value: "1"

► Code: (low nibble in byte 1 and byte 2)

   C0012: 12 = $0C_{hex}$

► Lenze subcode (byte 3):

   Subindex = 0, as there is not subindex under code C0012.

► Bytes 5 ...8: Data

   Data 1 ... 4 = 20 s x 10000 = 200000 = $00030D40_{hex}$

Result:

► Request telegram from master to drive:

Byte 1*	Byte 1* +2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
AK	Code	Subcode	Reserved	Parameter value			
3_{hex}	$00C_{hex}$	00_{hex}	00_{hex}	00_{hex}	03_{hex}	$0D_{hex}$	$40hex$
0011_{bin}	000000001100_{bin}	00000000_{bin}	00000000_{bin}	00000000_{bin}	00000011_{bin}	00001101_{bin}	01000000_{bin}
			Wait for response identification with code = 00C and subcode 0				

► Response telegram from drive to master (for faultless execution):

Byte 1*	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
AK	Code	Subcode	Reserved	Parameter value			
2_{hex}	$00C_{hex}$	00_{hex}	00_{hex}	00_{hex}	00_{hex}	00_{hex}	00_{hex}
0010_{bin}	000000001100_{bin}	00000000_{bin}	00000000_{bin}	00000000_{bin}	00000000_{bin}	00000000_{bin}	00000000_{bin}

8.3.2 Error codes (PROFIdrive)

Error code	Meaning	Description	Additional info
0x0000	Impermissible parameter number	Access to unavailable parameter	-
0x0001	Parameter value cannot be changed	Change access to a parameter value that cannot be changed	Subindex
0x0002	Low or high limit exceeded	Change access with value outside the value limits	Subindex
0x0003	Faulty subindex	Access to unavailable subindex	Subindex
0x0004	No array	Access with subindex to non-indexed parameter	-
0x0005	Incorrect data type	Change access with value that does not match the data type of the parameter	-
0x0006	Setting not permitted (can only be reset)	Change access with value unequal to 0 where this is not permitted	Subindex
0x0007	Description element cannot be changed	Change access to a description element that cannot be changed	Subindex
0x0008	Reserved	(PROFIdrive profile V2: PPO-write requested in IR not available)	-
0x0009	No description data available	Access to unavailable description (parameter value is available)	-
0x000A	Reserved	(PROFIdrive profile V2: Access group wrong)	-
0x000B	No operation priority	Change access without rights to change parameters	-
0x000C	Reserved	(PROFIdrive profile V2: Wrong password)	-
0x000D	Reserved	(PROFIdrive profile V2: Text cannot be read in cyclic data transfer)	-
0x000E	Reserved	(PROFIdrive profile V2: Name cannot be read in cyclic data transfer)	-
0x000F	No text array available	Access to text array that is not available (parameter value is available)	-
0x0010	Reserved	(PROFIdrive profile V2: No PPO-write)	-
0x0011	Request cannot be executed because of operating status	Access is temporarily not possible for reasons that are not specified in detail	-
0x0012	Reserved	(PROFIdrive profile V2: Other error)	-
0x0013	Reserved	(PROFIdrive profile V2: Data cannot be read in cyclic interchange)	-
0x0014	Value impermissible	Change access with a value that is within the value limits but is not permissible for other long-term reasons (parameter with defined single values)	Subindex
0x0015	Response too long	The length of the current response exceeds the maximum transmittable length	
0x0016	Parameter address impermissible	Illegal value or value which is not supported for the attribute, number of subindexes, parameter number or subindex or combination	
0x0017	Illegal format	Write request: Illegal format or format of the parameter data which is not supported	
0x0018	Number of values not consistent	Write request: Number of values of the parameter data do not match the number of subindexes in the parameter address	
0x0019 ... 0x0064	Reserved	-	-
0x0065 ... 0x00FF	Manufacturer-specific	-	-

8.4 Consistent parameter data

In the PROFIBUS communication system, data are permanently exchanged between the host (**CPU + PROFIBUS master**) and the standard device via the plugged-on slave interface module.

Both the PROFIBUS master and the CPU (central processing unit) of the host access a joint memory - the dual port memory (DPM).

The DPM allows data exchange in both directions (write/read):
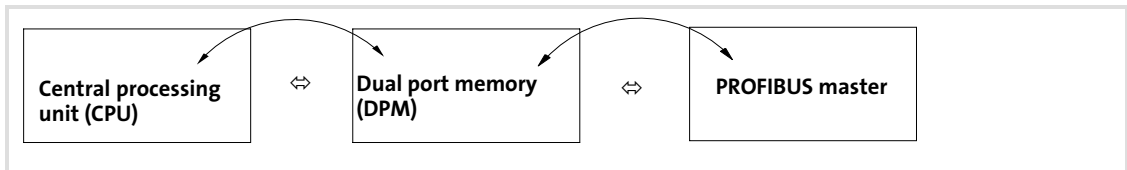
It could happen that a slower PROFIBUS master writing would be overtaken by a faster CPU reading within a cycle time without any further data organisation.

