7 2175 (CANopen) fieldbus module

7.1 Contents

7.2 General information

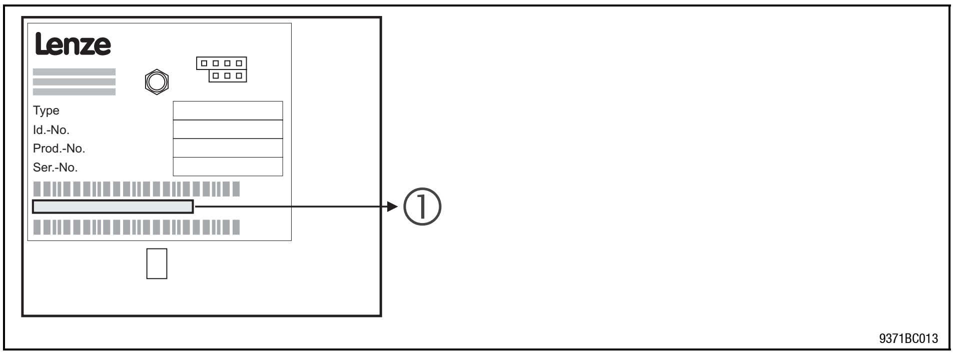
Validity of the Instructions

These Operating Instructions apply to the following fieldbus modules:

2175IB 1x. 1x.

These Operating Instructions are only valid together with the documentation of the corresponding basic devices.

Identification



Type code ① → **33.2175IB 1x 1x**

Series

Hardware version

Software version

Application range

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

Type	Design	Hardware version	Software version	Variant	Explanation
33. 820X	E.	2x.	1x.		(8201 - 8204)
33. 820X	E. / C.	2x.	1x.	Vxxx	(8201 - 8204)
33. 821X	E.	2x.	2x.		(8211 - 8218)
33. 821X	E. / C.	2x.	2x.	Vxxx	(8211 - 8218)
33. 822X	E.	1x.	1x.		(8221 - 8225)
33. 822X	E.	1x.	1x.	Vxxx	(8221 - 8227)
33. 824X	E.	1x.	1x.		(8241 - 8246)
33. 824X	E. / C.	1x.	1x.	Vxxx	(8241 - 8246)
82EVxxxxxBxxxXX		Vx	1x		(8200 vector)
82CVxxxxxBxxxXX		Vx	1x		(8200 vector, Cold plate)
82DVxxxKxBxxxXX		Vx	1x		(8200 vector, thermally separated)
EPL 10200	E	1x	1x		(Drive PLC)
33. 93XX	xE.	2x	1x	Vxxx	(9321 - 9332)
33. 93XX	xC.	2x	1x	Vxxx	(9321 - 9332, Cold plate)
33. 93XX	EI / ET	2x	1x	Vxxx	(9300 Servo PLC)
33. 93XX	CI / CT	2x	1x	Vxxx	(9300 Servo PLC, Cold plate)

Features

The internationally standardised CAN bus is mainly characterised by

- relatively short transfer times
- low expenditure for connection

These advantages have made CAN products interesting for other industries too.

In order to reach a standardisation, drive, control and sensor manufacturers have specified a communication profile with CANopen for solving control tasks.

The necessary parts of the protocol CiA DS301, version 4.01 have been implemented in the 2175 bus module.

- Attachable additional module for the basic Lenze devices 82XX, 8200 vector, 93XX, Servo PLC 9300 and Drive PLC.
- The front DIP switch enables easy setting of
  - – Communication profile DeviceNet or CANopen (DS301)
  - – Baud rate 10, 20, 50, 125, 250 , 500 and 1000 kbit/s (depending on the communication profile)
  - – Node address (max. 63 participants)
- Bus extension up to max. 5000m
- Topology: Line terminated at both ends (R = 120 Ohm)
- Easy connection because of pluggable screw terminals

7.3 Technical data

7.3.1 General data and application conditions

Field	Values			
Order name	EMF2175IB			
Communication medium	DIN ISO 11898			
Netzwork topology	Line (terminated at both ends with 120 Ohm)			
Number of nodes	Max. 63			
Cable length	Max. 7450 m (depending on the baud rate, 🕮 7.5-2)			
Communication profile	CANopen			
Ambient temperature	during operation:	-20 °C	to	60 °C
	Transport:	-25 °C	to	70 °C
	during storage	-25 °C	to	60 °C
Permissible humidity	Class 3K3 to EN 50178 (without condensation, average relative humidity 85%)			
Degree of pollution	VDE0110, part 2, pollution degree 2			
Voltage supply (internal/external), see 🕮 7.4-4	External supply via separate power supply			

7.3.2 Rated data

Insulation voltage between bus and ...	Rated insulation voltage	Type of insulation
• Remote earth / PE	50 V AC	Electrical isolation
• External supply (cl. 39/59)	-	No mains isolation
• Power supply unit		
– 820X / 821X	270 V AC	Basic insulation
– 822X / 8200 vector	270 V AC	Double insulation
– 93XX	270 V AC	Double insulation
• Control terminals		
– 820X / 8200 vector	-	No mains isolation
– 821X	50 V AC	Electrical isolation
– 822X	270 V AC	Basic insulation
– 93XX	270 V AC	Basic insulation
• External bus systems	0 V AC	No mains isolation

7.3.3 Communication times

Note!

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

The CAN bus communication times depend on

- Processing time in the controller
- Telegram time
  - – Baud rate
  - – Telegram length
- Data priority
- Bus load

More information about bus access control can be obtained from corresponding literature specialised on Controller Area Networks.

Processing times 820X

The processing times for the 8200 controllers differ from the times for the 821X/822X/8200 vector series.

In opposite to the 821X/822X/824X series, which have parallel process data processing, the 8200 series process process and parameter data sequentially. Therefore the time needed to respond process data depends on previous actions.

The processing time needed for telegrams also depends on the actual value conditioning (process data from controller). If these data (status word, actual frequency) are not required, they can be deactivated with the control word ”Bit 15” (PE inhibit).

The individual telegram times are:

Telegram	**Processing time**	
	PE-inhibit = 0	PE-inhibit = 1
Parameters	62...140 ms	62...70 ms
Change of a process data value to controller (*)	27...105 ms	27...35 ms
Change of both process data values to controller *	62...140 ms	4...70 ms
Process data from controller *	108...140 ms	not possible

Processing times 821X/822X/8200 vector

Parameters	**30 ... 50 ms**
Process data	**3 ... 5 ms** The processing times for the process data refer to the sync telegram 🕮 7.6-10

Processing time 93XX

The parameter data and process data are independent of each other.

Parameters	**approx. 30 ms + 20 ms tolerance (typical)** With some codes the processing time can be longer (see 9300 Manual).
Process data	**approx. 3 ms + 2 ms tolerance**

Telegram run time

The telegram run time depends on the baud rate and the telegram length:

Baud rate [kBit/s]	**Telegram length** [Byte]		
	0	**2**	**8**
10	5.44	7.36	13.12
20	2.72	3.68	6.56
50	1.09	1.47	2.62
125	0.44	0.59	1.05
250	0.22	0.29	0.52
500	0.11	0.15	0.26
1000	0.05	0.07	0.13

Tab. 7.3-1 Maximum telegram time in [ms]

The telegram times indicated in the table above are calculated according to the following equation. This equation allows to calculate any intermediate value t_{Tmax}.

$$t_T \leq \frac{54.4 + 9.6 \cdot L_D}{d_{Ü}}$$

t_T = telegram time [ms]
L_D = telegram length [byte]
$d_{Ü}$ = baud rate [kBit/s]

7.3.4 Dimensions

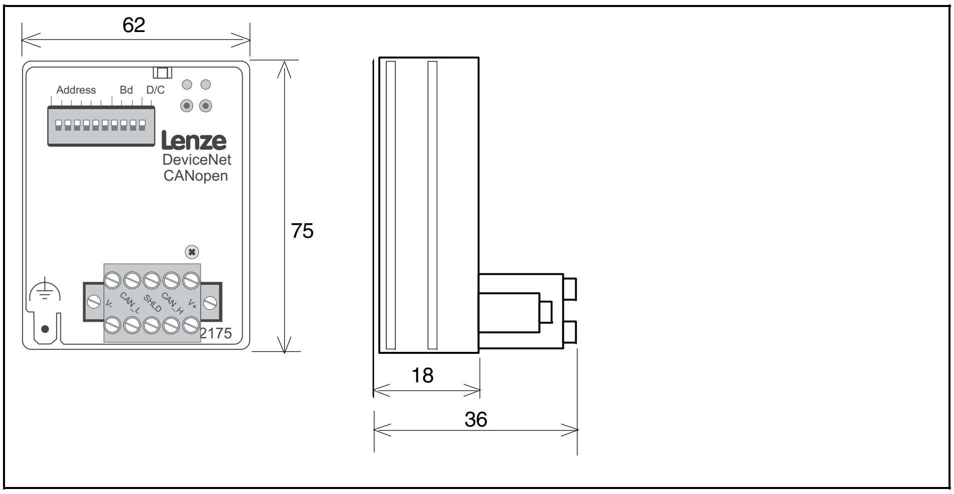


Fig. 7.3-1 Dimensions of the 2175 fieldbus module (all dimensions in mm)

7.4 Installation

7.4.1 Components of the communication module

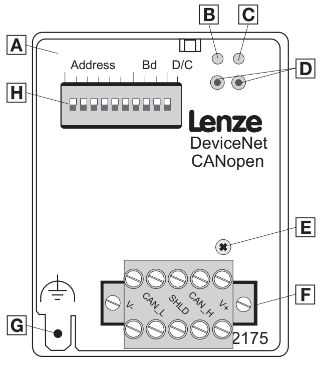


Fig. 7.4-1 2175 communication module

Pos.	Designation	Meaning		Notes
A	**2175 communication module**			
B	**Connection status to the basic device** (two-colour LED	OFF		Communication module is not supplied with voltage; basic device or external voltage supply is switched off.
		GREEN	BLINKING	Communication module is supplied with voltage but is not connected to the controller (controller is switched off, initialising, or not available).
			constantly ON	The communication module is supplied with voltage and is connected to the drive controller.
C	**Connection status to the bus** (two-colour LED)	OFF		• No communication with the communication module • Communication module is not supplied with voltage
		GREEN	BLINKING	Communication via the communication module has been set up
		RED	ON	Internal fault of the communication module
D	**Green and red Drive-LED (Drive)**	Operating status of the basic device 82XX, 8200 vector, 93XX, Servo PLC 9300 and Drive PLC (see Operating Instructions of the basic device)		

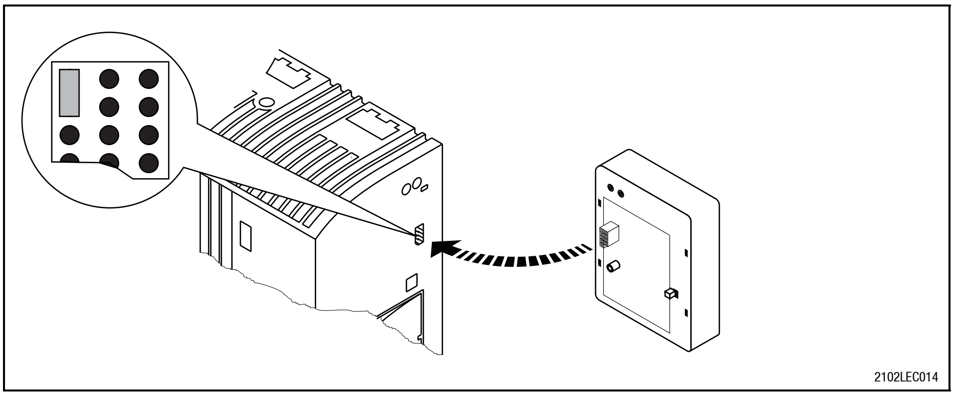
E	**Fixing screw**	
F	**Plug connector with double screw connection, 5-pole**	
G	**PE cable connection**	See note below
H	**DIP switch**	For settings see chapter 7.5

Note!

Only for 820X and 821X:

If necessary use an additional PE shield cable which avoids EMC-related communication interference in especially noisy environments.

7.4.2 Mechanical installation



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

Note!

An internal supply of the communication module through the 8200 vector is only possible if the jumper in the interface cutout (see figure above) is changed.
Please see the corresponding notes 📖 7.4-4 .

7.4.3 Electrical installation

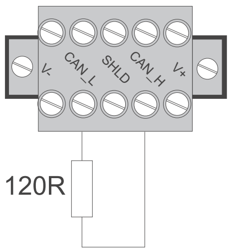
Note!

Only for 820X and 821X:

If necessary use an additional PE shield cable which avoids EMC-related communication interference in especially noisy environments.

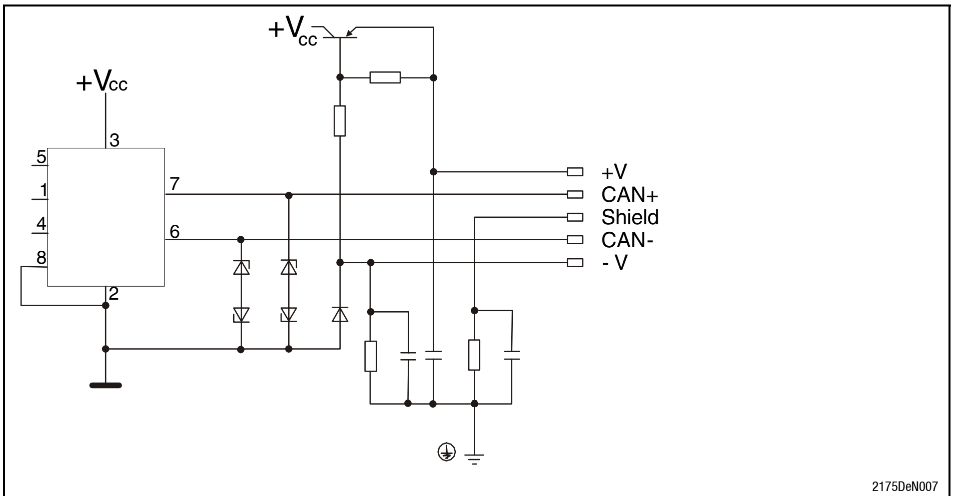
Assignment of the plug connector

The 2175 fieldbus module is connected to the bus through a 5 pole plug connector with double screw connection..