To avoid such an impermissible state, the parameter data to be transmitted must be marked as "consistent".

Data communication with existing consistency

With consistency, either "reading" or "writing" is possible when the master and the CPU simultaneously access the memory:

▶ The PROFIBUS master transfers data only as a complete data set.

▶ The CPU can only access completely updated data sets.

▶ The PROFIBUS master cannot read or write data as long as the CPU accesses consistent data.

The result becomes clear from the example below:
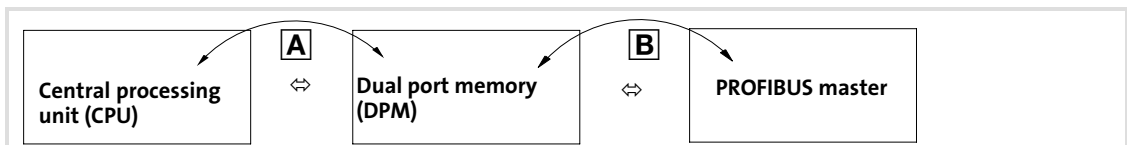
Ⓐ CPU wants to read!

Ⓑ PROFIBUS master wants to write simultaneously!

    1. As the PROFIBUS master can only write if the CPU does not read, the master has to wait until the data are read completely by the CPU.

    2. The PROFIBUS master only writes a complete data set into the DPM.

Configuring consistent data

Consistency is achieved by an appropriate PROFIBUS master configuration. Please refer to the corresponding documentation for your configuring software for this purpose.

Tip!

Consistency configuration depends on the PROFIBUS master configuring software. When using a Siemens-S5 PLC, please consider:

▶ Consistency is switched on by any word in the consistent area

▶ Consistency must be switched off by a specific switch-off word.

▶ The type of CPU and consistency and the address area determine which word switches off consistency.

9   Diagnostics

9.1   LED status displays

2133PFB003
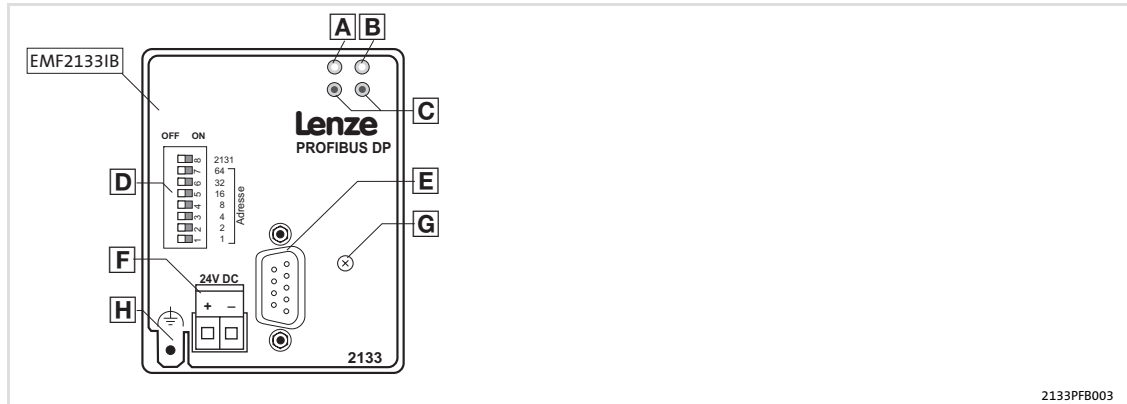
Pos.	Colour	Status	Description
A	green	on	The communication module is supplied with voltage and has a connection to the drive controller.
		off	The communication module is not supplied with voltage. The controller or the external voltage supply is switched off.
		blinking	The communication module is supplied with voltage, but is not (yet) connected to the controller because ... • the communication module has not been correctly attached to the controller; • data transfer to/from the controller is not yet possible (e.g. controller is in initialisation mode).
B	yellow	off	The communication module is not yet initialised.
		blinking	Communication via the communication module has been set up.
C	Red / green		The **red and green drive LED** indicates the operating status of the standard device (see documentation of the standard device).

Lenze

Diagnostics **9**
Troubleshooting and fault elimination
Controller is inhibited

9.2 Troubleshooting and fault elimination

9.2.1 Controller is inhibited

The controller cannot be enabled via PROFIBUS process data, i.e. the status "OPERATION ENABLED" is not reached.
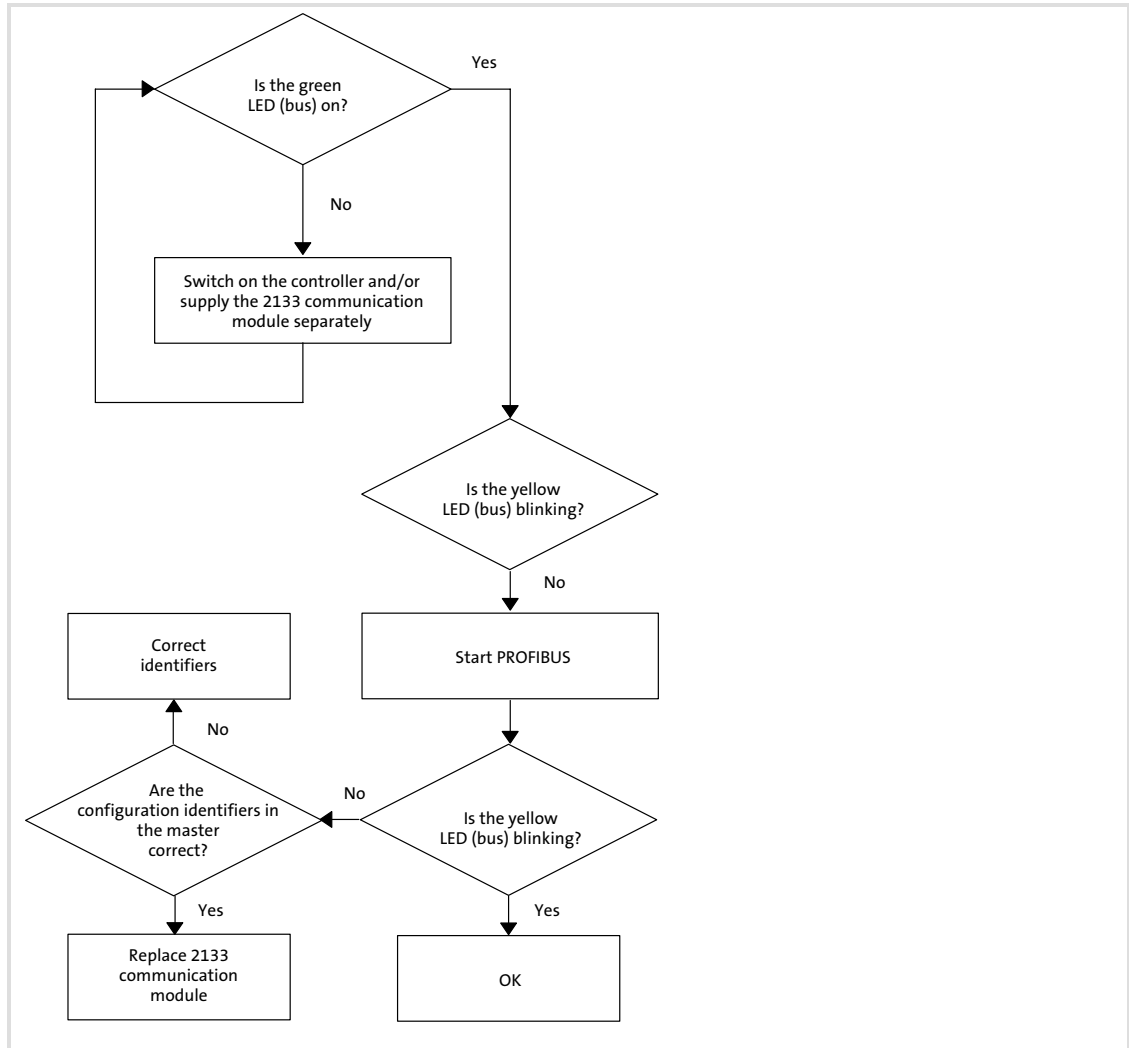
9 **Diagnostics**
Troubleshooting and fault elimination
Controller is inhibited

```
                        ( 1 )
                          │
                          ▼
              ┌───────────────────────┐
              │  Select POW1 = 007Ehex │
              └───────────────────────┘
                          │
                          ▼
                        ╱   ╲◄──────────────┐
                      ╱       ╲      No      │
                    ╱  PIW1 =   ╲────────────┘
                    ╲ xxxx xxxx x011 0001bin╱
                      ╲(ready to switch on)?╱
                        ╲   ╱
                          │ Yes
                          ▼
              ┌───────────────────────┐
              │  Select POW1 = 007Fhex │
              └───────────────────────┘
                          │
                          ▼
                        ╱   ╲                    ┌─────────────────────────────┐
                      ╱       ╲    No            │    Enable controller        │
                    ╱  PIW1 =   ╲────────────────▶│    (terminal 28, C40)       │
                    ╲ xxxx xxxx x011 0111bin      │ For 821X/8200 vector/822X   │
                      ╲(operation enabled)?╱      │ setpoint selection with     │
                        ╲   ╱                     │ C0106 <> 0                  │
                          │ Yes                   │ via POW2                    │
                          ▼                       └─────────────────────────────┘
              ┌───────────────────────┐
              │                       │      Setpoints or other control
              │         OK            │       signals can be selected
              │                       │
              └───────────────────────┘
```