Name	Explanation
V-	GND; reference for external supply
CAN_L	Data cable / input for terminating resistance of 120 Ohm
SHIELD	Shield
CAN_H	Data cable / input for terminating resistance of 120 Ohm
V+	External voltage supply, see 7.4-4

Internal bus terminal assignment



External supply voltage

If necessary, supply the 2175 fieldbus module with a separate supply voltage 24 V DC via the plug-in contacts V+/V- ± 10 %.

820X controllers always require a separate voltage supply.

Use a separate supply unit in every control cabinet if the distance between the control cabinets is larger than normal.

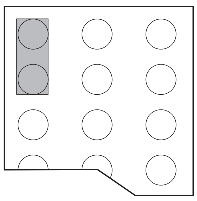
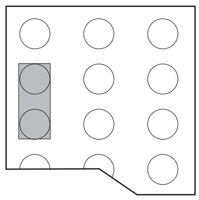
Controller	External voltage supply
820X	always required
821X / 822X / 824X and 93XX	Only necessary if the mains which supplies the corresponding controllers is to be switched off but the communication must not be interrupted.
8200 vector	see “internal DC voltage supply”

Internal DC voltage supply

Note!

Controllers with an extended AIF interface (front of the 8200 vector) can be internally supplied. The part of the drawing highlighted in grey shows the jumper position.

- In the delivery state of the frequency inverter these are not internally supplied.
- For internal voltage supply, put the jumper in the position indicated below.

Lenze setting only external voltage supply possible	**Internal voltage supply**
	

Wiring to a host

Danger!

An additional mains is required, if

- a 820X or 821X controller is connected to a host and
- a safe mains isolation (double basic insulation) to VDE 0160 is necessary.

For this, you can use an interface module for the host with an additional electrical isolation (see the corresponding manufacturer's information).
For wiring, the electrical isolation of the supply voltage must be taken into account. The supply voltage is assigned to the same potential as the data bus.

Wiring of the CAN bus

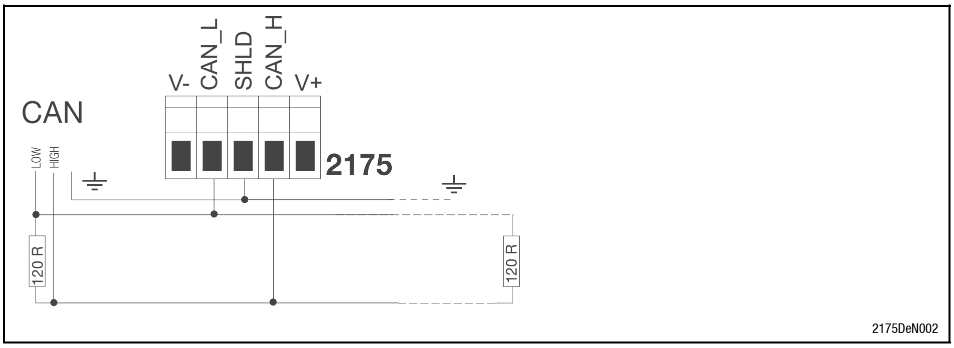


Fig. 7.4-2 Connection to the plug connector

Specification for system bus cable

Please observe our recommendations for signal cables:

Total length	**≤ 300 m**	**≤ 1000 m**
Cable type	LIYCY 2 x 2 x 0.5 mm^2 (twisted in pairs with shield)	CYPIMF 2 x 2 x 0.5 mm^2 (twisted in pairs with shield)
Cable resistance	≤ 40 Ω/km	≤ 40 Ω/km
Capacitance per unit length	≤ 130 nF/km	≤ 60 nF/km

Structure of a CAN bus system (example)

The CAN bus system is designed as 2 conductor (twisted pair) shielded with additional mass and termination at both ends of the line.

For sending and receiving data the following paths are available:

- Max. 3 process data channels (PDO = Process Data Object)
  – Process data are send via the process data channel and are used for high-speed and high-priority control tasks. Typical process data are, for instance, setpoints and actual values of a basic device.
- 2 parameter data channels (SDO = Service Data Object)
  – The parameters are transferred at lower priority than the process data. The parameters are set or changed e.g. during commissioning or product change.
  – The parameters are accessed via the parameter data channel of the 2175 fieldbus module to the codes of the basic device by Lenze or the corresponding CANopen objects (description see chapter ”Parameter setting CANopen”).
  – 2 masters can be connected to the basic devices because of the 2 parameter channels. Thus, parameters can be changed directly at the basic device during operation of a machine or system networked via PLC, using a PC (e.g. with user software “Global Drive Control) or a keypad. The second parameter data channel can be reached under the set address (switch or L-C0009) with an offset of 64. For instance, if a PLC addresses the controller with address 1 and second commanding unit the address 65, the same basic device will be addressed. Please observe that the last telegram determines the parameter when a parameter is accessed by two units (see “Server SDO Parameters” (🕮 7.7-25)).

Note!

The last telegram determines the parameter when a parameter is accessed by two units.

Please observe the notes in chapter 7.5, if you do not select the baud rate and address via the front switch.

7.4.4 Bus cable length

It is absolutely necessary to comply with the permissible cable lenghts.

1. Please check the compliance with the total cable length in Tab. 7.4-1.

The total cable length is specified by the baud rate.

Baud rate [kBit/s]	10	20	50	125	250	500	1000
Total cable length [m]	7450	3950	1550	630	290	120	25

Tab. 7.4-1 Total cable length

2. Please check the compliance with the segment cable length in Tab. 7.4-2.

The segment cable length is specified by the cable cross-section used and the number of participants. Without a repeater the segment cable length corresponds to the total cable length.

	Cable cross-section			
Station	**0.25 mm^2**	**0.5 mm^2**	**0.75 mm^2**	**1.0 mm^2**
2	240 m	430 m	650 m	940 m
5	230 m	420 m	640 m	920 m
10	230 m	410 m	620 m	900 m
20	210 m	390 m	580 m	850 m
32	200 m	360 m	550 m	800 m
63	170 m	310 m	470 m	690 m

Tab. 7.4-2 Segment cable length

3. Please compare both detected values.

If the value detected from Tab. 7.4-2 is smaller than the total cable length from Tab. 7.4-1, repeaters must be used. Repeaters divide the total cable length into segments.

Note!

- Please note the reduction of the total cable length due to the signal delay of the repeater (see example 📖 7.4-8).
- Mixed operation
  – There is a mixed operation, if different devices are connected to the same mains.
  – If the total cable lengths of the participants are different at the same baud rate, the smaller value must be used in order to determine the max. cable length.

Example: Selection help

Given:	
• Cable cross-section:	0.5 mm^2 (according to cable specification 📖 7.4-5)
• Number of devices connected:	63
• Repeater:	Lenze repeater, type 2176 (cable reduction: 30 m)

At maximum number of participants (63) the following cable lengths / number of repeaters must be complied with:

Baud rate [kBit/s]	**10**	**20**	**50**	**125**	**250**	**500**	**1000**
Max. cable length [m]	7450	3950	1550	630	290	120	25
Segment cable length [m]	310	310	310	310	290	120	25
Number of repeaters	24	12	4	2	-	-	-

Example: Check repeater application

Given:	
• Baud rate:	125 kBit/s
• Cable cross-section:	0.5 mm^2
• Number of devices connected:	28
• Cable length:	450 m
1. Total cable length at 125 kbits/s	
630 m	from Tab. 7.4-1
2. Segment cable length for 28 participants and a cable cross-section of 0.5mm^2 .	
360 m	from Tab. 7.4-2
3. Comparison	
The value in point 2. is smaller than the cable length of 450 m.	
4. Conclusion	
• It is not possible to use a cable length of 450 m without applying a repeater. • After 360 m (point 2.) a repeater must be applied.	
5. Max. cable length with repeater application	
• The Lenze repeater is used, type 2176 (cable reduction: 30 m) • Calculation of the max. cable length: 630 m (according to Tab. 7.4-1) minus 30 m (cable reduction) → Max. possible cable length with repeater: 600 m. → The cable length wanted is now possible.	

Note!

Repeaters are recommended as a
- service interface
  Advantage: trouble-free connection during bus operation is possible.
- Calibration interface
  Advantage: calibration/programming unit remains electrically isolated.

7.5 Commissioning

Note!

Settings via GDC or operating module.

The settings of controller address and baud rate can be carried out by means of GDC or the operating module. For this purpose the DIP switches S1 to S6 must always be set to the OFF position.

The codes L-C0009 (controller address) and L-C0125 (baud rate) are not active, if one address switch has been set to ON before restarting the controller.

Concerning this please observe the descriptions (📖 7.9-1) of

- Node adress L-C1850/2350 and
- Baud rate L-C1851/2351.

The following settings can be easily carried out via the front DIP switch of the 2175 field bus module:

- Controller address S1 - S6
- Baud rate S7 - S9
- Communication profile CANopen S10

Note!

The Lenze setting for all switches is OFF.

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

Only the switch combinations described in the following tables are defined states.

Communication profile setting

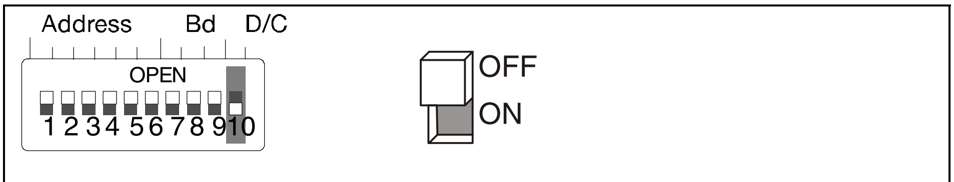


Fig. 7.5-1 Communication profile setting

Communication profile	S10
DeviceNet	OFF
CANopen	ON

Device address setting

> **Note!**
> Please observe that the addresses are not the same when using several controllers.

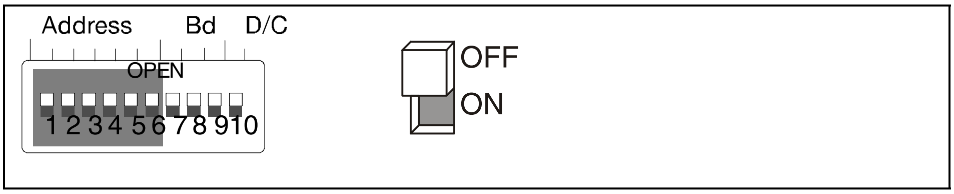


Fig. 7.5-2 Setting of the controller address

$$Adresse_{dec} = S_6 \cdot 2^0 + S_5 \cdot 2^1 + S_4 \cdot 2^2 + S_3 \cdot 2^3 + S_2 \cdot 2^4 + S_1 \cdot 2^5$$

The address calculation (decimal number) is based on the positions of switches S1 ... S6 ('0' = OFF and '1' = ON). The numbers must be entered into the equation above.

The equation also indicates the valency of a switch. The sum of valencies results in the controller addresses to be set

Switch valencies:

Switch	S1	S2	S3	S4	S5	S6
Valency	32	16	8	4	2	1

Example:

Switch	S1	S2	S3	S4	S5	S6
Switch position	ON	ON	ON	OFF	OFF	OFF
Address (= 56)	32	16	8	0	0	0

Baud rate setting

> **Note!**
> Please observe that the baud rate must be the same for all controllers and the host.

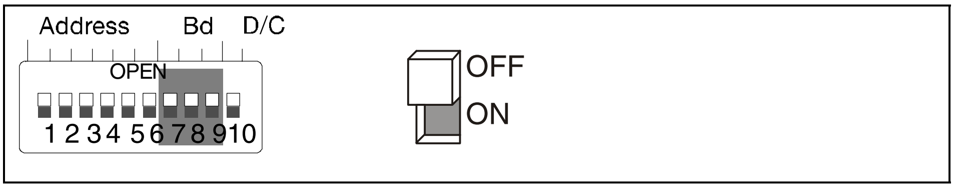


Fig. 7.5-3 Baud rate setting

Baud rate [kBit/s]	**S7**	**S8**	**S9**
10	ON	ON	OFF
20	ON	OFF	ON
50	OFF	ON	ON
125	OFF	ON	OFF
250	OFF	OFF	ON
500	OFF	OFF	OFF
1000	ON	OFF	OFF

7.5.1 Before switching on

Stop!

Before switching on the mains voltage, check the wiring for completeness, earth fault and short circuit.

Note!

Do not change the switch-on sequence!

1. Switch on the controller and, if necessary, the external supply of the 2175 fieldbus module.
   – The operating status display of the controller (🕮 7.4-1) must be on or blinking. If this is not the case, see chapter "Troubleshooting" (🕮 7.8-1)
   – The green LED ("status controller connection") must also be on (🕮 7.4-1). If this is not the case, see chapter "Troubleshooting" (🕮 7.8-1).
2. It is now possible to communicate with the drive, i.e.
   – all parameters (SDO) can be read and written
   – all writable parameters (SDO), except for process data (PDO) such as frequency setpoint or control word, can be overwritten.
   – For more information on the communication phases of the CAN network see (🕮 6.6-3).

7.5.2 Enable drive via 2175 fieldbus module

Note!

During operation the change of a 2175 module to another controller can lead to undefined operating states.