9.2.2   **Checking PROFIBUS**

Short check of the PROFIBUS system in the case of faulty initialisation:

► Take into account the diagnostics information of the PROFIBUS communication module in the host.

► It might be useful for troubleshooting to reduce the bus to such an extent that only one device remains connected to the PROFIBUS.

```
        ┌──────────────┐                  ┌─────────────────────┐
        │ Are all      │      No          │ Switch on all       │
        │ devices      │─────────────────▶│ devices or          │
        │ connected    │                  │ disconnect all      │
        │ to PROFIBUS  │                  │ devices that are    │
        │ switched on? │                  │ switched off from   │
        └──────────────┘                  │ the PROFIBUS        │
             │ Yes                         └─────────────────────┘
             ▼
  ┌─────────────────────┐
  │ Reduce the number   │
  │ of devices          │
  │ connected to the    │
  │ PROFIBUS until the  │
  │ device in question  │
  │ is the last one     │
  │ connected           │
  └─────────────────────┘
             │
             ▼
        ┌──────────────┐      No          ┌─────────────────────┐
        │ PROFIBUS     │─────────────────▶│ Replace device      │
        │ initialisation│                  └─────────────────────┘
        │ OK?          │
        └──────────────┘
             │ Yes
             ▼
  ┌─────────────────────┐
  │ Reconnect the       │
  │ next device         │
  └─────────────────────┘
```
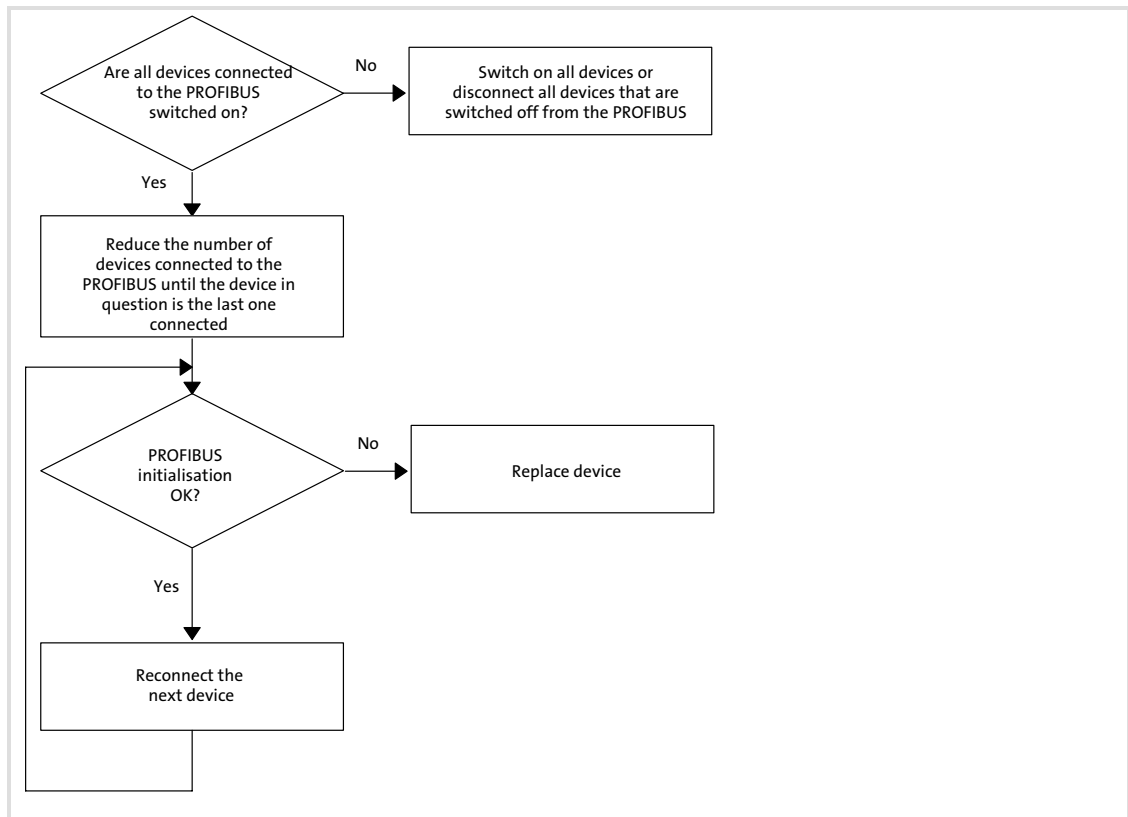
9.2.3 Activation of communication module