82XX / 8200 vector	1. Set the Lenze parameter Operating Mode (L-C0001) from 0 to 3 to enable the drive via the 2175 fieldbus module. This can be carried out with the keypad or directly via CANopen. Examples for Write (L-C0001=3): – Index = $5FFE_{hex}$ (results from: $5FFF_{hex}$ – $(L\text{-}C0001)_{hex}$;) – Subindex: 0 – Value: 30000_{dec} (results from: L-C0001 = 3 x 10000)
	2. Terminal 28 (controller enable) is always active and must be set to HIGH level during CANopen operation (see Operating Instructions for the controller). Otherwise, the controller cannot be enabled via CANopen. – With 821X, 8200vector and 822X, the function QSP (quick stop) is always active. If QSP is assigned to an input terminal (default setting: not assigned), this must be at HIGH level during CANopen operation (see Operating Instructions for the controller). The controller now accepts parameter and process data.
93XX	1. For drive control via CANopen set the Lenze parameter Signal Configuration (L-C0005) to a value xxx3. This change can be carried out using the 9371BB keypad or the CANopen. For the first commissioning, select the signal configuration 1013. Examples for Write (L-C0005=1013): – Index = $5FFA_{hex}$ (results from: $5FFF_{hex}$ – $(L\text{-}C0005)_{hex}$) – Subindex: 0 – Value: 10130000_{dec} (results from: L-C0005 = 1013 x 10000) 2. Set the parameter L-C0142 to 0. Observe the chapter “Protection against unexpected start“. 3. Terminal 28 (controller enable) is always active and must be set to HIGH level during CANopen operation (see Operating Instructions 93XX). Otherwise, the controller cannot be enabled via CANopen. – With the signal configuration L-C0005=1013, the function QSP (quick stop) and the CW/CCW changeover are assigned to the digital input terminals E1 and E2 and thus they are always active. For CANopen operation E1 must be set to HIGH level (see Operating Instructions 93XX). **Note** With the signal configuration L-C0005=xx13, terminal A1 is switched as voltage output. Connect the following terminals: – X5.A1 with X5.28 (controller enable). – X5.A1 with X5.E1 (CW/QSP)
	The controller now accepts parameter and process data.

Protection against uncontrolled restart

Note!

After a fault (e.g. short-term mains failure) a restart of the drive is not always wanted.

- By setting L-C0142 = 0, the drive can be inhibited if
  – the corresponding controller sets a ”fault message“
  – the fault is active for more than 0.5 s

Parameter function:

- L-C0142 = 0
  – Controller remains inhibited (even if the fault is not active any longer)
  – The drive restarts in a controlled mode: LOW-HIGH transition at one of the inputs for ”Controller inhibit“ (CINH, e.g. at terminal X5/28)
- L-C0142 = 1
  – Uncontrolled restart of the controller possible

7.6 Data transfer

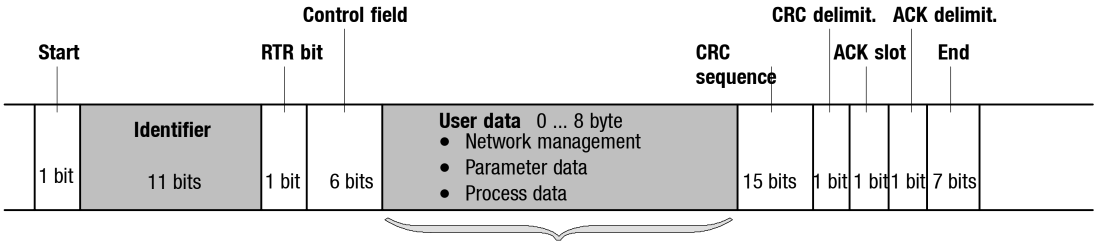
Master and slave communicate with each other by exchanging data telegrams via the CAN bus. The user data range of the data telegram contains either network management data, parameter data or process data.

In the controller, different communication channels are assigned to the parameter data and process data:

Telegram type		Communication channel	
Parameter data (SDO, Service-Data-Objects)	These are, for instance, • Operationg parameters • Diagnostics information • Motor data In general, the parameter transfer is not as time-critical as the tranfer of process data.	**Parameter data channel**	• Enables the access to all Lenze codes and the CANopen index. • Parameter changes are usually stored automatically in the controller (observe L-C0003).
Process data (PDO, Process-Data-Objects)	These are, for instance, • Setpoints • Actual values Exchange between host and controller required as fast as possible. Small amounts of data which can be transferred cyclically.	**Process data channel**	• You can control the controller using the process data. • The host has direct access to the process data. Data are for instance directly assigned to the I/O area. • Process data are – not stored in the controller. – transferred between host and controller in order to provide a continuous exchange of current input and output data.

Tab. 7.6-1 Division of parameter data and process data into different communication channels

7.6.1 Structure of a CAN data telegram



Three different data types are transported in this field:

- **Network telegrams (NMT)**
  – The NMT user data contain information about the structure of communication via CAN bus.
- **Parameter data (SDO)**
  – The user data serve to parameterize the devices.
- **Process data (PDO)**
  – The process data in the user data are used for fast, mainly frequent processes (e.g. speed setpoint and actual speed value).

Fig. 7.6-1 Basic structure of the CAN data telegram

The chapter mentioned goes into more detail about the data relevant for programming the bus module (identifier and user data).

The other signals refer to the transfer characteristics of the CAN telegram. These Instructions do not describe this matter. For further information please refer to the homepage “CAN in Automation (CiA)”: www.can-cia.org.

2175 (CANopen) fieldbus module

Data transfer

Structure of a CAN data telegram

Identifier

The identifier is an important part of the data telegram. Every identifier - except the network manager and the sync telegram (see chapter 6.6.3), contains the controller address:

Identifier = Basic identifier + unit address

In case of the CANopen communication profile the controller address is used to realise a participant-oriented message addressing.

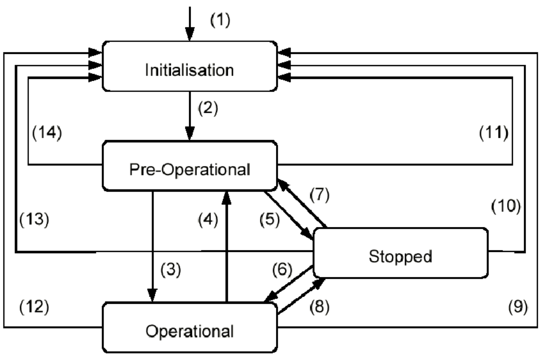
The identifier allocation is is defined in the CANopen protocol. According to Lenze settings, the basic identifier is preset with the following values (see CiA DS301, Pre-Defined Connection Set):

	Direction		**Basic identifier**		**+ Controller address**	**see**
	from the controller	to the controller	dec	hex		
Network manager (NMT)			0	0	no	
Sync telegram			128	80		📖 7.6-10 📖 7.7-21
Emergency object	X		128	80	yes	📖 7.7-24
Process data channel 1	X		384	180	yes	📖 7.7-26 ff.
		X	512	200		
Process data channel 2	X		640	280		
		X	768	300		
Process data channel 3	X		896	380		
		X	1024	400		
Parameter data channel 1	X		1408	580		📖 7.6-26, 📖 7.7-25
		X	1536	600		
Parameter data channel 2	X		1472	5C0		
		X	1600	640		
Node guarding	X		1792	700	yes	📖 7.6-5, 📖 7.7-21

Network manager (NMT)

The telegram structure used for the network manager contains the identifier and the command being located in the user data and consisting of the command byte and the device address:

User data (2 bytes)	
1. byte: command (hex)	2. byte: controller address
01, 02, 80, 81 or 82	**Controller address: xx** The following applies to the assignment of the bytes marked with "xx" in the table below: • **xx = 00_{hex}** With this assignment, all controllers connected are addressed by the telegram. All controller can change their status at the same time. • **xx = Controller address** If a certain address is indicated, the status will only be changed for the controller addressed.



State transition	**Command (hex)**	**Network status after change**	**Effect on process and parameter data after state transition**
(1)	-	Initialisation	At power-on the initialisation automatically started. The drive does not take part in the data transfer.
(2)	-	Pre-Operational	After initialisation • the participant automatically passes over to the Pre-Operational state. • the master decides how the controller/s are to participate in the communication. The master changes a status for the whole network. A target address, which is part of the command, selects the slaves.
(3), (6)	01 xx	Operational	Network manager telegrams, sync, emergency, process data (PDO) and parameter data (SDO) active (corresponds to "Start Remote Node")
(4), (7)	80 xx	Pre-Operational	Network manager telegrams, sync, emergency and parameter data (SDO) active (corresponds to "Enter Pre-Operational State")
(5), (8)	02 xx	Stopped	Network manager telegrams can only be received.
(9)	81 xx	Initialisation	Initialisation of all parametersin the field bus module with the stored values (corresponds to "Reset-Node")
(10)			
(11)			
(12)	82 xx		Initialisation of communication-relevant parameters (CIA DS 301) in the field bus module with the stored values (corresponds to "Reset Communication")
(13)			
(14)			

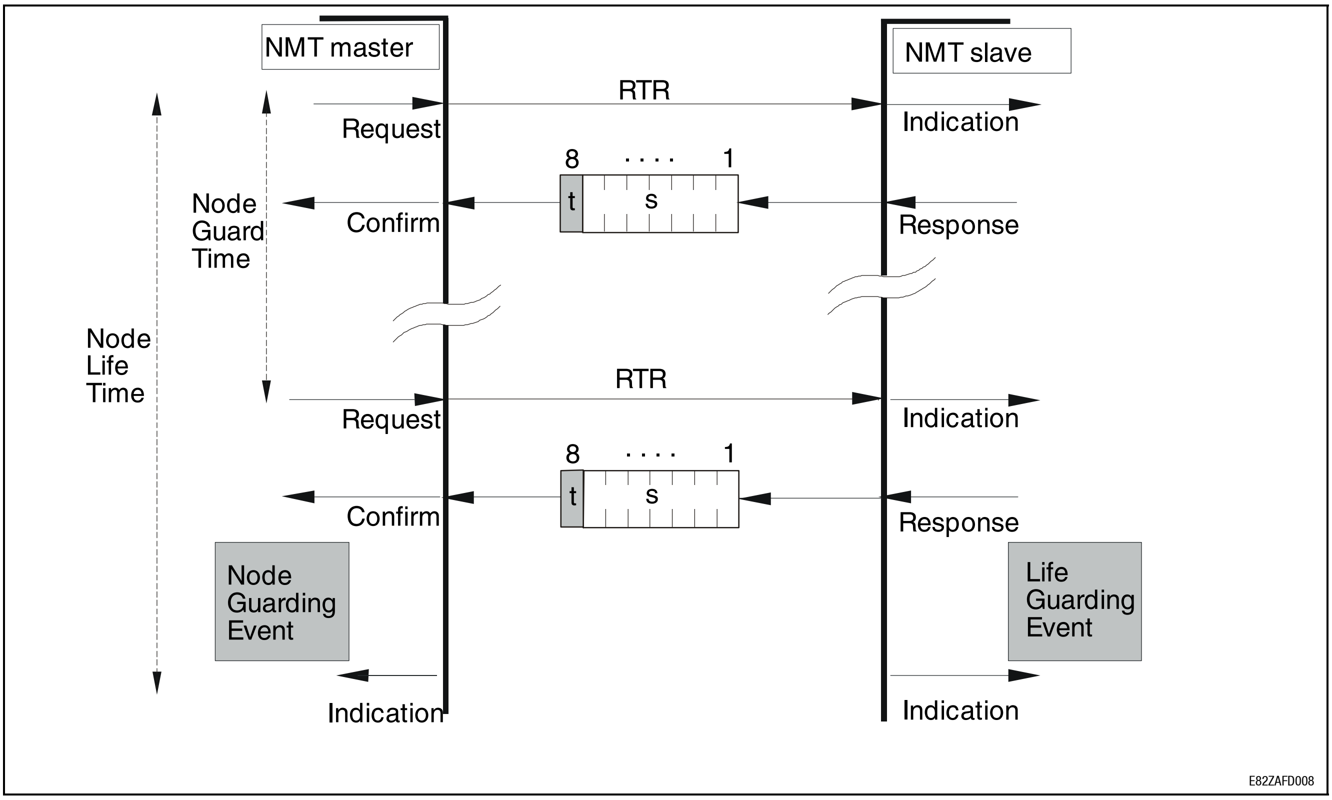
Node Guarding Protocol

The Node Guarding Protocol serves to monitor the connection between the NMT master and the NMT slave(s) within a CAN network.

Note!

A NMT master is e.g. a PLC with a CAN interface or a PC card.

The NMT slave function of the Node Guarding Protocols (DS301, version 4.01) is supported by the field bus module.



RTR telegram

The NMT master sends a data telegram called "Remote Transmit Request" (RTR) to the NMT slave in cyclic time intervals ("Node Guard Time", monitoring time).

– For this purpose, the RTR bit is set to the significance LOW (dominant level) in the arbitration field.
– The RTR does not contain any user data.

The NMT slave is asked by the RTR to send its actual data.

Response telegram

Every NMT slave on its part sends a response telegram with a user data width of one byte. Its most significant bit is a toggle bit (t). The toggle must change its significance at each response. The significance of the toggle bit at the initial activation of the Node Guarding Protocol is "0".

The toggle bit is reset to the value 0 by the "Reset_communication (NMT) telegram" of the NMT master.

The data value (s) of the other seven bits indicates one of the three possible states of the NMT slave:

Value s	Status
4	STOPPED
5	OPERATIONAL
127	PRE-OPERATIONAL

Identifier

The request of the NMT master and the corresponding response of the NMT slave are sent with an identifier (1792_{dec} + slave address, (📖 7.6-3)).

The identifier (see also (📖 7.6-3)) is calculated as follows:

Basic address (1792_{dec}) + *slave address* ($1....63_{dec}$)

Node Life Time

For each NMT slave a different "Node Life Time" can be set.

The Node Life Time is the product of the "Node Guard Time" (📖 7.7-21) and the "Life Time Factor" (📖 7.7-22).

The NMT master must recognise these two values. This can be done by reading the values at every restart out of the NMT slave by the NMT master.

OK state

The connection state is OK, if within the "Node Life Time"

- the NMT master has received a correct response of the NMT slave or
- the NMT slave has received a request from the NMT master.

In this case

- the monitoring times for the NMT master and NMT slave have been reset
- the Node Guarding Protocol will be continued.

Life Guarding Event

Through the "Life Guarding Event" a fault is released in the NMT slave, if the NMT slave is not triggered by an RTR or NMT master within the "Node Life Time".

> **Note!**
>
> The reaction to a "Life Guarding Event" is set with the code L-C1882 / L-C2382.

Node Guarding Event

The "Node Guarding Event" is to appear in the NMT master, if

- within the "Node Life Time" the NMT master does not receive any response of the NMT slave although a request has been made,
- the toggle bit has not changed within the "Node Life Time".

> **Note!**
>
> Please also note in this connection that the monitoring times are not to be reset. The reaction to a "Node Guarding Event" in the NMT master shall be implemented accordingly, if the significance of the toggle bit equals the NMT slave telegram received before.

7.6.2 Process data channel

Setpoint source selection

82XX controller

The selection of the setpoint source for these controllers is determined under code number L-C0001 (index: $5FFE_{hex}$). For process data evaluation, the code L-0001 must be set to the value "3" when operating the controller with the fieldbus module. The setpoint source is the process data channel which overwrites the frequency setpoint (L-C0046) and the control word (L-C0135) (see 82XX Operating Instructions).

Note!

Please observe that the setpoint source selection (LC0001) must be set equally in all parameter sets.

Controller 8200 vector

The selection of the setpoint source for these controllers is determined under code number L-C0001 (index: $5FFE_{hex}$). For process data evaluation, the code L-0001 must be set to the value "3" when operating the controller with the fieldbus module. (Selection: process data channel of a field bus module AIF-IN.W1 or AIF-IN.W2). The setpoint source is the process data channel which overwrites the frequency setpoint (L-C0046) and the control word (L-C0135) (see the Operating Instructions for 8200 vector).

Note!

Please observe that the setpoint source selection (LC0001) must be set equally in all parameter sets.

93XX controllers

The 93XX controller does not offer a setpoint source selection which can be set by one code only. For this purpose, you only need to connect preconfigured function blocks in order to adapt the controller to the drive task without being an expert in programming.

The user itself is able to carry out the interconnection. However, it is recommended to use the preconfigurations provided by Lenze, which are saved in the read-only memory of the controller. The Lenze preconfigurations (see code C0005) define, which source (terminal, keyboard, field bus module) describe the frequency setpoint and the control word.

The value to be set of code C0005 must be set to "xxx3" for operation via CAN bus (x = wildcard for selected preconfiguration).

For more information, please see the corresponding Manual or Operating Instructions for the controller.

Basic devices 9300 Servo PLC and Drive PLC

For communicating via an AIF fieldbus module (e.g. 2175 CANopen/DeviceNet) it is necessary that the system modules AIF-IN 1 ... 3 or AIF-OUT 1 ... 3 and if required the AIF manager are integrated into the control configuration of the IEC1131 project.

7.6.3 Process data transfer

Process data telegrams between host and controllers are distinguished as follows:

- Process data telegrams **to** the drive
- Process data telegrams **from** the drive

Process data telegram to drive

The process data telegram (device series 93XX) has a useful data length of 8 bytes (see example below).
The process data telegram identifier also includes the drive's address.

The CAN bus is connected to the automation interface X1.

X1 is connected to the function block AIF-IN. Here, the useful data is transformed into corresponding signal types in order to use them for further function blocks. The control word is especially important for the drive. It contains the drive setpoint in byte 1 and byte 2 of the useful data.

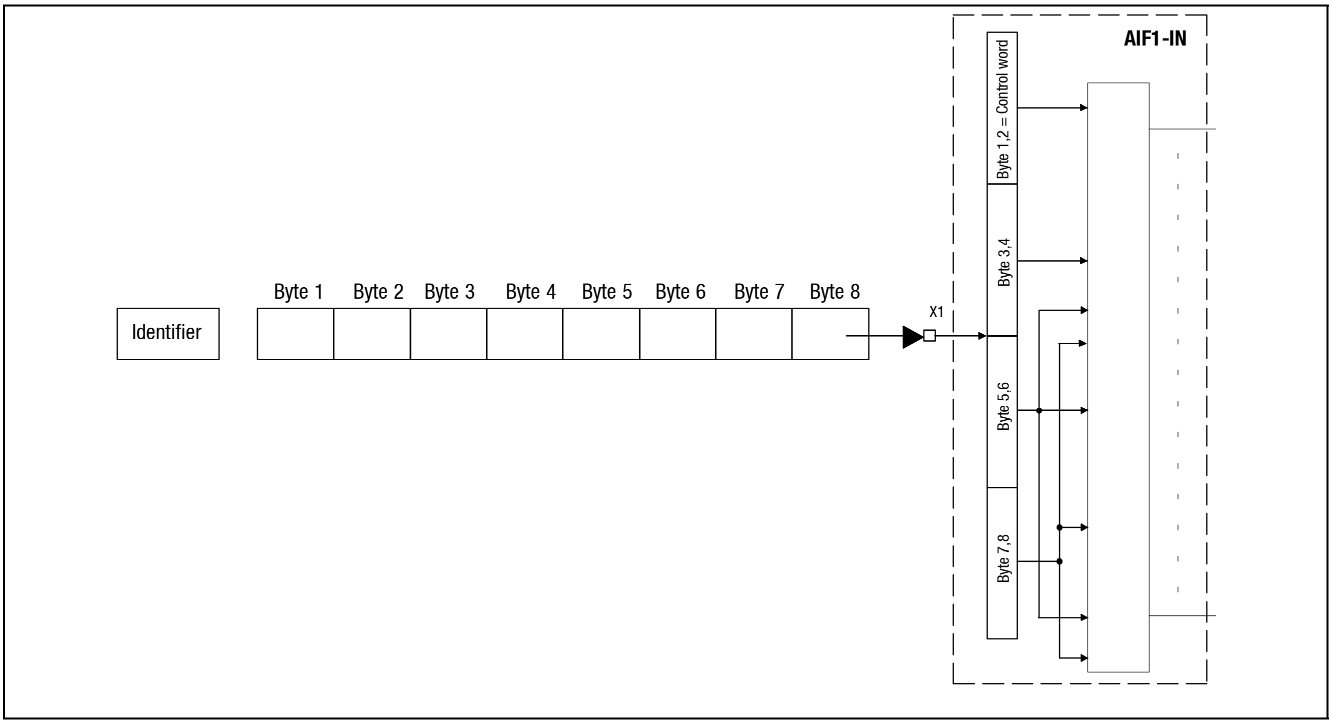


Fig. 7.6-2 Example: Device series 93XX

Process data telegram from drive

For the cyclic process telegrams from drive, the function block to be used is called AIF-OUT. The status word (byte 1 and byte 2) included in the process data telegram is placed on the CAN-BUS via this function block and is sent to, e.g. the master (see also chapter 7.6.6).

The sync telegram ensures that process data is sent to the drive (see also (🕮 7.6-3)).

For cyclic process data processing, the sync telegram must be generated accordingly.

Synchronisation of process data

The sync telegram is the trigger point for

- data sent to the drive controller
- starting sending process from the drive controller.

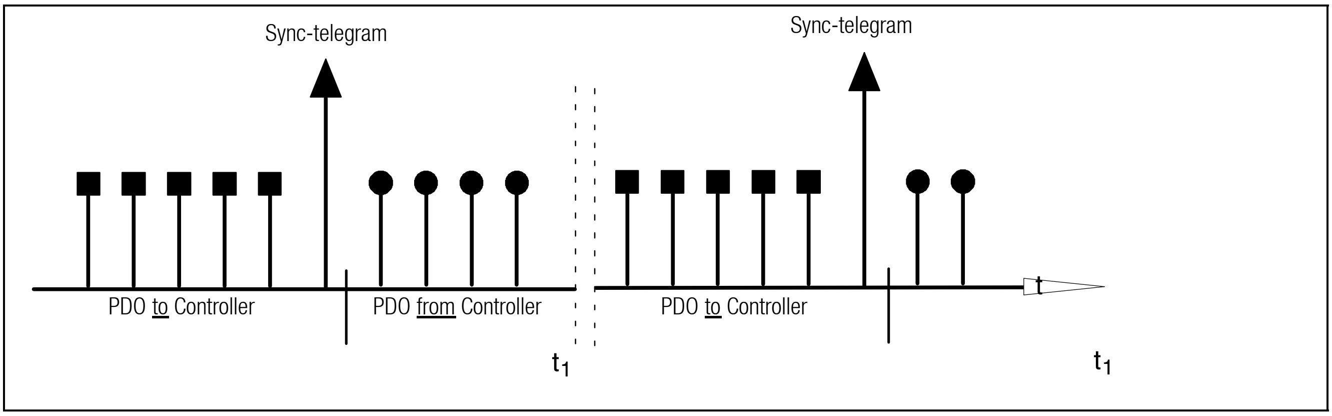


Fig. 7.6-3 Synchronisation of cyclic PDOs (represented by a bus participant)

Explanation for Fig. 7.6-3: At the time t_1, the process data are accepted for all PDOs as soon as a sync telegram will be received.

Note!

SDOs or event-controlled PDOs are accepted asynchronously, i.e. after transfer has been carried out.

The asynchronous data are not considered above!

7.6.4 Process-data assignments for 82XX

Process data telegram to drive

User data length: 8 Byte

Identifier	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	Control word L-C0135 Low byte	Control word L-C0135 High byte	Setpoint L-C0046 Low byte	Setpoint L-C0046 High byte	xx	xx	xx	xx

More information about the process data Telegram	Byte 1	The bits 0 to 7 of the control word under C0135 are entered here
	Byte 2	Bits 8 to 15 of the control word under C0135 are entered here (see (🕮 7.6-15) The description of the bits can be obtained from the Code Table.
	Byte 3	The frequency setpoint, which can also be written as parameter under C046, is entered here as process data word.
	Byte 4	The normalization differs from the setting under C046. It is a signed value, 24000 = 480 Hz.
	Byte 5	No evaluation of these data, any content is possible
	Byte 6	
	Byte 7	
	Byte 8	

Control word: see (🕮 7.6-15)

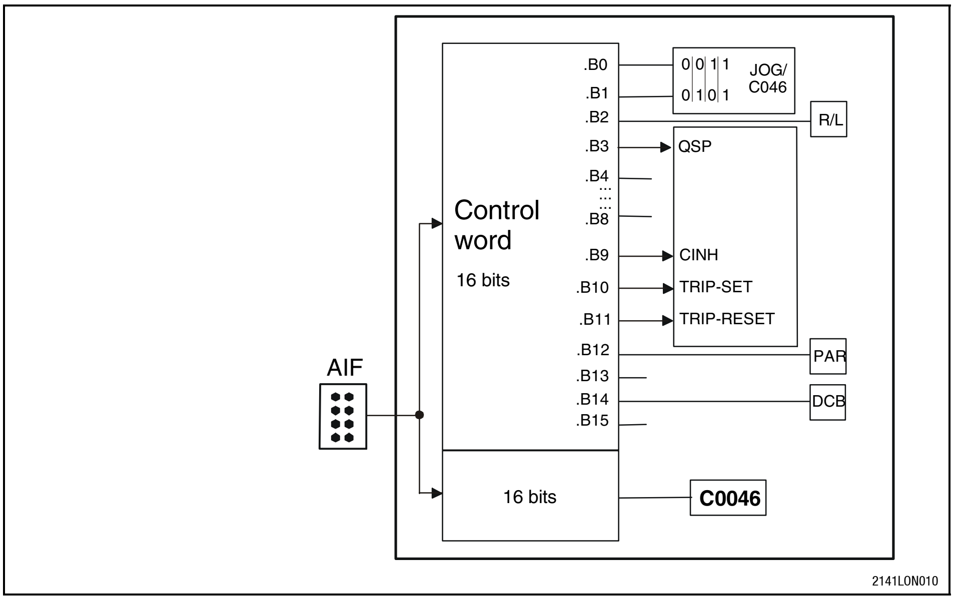


Fig. 7.6-4 Access to the control word and the frequency setpoint in 82XX (fixed assignment, see (🕮 7.6-15))

Process data telegram <u>from</u> drive

User data length: 8 Byte

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Identifier	Status word L-C0150 Low byte	Status word L-C0150 High byte	Actual value L-C0050 Low byte	Actual value L-C0050 High byte	xx	xx	xx	xx

More information about the process data telegram:		
	Byte 1	The bits 0 to 7 of the status word under L-C0150 are entered here
	Byte 2	Bits 8 to 15 of the status word under L-C0150 are entered here (see (🕮 7.6-16)). The description of the bits can be obtained from the Code Table.
	Byte 3	The actual frequency value with signed normalization (L-C0050) 24000 = 480 Hz is provided here.
	Byte 4	
	Byte 5	No evaluation of these data, any content is possible
	Byte 6	
	Byte 7	
	Byte 8	

Status word: see (🕮 7.6-16)

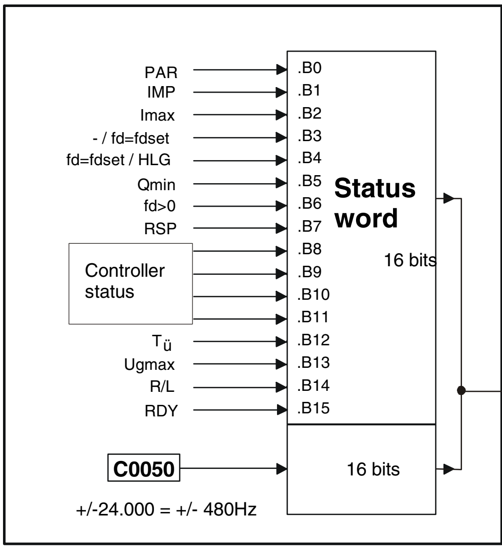


Fig. 7.6-5 Read access to the status word and the actual frequency value in 82XX (fixed assignment, see (🕮 7.6-16))

7.6.5 Process data assignment for 8200 vector

A change of code L-C0001 to 3 preconfigures the process data words in the controller.

Note!

Frequency and speed values are normalised with ±24000 ≡ ±480 Hz.

Process data telegram to drive

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

Control word: see (7.6-15)

AIF-IN.Wx is parameterised under code L-C0412.