Activation of the communication module in conjunction with a controller:

```
                         ┌──────────────┐
                         │  Is the green│  Yes
                    ┌───◇│  LED (bus) on?│◇────────┐
                    │     └──────────────┘          │
                    │           │ No                │
                    │           ▼                   │
                    │   ┌──────────────────┐        │
                    │   │ Switch on the    │        │
                    │   │ controller and/or│        │
                    │   │ supply the 2133  │        │
                    │   │ communication    │        │
                    │   │ module separately│        │
                    │   └──────────────────┘        │
                    └───────────┘                   │
                                                    ▼
                                          ◇────────────────◇
                                          │  Is the yellow │
                                          │  LED (bus)     │
                                          │  blinking?     │
                                          ◇────────────────◇
                                                  │ No
                                                  ▼
    ┌──────────┐                          ┌──────────────┐
    │ Correct  │                          │ Start PROFIBUS│
    │identifiers│                         └──────────────┘
    └──────────┘                                  │
         ▲ No                                     ▼
    ◇────────────◇        No          ◇────────────────◇
    │  Are the   │◇──────────────────│  Is the yellow │
    │configuration│                   │  LED (bus)     │
    │identifiers in│                  │  blinking?     │
    │the master   │                   ◇────────────────◇
    │correct?     │                           │ Yes
    ◇────────────◇                            ▼
         │ Yes                        ┌──────────────┐
         ▼                            │      OK      │
    ┌──────────┐                      └──────────────┘
    │Replace 2133│
    │communication│
    │module      │
    └──────────┘
```

Lenze

9.2.4 Reset fault (TRIP)

Fault reset via PROFIBUS process data:

```
                    ┌─────────────────────┐
                    │    Select POW1 =     │
                    │     0080_hex         │
                    └─────────────────────┘
                              │
                              ▼
                          ╱───────╲
                        ╱   PIW1 =   ╲          Yes
                      ╱ xxxx xxxx x100 ╲──────────────┐
                      ╲  0000_bin       ╱             │
                        ╲(switch-on    ╱              │
                          ╲ inhibit)? ╱               │
                            ╲───────╱                 │
                              │ No                    │
        ┌─────────────────────┤                       │
        │                     ▼                        │
        │            ┌─────────────────────┐           │
        │            │     Selection:       │           │
        │            │   POW1 = 0000_hex    │           │
        │            │        then          │           │
        │            │   POW1 = 0080_hex    │           │
        │            └─────────────────────┘           │
        │                     │                         │
        │                     ▼                         │
        │                 ╱───────╲                     │
        │   No          ╱   PIW1 =   ╲                  │
        └─────────────╱ xxxx xxxx x100 ╲                │
                      ╲  0000_bin       ╱               │
                        ╲(switch-on    ╱                │
                          ╲ inhibit)? ╱                 │
                            ╲───────╱                   │
                              │ Yes                     │
                              ▼◄───────────────────────┘
                    ┌─────────────────────┐
                    │         OK           │
                    └─────────────────────┘
```