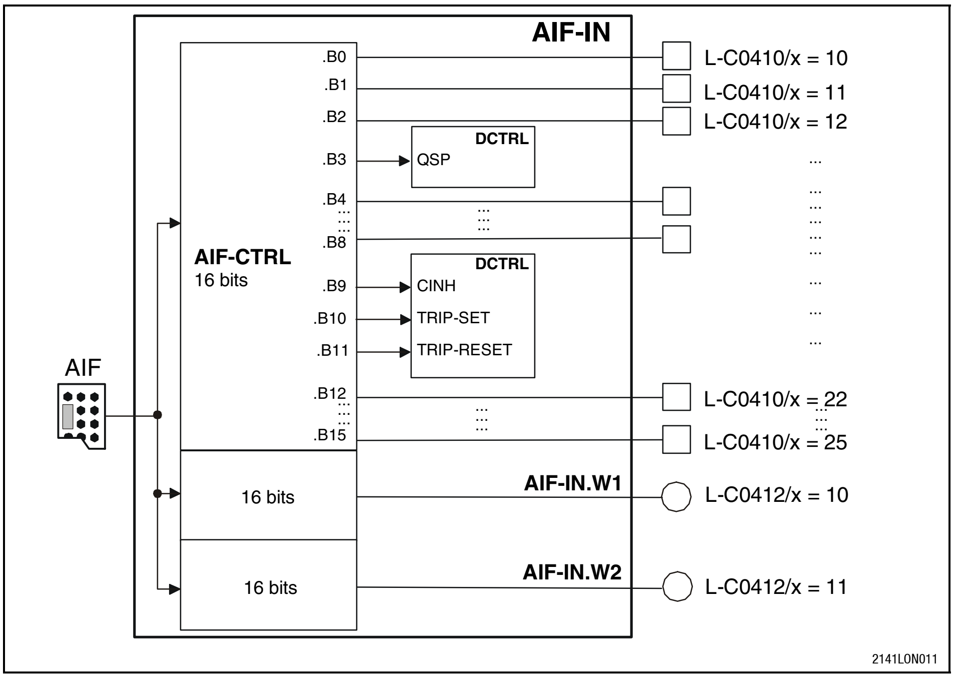


Fig. 7.6-6 Function block AIF-IN in 8200 vector (freely programmable assignment, factory setting see (7.6-15))
Note:
The subcode (wildcard "x" in illustration) determines the meaning of the bit or the word (see Operating Instructions for 8200 vector)

Process data telegram from drive

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Status word Low byte	Status word High byte	AIF-OUT.W1 Low byte	AIF-OUT.W1 High byte	AIF-OUT.W2 Low byte	AIF-OUT.W2 High byte	xx	xx

Status word: see (📖 7.6-16)

AIF-OUT.Wx is parameterized under code L-C0421.

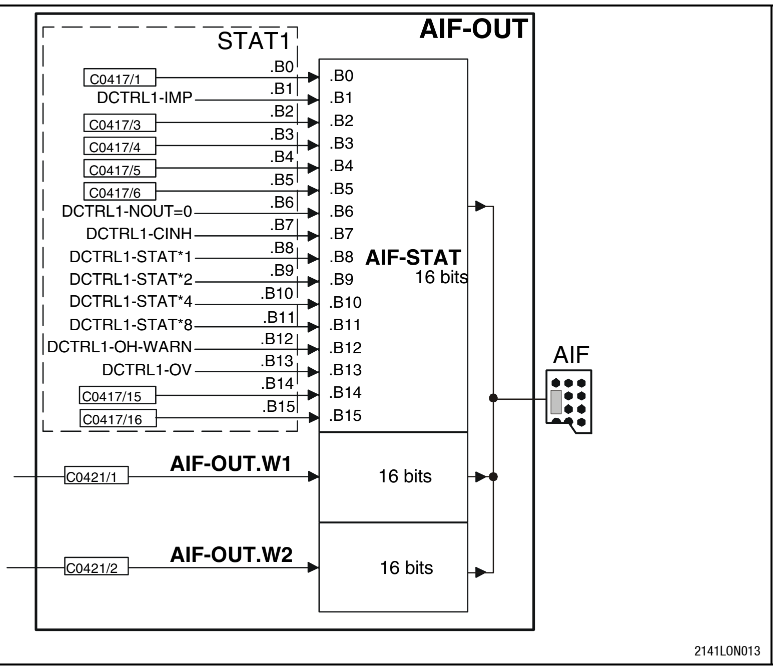


Fig. 7.6-7 Function block AIF-OUT in 8200 vector (freely programmable assignment, factory setting see (📖 7.6-16))

Control word for 82XX and 8200 vector

Bit	820X	821x, 822x	8200 vector	
			Default setting: C0001=3 if C0007 < 52	Default setting: C0001=3 if C0007 > 51
0, 1	00 = C0046 active 01 = JOG1 in C0037 active 10 = JOG2 in C0038 active 11 = JOG3 in C0039 active	00 = C0046 active 01 = JOG1 in C0037 active 10 = JOG2 in C0038 active 11 = JOG3 in C0039 active	00 = C0046 active 01 = NSET1-JOG1 (C0037) active 10 = NSET1-JOG2 (C0038) active 11 = NSET1-JOG3 (C0039) active	freely configurable through user
2	CW/CCW (CW rotation/CCW rotation) 0 = CW rotation 1 = CCW rotation	CW/CCW (CW rotation/CCW rotation) 0 = CW rotation 1 = CCW rotation	DCTRL1-CW/CCW 0 = not active 1 = active	
3	QSP (quick stop) 0 = QSP not active 1 = QSP active	QSP (quick stop) 0 = QSP not active 1 = QSP active	AIF-CTRL-QSP 0 = not active 1 = active	
4	Reserved	RFG stop (stop of the ramp function generator) 0 = RFG stop not active 1 = RFG stop active	NSET1-RFG1-STOP 0 = not active 1 = active	freely configurable through user
5	Reserved	RFG zero (deceleration along the T_{if} ramp C0013) 0 = RFG zero not active 1 = RFG zero active	NSET1-RFG1-0 0 = not active 1 = active	
6	Reserved	UP function for motor potentiometer 0 = UP not active 1 = UP active	MPOT1-UP 0 = not active 1 = active	
7	Reserved	DOWN function for motor potentiometer 0 = DOWN not active 1 = DOWN active	MPOT1-DOWN 0 = not active 1 = active	
8	Reserved	Reserved	freely configurable through user	
9	Ctrl. inhibit (controller inhibit) 0 = controller not inhibited 1 = controller inhibited	Ctrl. inhibit (controller inhibit) 0 = controller not inhibited 1 = controller inhibited	AIF-CTRL-CINH 0 = not active 1 = active	
10	Reserved	Reserved	AIF-CTRL-TRIP-SET 0 = not active 1 = active	
11	Reserved	TRIP reset 0 -> 1 = Edge from 0 to 1	AIF-CTRL-TRIP-RESET 0 -> 1 = Edge from 0 to 1	
12	PAR1 (Parameter set changeover) 0 -> 1 = Parameter set 1 -> 0 = Parameter set	PAR1 (Parameter set changeover) 0 -> 1 = Parameter set 1 -> 0 = Parameter set	DCTRL1-PAR2/4 0 = not active 1 = active	freely configurable through user
13	Reserved	Reserved	DCTRL1-PAR3/4 0 = not active 1 = active	
14	DC brake (DC injection brake) 0 = DC brake not active 1 = DC brake active	DC brake (DC injection brake) 0 = DC brake not active 1 = DC brake active	MCTRL1-DCB 0 = not active 1 = active	
15	Reserved	Reserved	freely configurable through user	

Status word for 82XX and 8200 vector

Bit	820X	821x, 822x	8200 vector factory setting
0	Actual parameter set 0 = Parameter set 1 or 3 active 1 = Parameter set 2 or 4 active	Actual parameter set 0 = Parameter set 1 or 3 active 1 = Parameter set 2 or 4 active	DCTRL1-PAR-B0
1	IMP (pulse inhibit) 0 = Pulses for power stage enabled 1 = Pulses for power stage inhibited	IMP (pulse inhibit) 0 = Pulses for power stage enabled 1 = Pulses for power stage inhibited	DCTRL1-IMP
2	I_{max} (current limit reached) 0 = Current limit not reached 1 = current limit reached	I_{max} (current limit reached) 0 = Current limit not reached 1 = current limit reached	MCTRL1-IMAX
3	Not assigned	$f_d = f_{dset}$ 0 = $f_d \neq f_{dset}$ 1 = $f_d = f_{dset}$	MCTRL1-RFG1=NOUT
4	$f_d = f_{dset}$ 0 = $f_d \neq f_{dset}$ 1 = $f_d = f_{dset}$	RFG on = RFG off 0 = RFG on ≠ RFG off 1 = RFG on = RFG out	NSET1-RFG1-I=0
5	Qmin ($f_d \leq f_{dQmin}$) 0 = Qmin not active 1 = Qmin active	Qmin ($f_d \leq f_{dQmin}$) 0 = Qmin not active 1 = Qmin active	PCTRL1-QMIN
6	$f_d + 0$ (act. frequency = 0) 0 = $f_d \neq 0$ 1 = $f_d + 0$	$f_d + 0$ (act. frequency = 0) 0 = $f_d \neq 0$ 1 = $f_d + 0$	DCTRL1-NOUT=0
7	Ctrl. inhibit (controller inhibit) 0 = controller not inhibited 1 = controller inhibited	Ctrl. inhibit (controller inhibit) 0 = controller not inhibited 1 = controller inhibited	DCTRL1-CINH
8...11	Controller status 0 = Controller initialisation 8 = Error active	Controller status 0 = Controller initialisation 2 = Switch-on inhibit 3 = Operation inhibited 4 = Flying-restart circuit active 5 = DC brake active 6 = Operation enabled 7 = Message active 8 = Error active	Controller status 0 = Controller initialisation 2 = Switch-on inhibit 3 = Operation inhibited 4 = Flying-restart circuit active 5 = DC brake active 6 = Operation enabled 7 = Message active 8 = Error active
12	Overtemperature warning 0 = No warning 1 = warning	Overtemperature warning 0 = No warning 1 = warning	DCTRL1-OH-WARN
13	V_{Gmax} (DC-bus overvoltage) 0 = No overvoltage 1 = overvoltage	V_{Gmax} (DC-bus overvoltage) 0 = No overvoltage 1 = overvoltage	DCTRL1-OV
14	Direction of rotation 0 = CW rotation 1 = CCW rotation	Direction of rotation 0 = CW rotation 1 = CCW rotation	DCTRL1-CCW
15	Ready for operation 0 = not ready for operation 1 = ready for operation	Ready for operation 0 = not ready for operation 1 = ready for operation	DCTRL1-RDY

7.6.6 Process-data assignment for 93XX

With the 93XX controller the process data assignment can be changed by reconfiguring the function blocks AIF-IN and AIF-OUT.

Process data telegram to drive

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

Control word: see (📖 7.6-18)

AIF-IN.W1 to AIF-IN.W3 depend on the signal configuration selected under L-C0005.

For detailed description of the 93XX signal configuration, see the Operating Instructions for 93XX (only the main configurations: 1000, 4000, 5000, etc.) or the Manual 93XX.

In the controller, other signals can be assigned to AIF-IN.W1 to AIF-IN.W3. For this, the function-block configuration - described in the Manual 93XX - is used. The function block AIF-IN determines the input data of the controller as data interface for the 2175 fieldbus module.

For more detailed information about the function block AIF-IN, see the Manual 93XX.

<table>
<tr><th colspan="2">Signal configuration
(L-C0005)</th><th>AIF-IN.W1</th><th>AIF-IN.W2</th><th>AIF-IN.W3</th><th>AIF-IN.D1</th></tr>
<tr><td>Speed control</td><td>1003 / 1013 / 1113</td><td>NSET-N
Speed setpoint
100 % = 16383</td><td rowspan="3">not assigned</td><td rowspan="13">not assigned</td><td rowspan="13">not assigned</td></tr>
<tr><td>Torque control</td><td>4003 / 4013 / 4113</td><td>MCTRL-MADD
Torque setpoint
100 % = 16383</td></tr>
<tr><td>DF master</td><td>5003 / 5013 / 5113</td><td>NSET-N
Speed setpoint
100 % = 16383</td></tr>
<tr><td>DF-slave bus</td><td>6003 / 6013 / 6113</td><td>DFSET-A-TRIM
Phase trimming</td><td>DFSET-N-TRIM
Speed trimming</td></tr>
<tr><td>DF-slave cascade</td><td>7003 / 7013 / 7113</td><td>DFSET-VP-DIV
DF factor</td><td>DFSET-A-TRIM
Phase trimming</td></tr>
<tr><td>Cam profiler</td><td>1xxx3</td><td>YSET1-FACT</td><td rowspan="4">not assigned</td></tr>
<tr><td>Positioning</td><td>2xxx3</td><td>not assigned</td></tr>
<tr><td>vector control</td><td>1xx3 / 2xx3 / 3xx3 / 5xx3 / 100x3</td><td>NLIM-IN1</td></tr>
<tr><td>vector control</td><td>4xx3</td><td>NCTRL-MADD</td></tr>
<tr><td>vector control</td><td>6xx3</td><td>DFSET-A-TRIM</td><td>DFSET-N-TRIM</td></tr>
<tr><td>vector control</td><td>7xx3 / 8xx3 / 9xx3</td><td>DFSET-VP-DIV</td><td>DFSET-A-TRIM</td></tr>
<tr><td>vector control</td><td>100x3</td><td>NLIM-IN1</td><td rowspan="2">not assigned</td></tr>
<tr><td>vector control</td><td>110x3</td><td>not assigned</td></tr>
</table>

Control word for 93XX

9300	9300 Servo				9300 Positioning controller	9300 Cam profiler	9300 Vector		
C0005	1xx3	4xx3	5xx3	6xx3,7xx3	2xxx3	xxx3	1xxx, 2xxx, 3xxx, 5xxx, 10xxx, 11xxx	4xx3	6xx3,7xx3
0	NSET-JOG*1	not assigned	NSET-JOG*1	not assigned	not assigned	CSEL1-CAM*1	NSET-JOG*1	not assigned	not assigned
1	NSET-JOG*2	not assigned	NSET-JOG*2	not assigned	not assigned	CSEL1-CAM*2	NSET-JOG*2	not assigned	not assigned
2	NSET-N-INV	NSET-N-INV	NSET-N-INV	NSET-N-INV	not assigned	CSEL1-CAM*4	NSET-N-INV	not assigned	not assigned
3	AIF-CTRL.QSP	AIF-CTRL.QSP	AIF-CTRL.QSP	AIF-CTRL.QSP	AIF-CTRL.QSP	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	POS-PRG-START	CSEL1-EVENT	NSET-RFG-STOP	NSET-RFG-STOP	not assigned
5	NSET-RFG-0	NSET-RFG-0	NSET-RFG-0	NSET-RFG-0	POS-PRG-STOP	CDATA-CYCLE	NSET-RFG-0	NSET-RFG-0	not assigned
6	not assigned	not assigned	not assigned	not assigned	not assigned	CSEL1-LOAD	not assigned	not assigned	not assigned
7	not assigned	not assigned	not assigned	not assigned	POS-PRG-RESET	CSEL1-LOAD	not assigned	not assigned	not assigned
8	not assigned	not assigned	not assigned	not assigned	not assigned	not assigned	not assigned	not assigned	not assigned
9	AIF-CTRL.CINH	AIF-CTRL.CINH	AIF-CTRL.CINH	AIF-CTRL.CINH	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	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	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	POS-PS-CANCEL	not assigned	DCTRL-PAR*1	DCTRL-PAR*1	DCTRL-PAR*1
13	DCTLR-PAR-LOAD	DCTLR-PAR-LOAD	DCTLR-PAR-LOAD	DCTLR-PAR-LOAD	POS-PARAM-RD	not assigned	DCTLR-PAR-LOAD	DCTLR-PAR-LOAD	DCTLR-PAR-LOAD
14	NSET-Ti*1	NSET-JOG*1	REF-ON	REF-ON	POS-LOOP-ONH	not assigned	NSET-Ti*1	NSET-JOG*1	not assigned
15	NSET-Ti*2	NSET-JOG*2	NSET-Ti*1	not assigned	POS-STBY-STP	not assigned	NSET-Ti*2	NSET-JOG*2	not assigned

Note!