9 | **Diagnostics**
Monitoring with interrupted PROFIBUS communication
Permanent interruption of communication
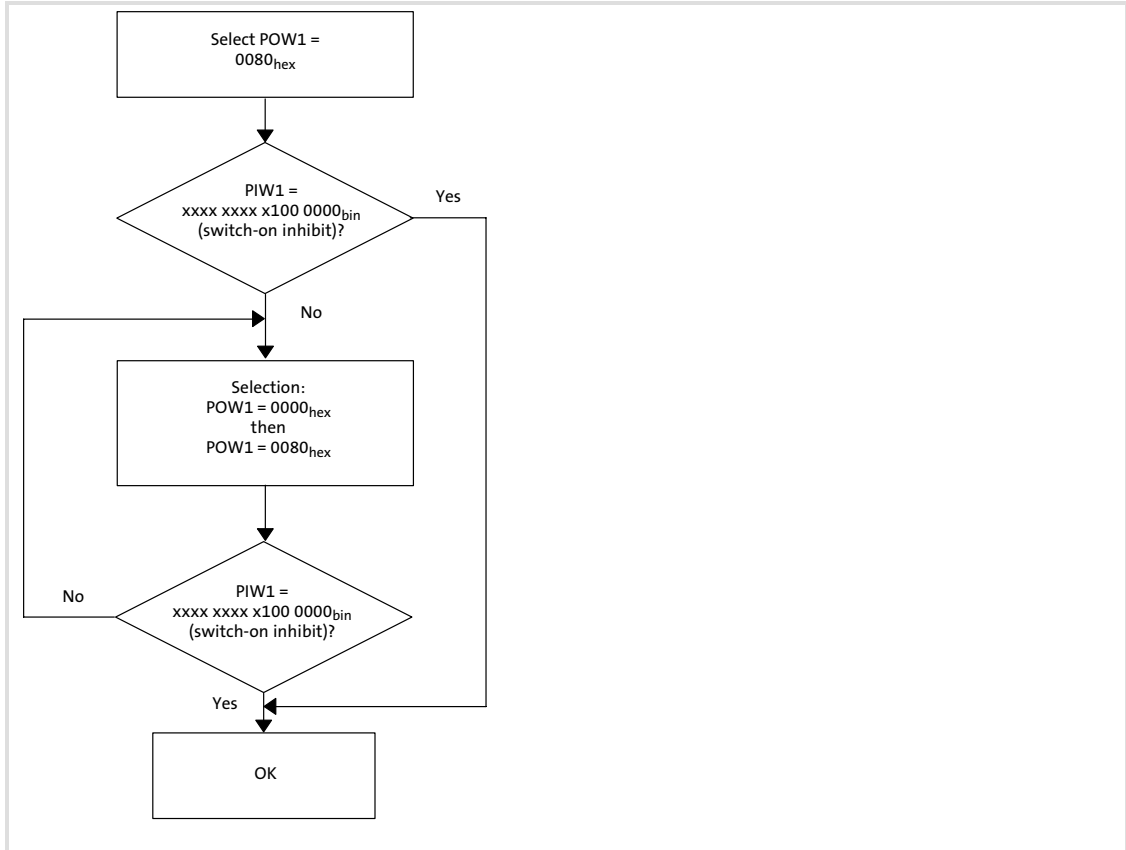
9.3 **Monitoring with interrupted PROFIBUS communication**

9.3.1 **Permanent interruption of communication**

If the PROFIBUS communication is interrupted permanently, e.g. by cable breakage or failure of the PROFIBUS master, no process data are transmitted to the slave in the "Data_Exchange" (DATA_EXCH) state.

After the watchdog monitoring time has expired, the response parameterised in code C1882 is effected.

The preconditions for a response by the slave are as follows:

1. The response has to be activated by the user by the selection "Fault", "Controller inhibit" (CINH), or quick stop "QSP".

2. The slave has to be in the "Data_Exchange" state.

3. The watchdog monitoring time in the master has to be configured correctly by the user.

If one of these preconditions is not met, there is no response to the absence of cyclic process data telegrams of the master.

Lenze

9.3.2 Short-time interruption of communication

2133PFB006
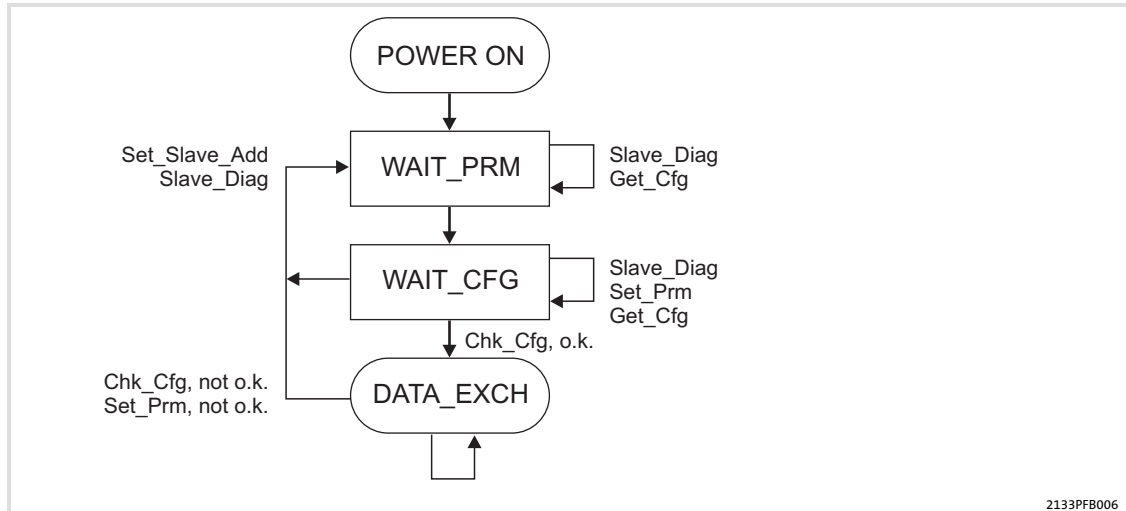
The master detects the communication fault and only after a few microseconds transfers the slave to the "WAIT_PRM" state of the DP state machine (see above).

Only after the state chain of the DP state machine meeting in the "Data_Exchange" (DATA_EXCH) state has been passed through, the watchdog monitoring time calculated for the slave (in milliseconds) goes on.

The watchdog monitoring time does *not* continue if the slave in the case of a repeated communication fault (e.g. by loose contact) does not reach the "Data_Exchange" state.

Code C1883 provides additional monitoring which already becomes active when exiting "Data_Exchange" after the parameterised time has expired (0 ... 65534 ms) and which actuates the response parameterised in code C1882.

Tip!

Be sure to observe the following precondition with regard to time setting:
Response time ≤ response monitoring time of the PROFIBUS.

10 Codes

10.1 Overview

Code	Subcode	Index	Designation	See
C0126	-	$24449_d =$ $5F81_h$	Behaviour with regard to communication errors	📖 122
C1812	-	$22763_d =$ $58EB_h$	Software ID	📖 124
C1813	-	$22762_d =$ $58EA_h$	Software creation date	📖 124
C1882	-	$22693_d =$ $58A5_h$	Monitoring response for PCD communication fault	📖 122
C1883	-	$22692_d =$ $58A4_h$	Response time to exiting "Data_Exchange"	📖 123

Lenze

How to read the code table

Column	Meaning
Code	(Lenze) code
	• The parameters of a configurable code marked with an asterisk (\<Code>*) can only be accessed via the communication module.
	• The value of a configurable code marked with a double asterisk (\<Code>**) is not transmitted with the parameter set transfer.
Subcode	Subcode
Name	Designation of the Lenze code
Index	Index under which the parameter is addressed.
Lenze	Lenze setting of the code

Lenze		
	Disp	Display code Configuration of this code is not possible.

Values	Fixed values determined by Lenze (selection list) *or* a value range:		
	Minimum value	[Smallest increment/unit]	Maximum value

Column	Meaning
Access	R = read access (reading permitted) W = write access (writing permitted)
Data type	• FIX32: 32-bit value with sign; decimal with 4 decimal positions • U16: 2 bytes bit-coded • U32: 4 bytes bit-coded • VS: visible string, character string with defined length

10.2 Monitoring codes

C0126:
Behaviour with communication error (extract)

Code	Subcode	Index	Possible settings		Data type
			Lenze	Selection	
C0126	-	24449$_d$ = 5F81$_h$	See documentation for the controller.		FIX32

Monitoring the internal communication between communication module and controller.

C1882:
Process data
monitoring response

Code	Subcode	Index	Possible settings		Data type
			Lenze	Selection	
C1882	-	22693$_d$ = 58A5$_h$	0	0　　[1]　　3 Value = [selection] x 10000	U32

Selection	Response
0	No response
1	Fault (TRIP)
2	Controller inhibit (CINH))
3	Quick stop (QSP)

The Lenze code is used to determine the controller reaction when the process data monitoring time has expired.

This function can be used for:

► DRIVECOM control

► PROFIdrive control

► device control

Tip!

► Code C1882 performs the same function as the index 6004$_{hex}$.

► Observe the notes with regard to the interruption of the PROFIBUS communication (□ 118)

C1883:
Response time to exiting "Data_Exchange"

| | | | Possible settings | | |
Code	Subcode	Index	Lenze	Selection	Data type
C1883	-	22692$_d$ = 58A4$_h$	65535	0 [1 ms] 65535	FIX32

A value of 65535 deactivates monitoring.

A change in the monitoring time becomes effective immediately. Monitoring starts with the arrival of the first telegram.

 Tip!

Observe the notes with regard to interruptions of the PROFIBUS communication (118)

10.3 Diagnostics codes

C1812:
Display of software identification code

| Code | Subcode | Index | Possible settings | | Data type |
			Lenze	Selection	
C1812	1 ... 4	22763_d = $58EB_h$	[Disp]	-	U32

Display of software identification code (ID) in 4 subcodes with 4 characters each.

C1813:
Display of software creation date

| Code | Subcode | Index | Possible settings | | Data type |
			Lenze	Selection	
C1813	1 ... 4	22762_d = $58EA_h$	[Disp]	-	U32

Display of the software creation date in 4 subcodes with 4 characters each.