The single bit control commands of the control word depend on other bit positions.

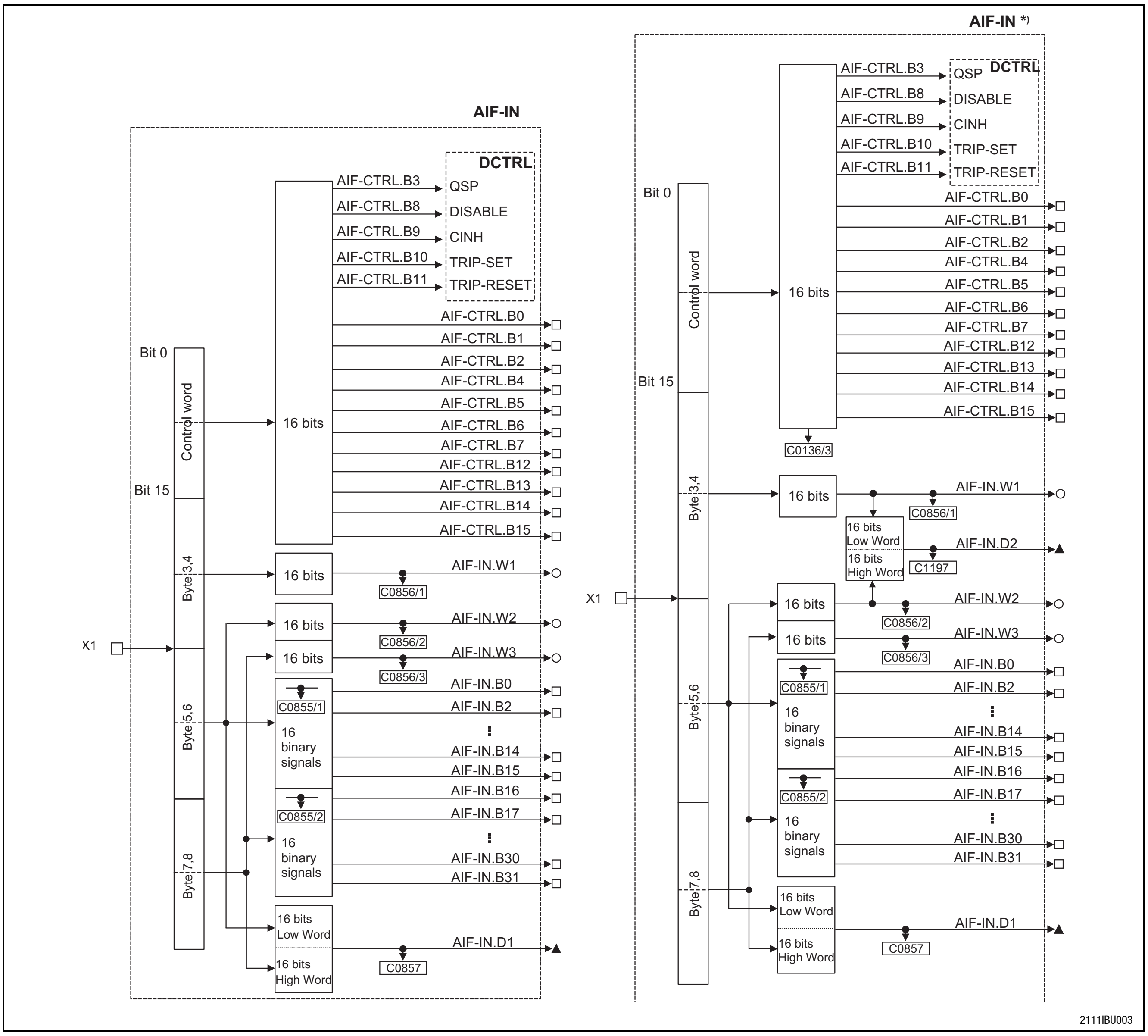


Fig. 7.6-8 Function blocks AIF-IN and AIF-IN*)
AIF-IN*) is available for the 9300 technology variants: servo, positioning controller and cam profiler as of software version 2.0. AIF-IN.D2 is new.

Process data telegram <u>from</u> drive

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

Status word: see (📖 7.6-21)

AIF-OUT.W1 to AIF-OUT.W3 depend on the signal configuration selected under L-C0005.

For detailed description of the 93XX signal configuration, see the Operating Instructions for 93XX (only the main configurations: 1000, 4000, 5000, etc.) or the Manual 93XX.

In the controller, other signals can be assigned to AIF-OUT.W1 to AIF-OUT.W3. For this, the function-block configuration - described in the Manual 93XX - is used. The function block AIF-OUT determines the output data of the controller as data interface for the 2175 fieldbus module.

For more detailed information about the function block AIF-OUT, see the Manual 93XX.

Signal configuration (L-C0005)		AIF-OUT.W1	AIF-OUT.W2	AIF-OUT.W3	AIF-OUT.D1
Speed control	1003/1013/1113	MCTRL-NACT Actual speed 100 % = 16383	MCTRL-MSET2 Torque display 100 % = 16383	MCTRL-NSET2 Speed controller input 100 % = 16383	not assigned
Torque control	4003/4013/4113	MCTRL-MSET2 Torque display 100 % = 16383	MCTRL-NACT Act. speed in % 100 % = 16383	MCTRL-NSET2 Speed controller input 100 % = 16383	
DF master	5003/5013/5113	MCTRL-NACT Actual speed 100 % = 16383	MCTRL-MSET2 Torque display 100 % = 16383	MCTRL-NSET2 Speed controller input 100 % = 16383	
DF-slave bus	6003/6013/6113	MCTRL-NACT Actual speed 100 % = 16383	MCTRL-PHI-ACT Actual phase	MCTRL-MSET2 Torque setpoint in % 100 % = 16383	
DF-slave cascade	7003/7013/7113	MCTRL-NACT Actual speed 100 % = 16383	MCTRL-PHI-ACT Actual phase	MCTRL-MSET2 Torque setpoint in % 100 % = 16383	
Cam profiler	1xxx3	MCTRL-NACT Actual speed 100 % = 16383	not assigned	not assigned	
Positioning	2xxx3	MCTRL-NACT Actual speed 100 % = 16383	not assigned	not assigned	
vector control	1xx3/4xx3/5xx3/ 10xx3	MCTRL-NACT Actual speed 100 % = 16383	MCTRL-IACT	MCTRL-NSET2 Speed controller input 100 % = 16383	
vector control	6xx3/7xx3/8xx3/ 9xx3	MCTRL-NACT Actual speed 100 % = 16383	MCTRL-PHI-ANA	MCTRL-MSET2 Torque setpoint in % 100 % = 16383	
vector control	110x3	not assigned	not assigned	not assigned	

For more detailed information about the function block AIF-OUT, see the Manual 93XX.

Status word for 93XX

9300	Servo				Servo positioning controller	Servo cam profiler	vector		
C0005	**1xx3**	**4xx3**	**5xx3**	**6xx3,7xx3**	**2xxx3**	**1xxx3**	**xxx, 2xxx, 3xxx, 5xxx, 10xxx, 11xxx**	**4xxx**	**6xxx, 7xxx, 8xxx, 9xxx**
0	DCTRL-PAR1-0	DCTRL-PAR1-0	DCTRL-PAR1-0	DCTRL-PAR1-0	not assigned	CERR1-ERR	DCTRL-PAR1-0	DCTRL-PAR1-0	DCTRL-PAR1-0
1	DCTRL-IMP	DCTRL-IMP	DCTRL-IMP	DCTRL-IMP	DCTRL-IMP	DCTRL-IMP	DCTRL-IMP	DCTRL-IMP	DCTRL-IMP
2	MCTRL-IMAX	MCTRL-IMAX	REF-OK	REF-OK	POS-REF-OK	MCTRL-IMAX	MCTRL-IMAX	MCTRL-IMAX	MCTRL-IMAX
3	MCTRL-MMAX	not assigned	MCTRL-MMAX	not assigned	not assigned	MCTRL-MMAX	MCTRL-MMAX	MCTRL-IMAX negated	MCTRL-MMAX
4	NSET-RFG-I=0	MCTRL-IMAX negated	NSET-RFG-I=0	MCTRL-IMAX negated	MCTRL-MMAX negated	DCTRL-TRIP	NSET-RFG-I=0	NSET-RFG-I=0	NSET-QSP-OUT
5	QMIN	QMIN	REF-BUSY	REF-BUSY	POS-IN-TARGET	CDATA-X0	QMIN	QMIN	QMIN
6	DCTRL-NACT=0	DCTRL-NACT=0	DCTRL-NACT=0	DCTRL-NACT=0	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	DCTRL-CINH	DCTRL-CINH	DCTRL-CINH	DCTRL-CINH
8 ... 11	Controller status: 0 = Unit initialisation 1 = Switch-on inhibit 3 = Operation inhibited 4 = Flying-restart circuit active 5 = DC-injection brake active 6 = Operation enabled 7 = Message active 8 = Fault active 10 = Fail-QSP (only 9300 servo positioning controller)								
12	DCTRL-WARN	DCTRL-WARN	DCTRL-WARN	DCTRL-WARN	DCTRL-WARN	DCTRL-WARN	DCTRL-WARN	DCTRL-WARN	DCTRL-WARN
13	DCTRL-MESS	DCTRL-MESS	DCTRL-MESS	DCTRL-MESS	DCTRL-MESS	DCTRL-MESS	DCTRL-MESS	DCTRL-MESS	DCTRL-MESS
14	DCTRL-CW/CCW	DCTRL-CW/CCW	DCTRL-CW/CCW	not assigned	DCTRL-AIFL-QSP	DCTRL-CW/CCW	DCTRL-CW/CCW	DCTRL-CW/CCW	DCTRL-CW/CCW
15	DCTRL-RDY	DCTRL-RDY	DCTRL-RDY	DCTRL-RDY	DCTRL-RDY	DCTRL-RDY	DCTRL-RDY	DCTRL-RDY	DCTRL-RDY

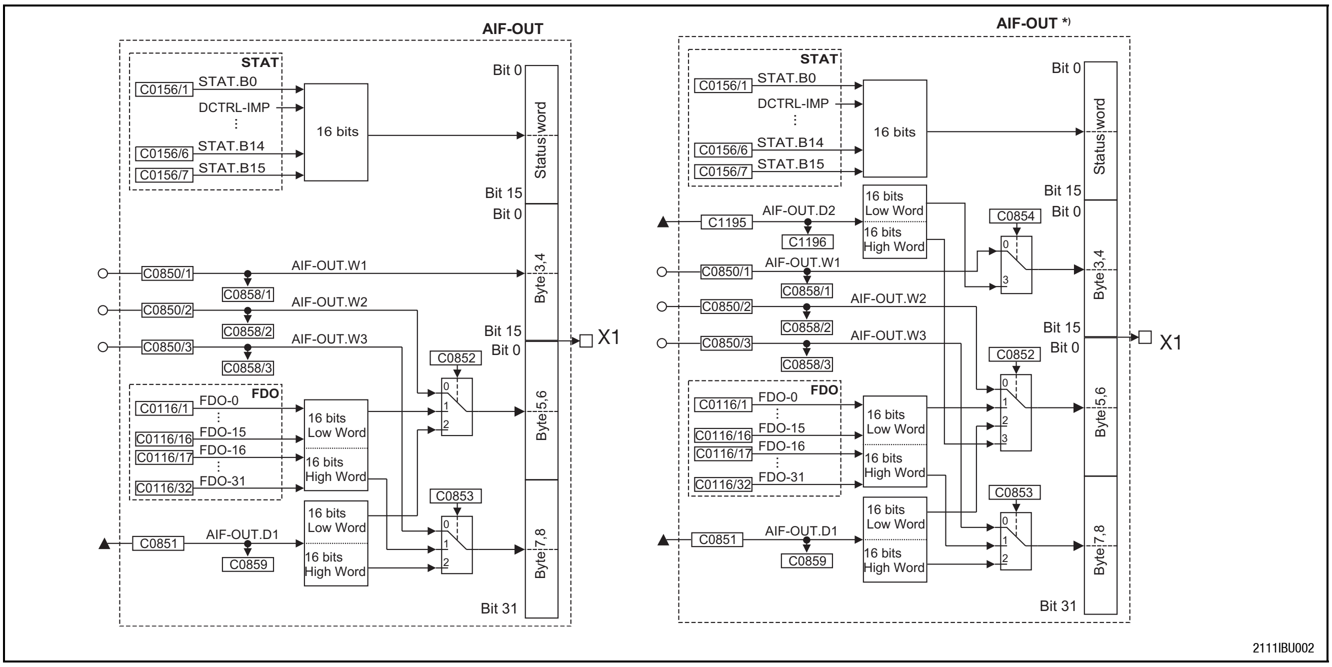


Fig. 7.6-9 Function blocks AIF-OUT and AIF-OUT*)
AIF-OUT*) is available for the 9300 technology variants: servo, positioning controller and cam profiler as of software version 2.0. AIF-OUT.D2 is new.

7.6.7 Process data assignment for 9300 Servo PLC and Drive PLC

Process data telegram to drive

The following data can be assigned to the PO data:

Name/variable name	Explanation
Device control word (AIF1_wDctrlCtrl)	
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

Please execute the following logic operations in the PLC program of the controller:

AIF1_wDctrlCtrl → DCTRL_wAIF1Ctrl

DCTRL_wStat → AIF1_wDctrlStat

Drive PLC

As we are dealing with a PLC here, it is necessary to use the device control.

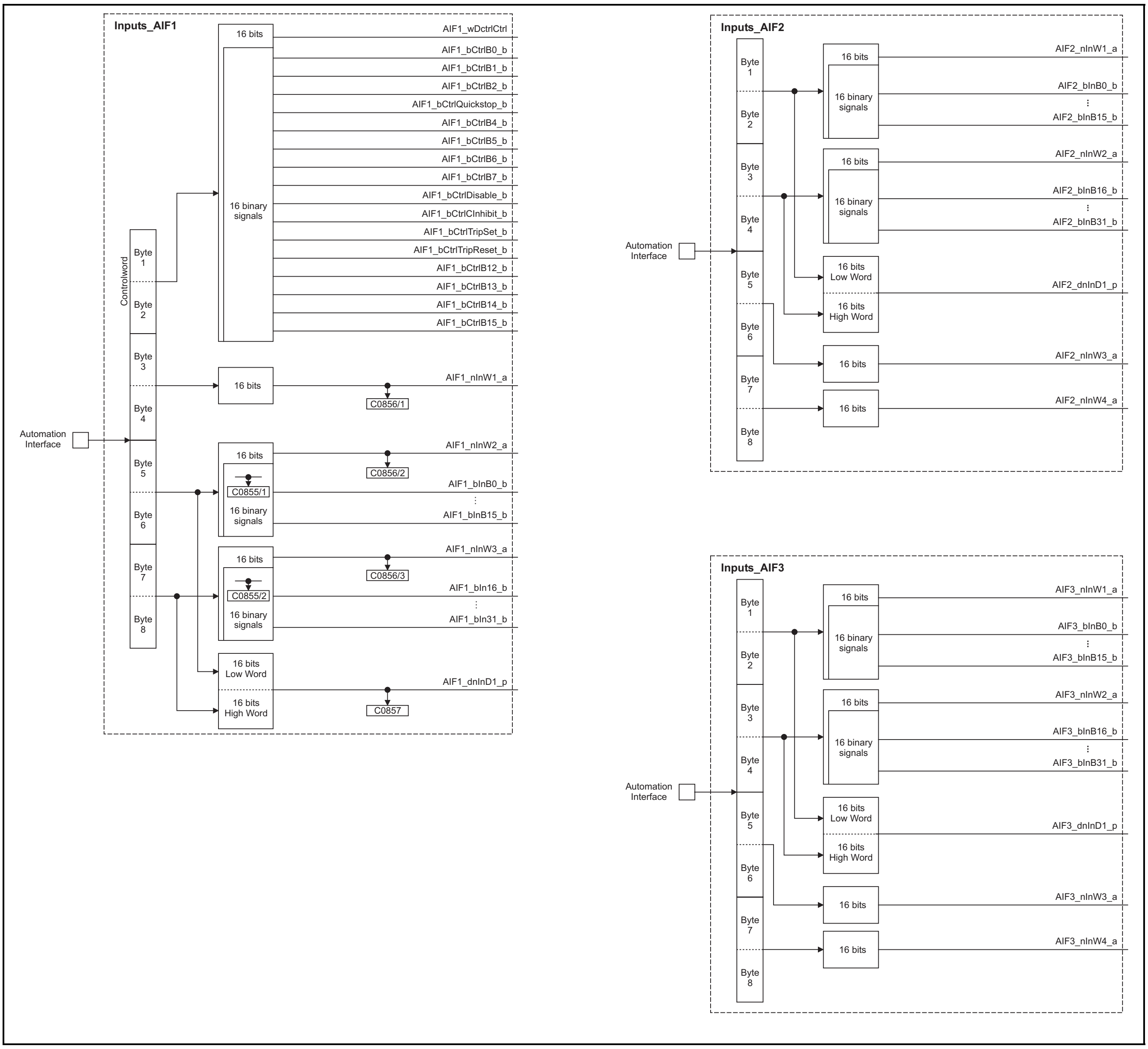


Fig. 7.6-10 Function blocks AIF-IN1, AIF-IN2 and AIF-IN3

Process data telegram from drive

The following data can be assigned to the PI data:

Name/variable name	Explanation
Device status word (AIF1_DctrlStat)	
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

Note!

9300 Servo PLC

Please execute the following logic operations in the PLC program of the controller:

AIF1_wDctrlCtrl ➝ DCTRL_wAIF1Ctrl

DCTRL_wStat ➝ AIF1_wDctrlStat

Drive PLC

As we are dealing with a PLC here, it is necessary to use the device control.

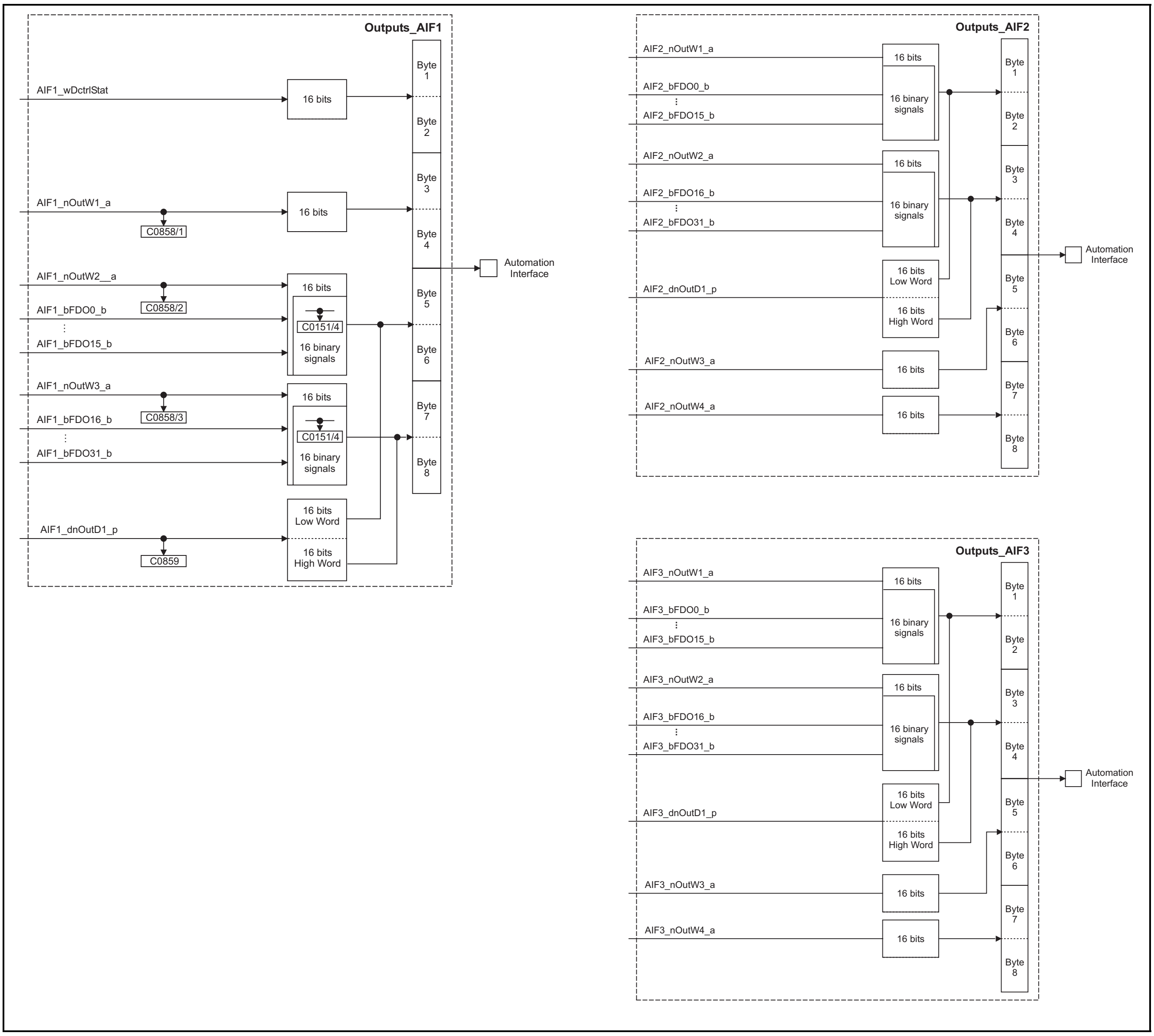


Fig. 7.6-11 Function blocks AIF-OUT1, AIF-OUT2 and AIF-OUT3

7.6.8 Parameter data channel

Note!

- In these Operating Instructions Lenze codes have an L in front to ensure that they are not mixed up with the CANopen index.
  – Example: 'L-C0001' stands for the Lenze code C0001.
- Please obtain the value range for the Lenze codes from the Operating Instructions for the corresponding controller (see: Code table).

Access to the codes of the controller

When using (intelligent) bus modules, it is possible to change the features and behaviour of each controller integrated into the network via a higher-level master (e.g. a PLC).

Lenze controllers store the parameters to be changed in codes.

The controller codes are addressed via the index when accessing the codes through the bus module (see chapter 5.6.8).

The index for Lenze code numbers is between 16576 ($40C0_{hex}$) and 24575 ($5FFF_{hex}$).

Conversion formula:

$$Index[dec] = 24575 - Lenze\ code$$

Example

Lenze codes		dec	hex
	• Addressing of Lenze codes via offset:	Index = 24575 - LENZE CODENO	$Index_{hex}$ = $5FFF_{hex}$ - LENZE-$CODENO_{hex}$
	– Example for operating mode L-C0001	Index = 24574 (= 24575 - 1)	$Index_{hex}$= $5FFE_{hex}$ (= $5FFF_{hex}$ - 1)

The parameter value is included in the useful data of the telegram (see examples (🕮 7.6-31)).

Lenze Parameter sets

Parameter sets are for special code saving which is necessary because of different configurations for different application processes.

The following table informs about number and addressing of parameter sets for your controller:

<table>
<tr><th>82XX</th><th>8200 vector</th><th>93XX</th></tr>
<tr><td colspan="2">The 82XX and 8200 vector have 2 and 4 parameter sets. The parameters can be directly addressed via CAN. They are addressed by means of a code-digit offset:</td><td rowspan="7">93XX controllers have 4 parameter sets (depending on the variant) for saving in the EEPROM. Another parameter set is in the user memory of the controller. This is the current parameter set. Only the current parameter set can be directly addressed through CAN. For the codes, see the Operating Instructions or Manual for 93XX. Changes of the current parameter set will be lost after switching off the controller. Code C0003 is for saving the current parameter set. After switching on the controller, parameter set 1 is automatically loaded into the current parameter set.
At first the parameter sets 2 - 4 must be activated, before the parameters included in them can be changed.</td></tr>
<tr><td colspan="2">• Offset 0 addresses parameter set 1 with the Lenze codes L-C0000 to L-C1999
• Offset 2000 addresses parameter set 2 with the Lenze codes L-C2000 to L-C3999</td></tr>
<tr><td>no additional parameter sets available.</td><td>• Offset 4000 addresses parameter set 3 with the Lenze codes L-C4000 to L-C5999
• Offset 6000 addresses parameter set 4 with the Lenze codes L-C6000 to L-C7999</td></tr>
<tr><td colspan="2">If a parameter is available only once (see the Operating Instructions for 82XX or 8200 vector), use the code digit offset 0.</td></tr>
<tr><td colspan="2">Example for L-C0011 (maximum field frequency):
L-C0011 in parameter set 1: Lenze code = 11
L-C0011 in parameter set 2: Lenze code = 2011</td></tr>
<tr><td>-</td><td>L-C0011 in parameter set 3: Lenze code = 4011
L-C0011 in parameter set 4: Lenze code = 6011</td></tr>
<tr><td colspan="2">Parameter changes:
82XX: Automatic saving in the controller
8200 vector: Automatic saving is factory-set (can be switched off with L-C0003)
Process data changes:
82XX, 8200 vector: no automatic saving</td></tr>
</table>

STOP

Stop! (only applies to 8200 vector and 82XX controllers and the 2175 fieldbus module)
Please observe that cyclic writing of parameter data into the EEPROM is not permissible.
Only for 8200 vector:
Please configurate the code to C0003 = 0 after each mains disconnection if you want to change the parameter data cyclically.

Structure of parameter data telegram

User data (up to 8 byte)							
1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
Command	**Index** Low byte	**Index** High byte	**Subindex**	**Data 1**	**Data 2**	**Data 3**	**Data4**
				Low Word		High Word	
				Low byte	High byte	Low byte	High byte
				Error message			

Note!

The user data are represented in a left-justified INTEL format.

Calculation example, see chapter 7.6.9.