11 Index table

11.1 DRIVECOM profile parameter

I-6004$_{hex}$:
Process data monitoring selection code

The parameter determines the controller reaction when the process data monitoring time has expired.

Index [$_{hex}$]	Subindex	Possible settings		Data type
		Lenze	Selection	
I-6004	-	0	0: No reaction 1: Controller inhibit (CINH) 2: Quick stop (QSP) 4: TRIP	I8

12 Appendix

12.1 Parallel operation of AIF and FIF interfaces

> **ℹ️ Note!**
>
> The option of parallel operation ...
> - ► of a communication module (AIF) and a function module (FIF) exists for the standard devices 8200 vector and Drive PLC.
> - ► of two function modules (FIF) exists for the standard devices 8200 motec, Drive PLC and starttec.

Notes on parallel operation

Attach/detach the communication module to/from the AIF interface. This is also possible during operation.

8200vec073
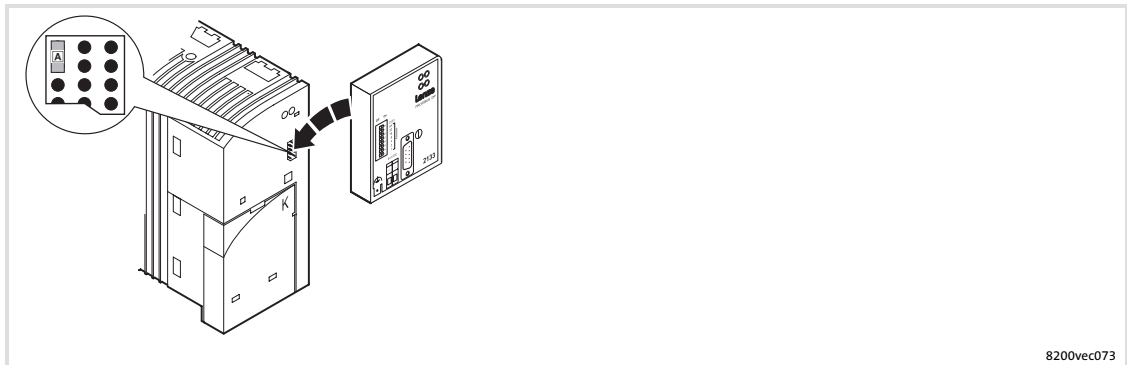
For internal voltage supply, you must set jumper Ⓐ at the indicated position.

External voltage supply (delivery state)	Voltage supply through internal voltage source

Lenze

Possible combinations

Function module on FIF (Design: Standard or PT)		Communication module on AIF	
		Keypad E82ZBC [1] Keypad XT EMZ9371BC [1]	PROFIBUS-DP 2131/2133
Standard I/O	E82ZAFS	✓✓	✓✓
Application I/O	E82ZAFA	✓✓	✓
INTERBUS	E82ZAFI	✓✓	⊠
PROFIBUS-DP	E82ZAFP	✓✓	⊠
LECOM-B (RS485)	E82ZAFL	✓✓	⊠
System bus (CAN) System bus I/O RS System bus I/O	E82ZAFC E82ZAFC100 E82ZAFC200	✓✓	✓✓
CANopen / DeviceNet [2]	E82ZAFD	✓✓	⊠
ASI	E82ZAFF	✓✓	⊠

[1] Is always supplied via internal voltage source regardless of the jumper setting.

[2] In preparation

✓✓ Combination possible, communication module is supplied internally or externally.

✓✓ Combination possible, communication module must be supplied externally

(✓) Combination possible, communication module can only be used for parameter setting (supplied internally or externally).

⊠ Combination not possible.

12.2 Accessories

In the following, you will find the accessory components for PROFIBUS:

> 🛈 **Note!**
>
> Please ask the manufacturer of the components for the latest order information and the technical data.

Designation	Note
Bus connector	Bus connector for 9-pole Sub-D plug with plug-in terminals for connecting the bus cable (order designation: Bus connector PROFIBUS RS485). ● Contains connectable bus terminating resistor. ● Order designation: Siemens, 6ES7 972-0BA00-0XA0
Bus cable	Features: ● Cable resistance: 135 ... 165 Ω/km (f = 3 - 20 MHz) ● Capacitance per unit length: $\leq$ 30 nF/km ● Loop resistance: < 110 Ω/km ● Core diameter: > 0.64 mm ● Core cross-section: 0.34 mm^2 ● Cores: double twisted, insulated and shielded Order designation: Siemens, SINEC L2 2-core bus cable

13 Index

Lenze

Lenze

Lenze

$C\,\epsilon$

10   9   8   7   6   5   4   3   2   1