“Command”

The command contains the following information which must be entered if not already indicated:

Command	**Access to Data 1 - Data 4**			**Block**
	4 byte data (5th - 8th byte)	**2 byte data** (5th + 6th byte)	**1 byte data** (5th byte)	
	hex	hex	hex	hex
Write request (Send parameters to drive)	23	2B	2F	Writing not possible
Write Response (Controller response to the write request (acknowledgement))	60	60	60	
Read Request (Request to read a parameter from the drive)	40	40	40	40
Read Response (Response to the read request with an actual value)	43	4B	4F	41
Error response (The controller indicates a communication error)	80	80	80	80

“Index Low Byte / Index High Byte”

The parameters or the Lenze codes are selected with these two bytes according to the formula:

$$Index = 24575 - (Lenze\ code + 2000 \cdot (parameter\ set - 1))$$

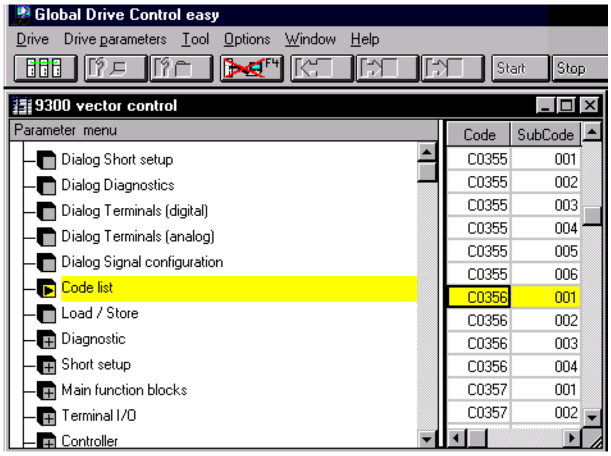
Example	**Calculation**	**Index Low/High Byte**
The code L-C0012 (acceleration time) in parameter set 1 is to be addressed.	24575 - 12 - 0 = 24563 = $5FF3_{hex}$	According to the left-justified Intel data format the entries are (see chapter 5.6.8): Index Low Byte = $F3_{hex}$ Index High Byte = $5F_{hex}$
The code L-C0012 (acceleration time) in parameter set 2 is to be addressed.	An offset of 2000 is to be added because of parameter set 2: 24575 - 12 - 2000 = 22563 = 5823_{hex}	According to the left-justified Intel data format the entries are (see chapter 5.6.8): Index Low Byte = 23_{hex} Index High Byte = 58_{hex}

"Subindex"

Table position of a parameter value under the index.

Example:

L-C0356. This code consists of 4 subcodes (see below). This results in the following entries for the subindex: 1 - 4_{hex} (1 - 4_{dec})



2171CAN001

Data
(Data 1 ... data 4)

<table>
<tr><td rowspan="6">Details about the parameter data telegram</td><td>Parameter value (length 1)</td><td>00</td><td>00</td><td>00</td></tr>
<tr><td colspan="2">Parameter value (length 2)</td><td rowspan="2">00</td><td rowspan="2">00</td></tr>
<tr><td>Low byte</td><td>High byte</td></tr>
<tr><td colspan="4">Parameter value (length 4)</td></tr>
<tr><td colspan="2">Low Word</td><td colspan="2">High Word</td></tr>
<tr><td>Low byte</td><td>High byte</td><td>Low byte</td><td>High byte</td></tr>
</table>

Depending on the data format (see 'Attribute list' in the Manual of the corresponding controller) the length of the parameter value needs 1 to 4 bytes.

Note!

Lenze parameters are mainly described as data type FIX32 (32 bit value with sign and four decimal positions, see attribute table in the corresponding system manual). In order to obtain integer values, the desired parameter value must be multiplied by 10.000_{dec}.

The parameters C0135 and C0150 must be transferred as bit code and without factor.

The attribute table of the drive controller (see Operating Instructions) contains notes concerning the parameters for which the factor 10.000_{dec} is to be considered.

Fault messages

Details about the parameter data telegram	Command	Index Low byte	Index High byte	Subindex	**error code**

Byte 1:
In the **Command** byte shows in code **128_{dec} or 80_{hex}** that an error has occured.

Byte 2, 3 and 4:
In the **Index byte** and **Subindex** the index and subindex of that code in which the error has occured, are entered.

Byte 5 - 8:
In the data bytes 5. - 8. the **error code** is entered.

The error code is described in reversed direction compared to the read direction.

Example: Error code 06 04 00 41_{hex} and description of error code:

← Read direction of error code			
41	**00**	**04**	**06**
5th byte	6th byte	7th byte	8th byte
Low Word		High Word	
Low byte	High byte	Low byte	High byte

The following table lists the explanations for the error numbers:

error code (hex)	**Explanation**
0503 0000	Toggle bit not changed
0504 0000	SDO protocol has been terminated
0504 0001	Invalid or unknown client/server command specifier
0504 0002	Invalid block size (only block mode)
0504 0003	Invalid sequence number (only block mode)
0504 0004	CRC fault (only block mode)
0504 0005	Not sufficient memory
0601 0000	Access to object is not supported
0601 0001	Attempt to read a writable object
0601 0002	Attempt to write a readable object
0602 0000	Object is not listed in the object directory
0604 0041	Object is non-transferable to PDO
0604 0042	Number and length of the objects to be transferred exceed length of PDO
0604 0043	General parameter incompatibility
0604 0047	General internal device incompatibility
0606 0000	Access denied due to a hardware error
0607 0010	Unsuitable data type, unsuitable service parameter length
0607 0012	Unsuitable data type, service parameter length exceeded
0607 0013	Unsuitable data type, service parameter length has not been reached
0609 0011	Subindex is not available
0609 0030	Value range of the parameter is exceeded
0609 0031	Parameter values are too high
0609 0032	Parameter values are too low
0609 0036	Maximum value falls below minimum value
0800 0000	General fault
0800 0020	Data cannot be transferred or saved for application
0800 0021	Data cannot be transferred or saved for application due to local control
0800 0022	Data cannot be transferred or saved due to the current device status
0800 0023	Dynamic generation of the object directory has been failed or the object directory is not available (e.g. object directory is created from a file, generation is not possible due to a file error)

7.6.9 Examples for parameter data telegrams

Read parameters

The heatsink temperature (value of 43 °C) C061 is to be read of the controller with the device address 5 via parameter channel 1.

- Identifier calculation

Identifier parameter channel 1 to controller	= 1536 + controller address
Identifier	= 1536 + 5 = 1541

- Command read request (request to read a parameter from the drive)

Command	= 40_{hex}

- Index calculation

Index = 24575 - code number	Index = 24575 - 61 = 24514 = $5FC2_{hex}$

Telegram to drive:

Identifier	Command	Index Low Byte	Index High Byte	Subindex	Data 1	Data 2	Data 3	Data4
1541	40_{hex}	$C2_{hex}$	$5F_{hex}$	00	00	00	00	00

Telegram from drive

Identifier:
Parameter channel 1 of controller (=1408) + controller address = 1413

Command:
Response to the read request with the actual value = 43_{hex}

Index of the read request
$5FC2_{hex}$

Subindex:
0

Data 1 to Data 4:
00 06 8F B0 = 430.000 → 430.000 : 10.000 = 43 °C

Identifier	Command	Index Low Byte	Index High Byte	Subindex	Data 1	Data 2	Data 3	Data4
1413	43_{hex}	$C2_{hex}$	$5F_{hex}$	00	$B0_{hex}$	$8F_{hex}$	06_{hex}	00

Write parameters

The acceleration time C0012 (parameter set 1) of the controller with address 1 is to be changed to 20 s via parameter channel 1.

- Identifier calculation

Identifier parameter channel 1 to controller	= 1536 + controller address
Identifier	= 1536 + 1 = 1537

- Command write request (send parameter to drive)

Command	= 23_{hex}

- Index calculation

Index = 24575 - code number	Index = 24575 - 12 = 24563 = $5FF3_{hex}$

- Subindex: 0
- Calculation of the acceleration time

Acceleration-time value	20 s · 10.000 = 200.000 = 00 03 0D 40_{hex}

- Telegram to drive

Identifier	Command	Index Low Byte	Index High Byte	Subindex	Data 1	Data 2	Data 3	Data4
1537	23_{hex}	$F3_{hex}$	$5F_{hex}$	00	40_{hex}	$0D_{hex}$	03_{hex}	00

Response of the controller when no error occurs

Identifier	Command	Index Low Byte	Index High Byte	Subindex	Data 1	Data 2	Data 3	Data4
1409	60_{hex}	$F3_{hex}$	$5F_{hex}$	00	00	00	00	00

Identifier parameter channel 1 from controller = 1408 + controller address = 1409

Command = write response (controller response (acknowledgement)) = 60_{hex}

Read block parameters

A product code (EKZ, code L-C0200) of the Lenze product 8200 vector is to be read from parameter set 1. The product code has 14 alphanumerical characters. It is transferred as a block parameter. A transfer of block parameters uses the entire data width (2nd - 8th byte).

The command byte (1. byte) contains the entry (40_{hex} or 41_{hex}) during the transfer of useful data in order to be able to

– signalise the end of the block transfer
– request the next block.

- Code L-C0200 - request

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
40_{hex}	37_{hex}	$5F_{hex}$	00	00	00	00	00

1st byte: 40 read request (request to read a parameter from the controller)
2nd/3rd byte: Index Low/High Byte: 24575 - 200 - 0 = 24375 = $5F37_{hex}$

- Response including the block length (14 characters)

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
41_{hex}	37_{hex}	$5F_{hex}$	00	$0E_{hex}$	00	00	00

1st byte: 41 read response. The entry 41_{hex} implies that it is a block telegram.
2nd/3rd byte: see above
5th byte: 0E (=$14_{dec.}$) data length 14 characters (ASCII format)

- First data block - request

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
60_{hex}	00	00	00	00	00	00	00

1st byte: 60_{hex}
Write response (acknowledgement) with access to bytes 2 - 8.
Note:
The single blocks are toggled* in succession, i.e. at first it is requested with command 60_{hex} (=0110 0000_{bin}), then with command 70_{hex} (=0111 0000_{bin}), after this again with 60_{hex} etc. The response is send accordingly. It is alternating because of a toggle bit. The process is completed by command 11_{hex} (Bit 0 is set, see below).
*Toggle bit = Bit 4 (starting at 0)

- Response

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
00	38_{hex}	32_{hex}	53_{hex}	38_{hex}	32_{hex}	31_{hex}	32_{hex}

2nd byte - 8th byte, ASCII format: 8 2 S 8 2 1 2

- Second data block - request

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
70_{hex}	00	00	00	00	00	00	00

1st byte: 70_{hex} (Toggle) write response (acknowledgement) with access to all 4 data bytes

- Second data block - response with over-detection

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
11_{hex}	56_{hex}	$5F_{hex}$	31_{hex}	34_{hex}	30_{hex}	30_{hex}	30_{hex}

1st Byte: 11 last transfer of the data block
2nd byte - 8th byte: V _ 1 4 0 0 0
Result of data block transfer: 82S8212V_14000

7.6.10 Notes to be observed when setting the parameters for the controllers

82XX controllers

The following applies to the inverter series 8200:

Danger!

Parameter setting (codes except C046, C0135) is only possible when the controller is inhibited. Parameters are accepted when the controller is enabled, but they are not saved.

After having set a parameter, the controller must not be addressed for approx. 50 ms; otherwise the command will be ignored.

After parameter setting, the controller needs up to approx. 70 ms to set the status 'enabled' (terminal, C040, C0135).

The function TRIP reset is activated by inhibiting the controller and enabling it again under C040 or C0135.

The function TRIP-Reset initialises the 8200 inverter and the 2175 field bus module. Therefore the TRIP reset command is not acknowledged for the master.

Controller 8200 vector

Digital and analog input and output signals can be freely configured (see Operating Instructions; codes C0410, C0412, C0417 and C0421)

Basic devices servo PLC 9300 and DRIVE PLC

AIF control/status byte

Drive controller and fieldbus module are able to exchange control and status information. For this purpose a control byte and a status byte are made available in the AIF protocol.

The program package “Drive PLC Developer Studio” (DDS) enables the user to directly access the status byte via the control configuration of the AIF module. The 2175 fieldbus module describes this byte with its fault messages.

The evaluation of these fault messages must be taken into consideration by the programmer of the PLC series, e.g. by displaying the signals as faults via a fault-warning-message-firmware module.

Assignment of the AIF status byte C2121 for the 2175 fieldbus module

AIF status bit	Function
0	CE11 fault
1	CE12 fault
2	CE13 fault
3	CE14 fault
4	Operational
5	Pre-Operational
6	Warning
7	internally assigned

The control byte is used, so that the controller can send messages or commands to the 2175 fieldbus module.

The control byte is available for the user via code C2120. The commands are described as numbers. Some of the command numbers are universally applicable for all fieldbus modules, but others are only valid especially for the different modules. Altogether not more than 16 commands can be available.

Command number in C2120, bit 0..3	Function	R/W code
0	No command	
1	Re-initialisation of the option module	all
2	Updating of all relevant codes for the actual option module. No re-init.	all
10	Accept XCAN-OUT cycle times or boot-up time	C2356
11	Accept monitoring times	C2357
12	Accept TX mode	C2375
13	Accept masks	C2376-C2378
14	Accept configuration of CE faults	C2382

Assignment of the AIF control byte for the 2175 fieldbus module

The command code C2120 is automatically reset to 0 after the command has been transferred to the option module. Because of this, no toggle bit is needed. The command itself is written into the lower 4 bits of the control byte, so that 4 bits are available for extensions to come.

7.7 CANopen objects and Lenze codes

7.7.1 Communication relevant Lenze codes

The behaviour of servo inverters or frequency inverters is determined by the corresponding parameter setting. Lenze devices can be parameterised by means of codes. These Lenze codes which are part of a telegram are exchanged between the master and 2175 fieldbus module via the CAN bus.

According to the Lenze inverter used, two Lenze codes are available for the communication via CAN bus:

- Codes ≥ L-C2350
  for devices with PLC operating system according to IEC1131 (e. g. Servo PLC 9300 and Drive PLC).
  These codes are saved in the controller.
- Module codes ≥ L-C1850
  for all other inverters (82XX, 8200 vector or 93XX).
  These codes are saved in the 2175 fieldbus module.

The condition for communication with the drive is that the drive is known as participant in the system. The recognition takes place during the module initialisation.

The setting of address and Baud rate can be made in different ways (see also from (📖 7.5-1)):

- Front switch 1 - 6 ≠ OFF
  Address and baud rate are defined by the switch setting.
  The 2175 fieldbus module writes the front switch position into the code L-C1859 or L-C2359.
- Front switch 1 - 6 = OFF
  Address and baud rate must be defined by the codes L-C0009 (address) and L-C0125 (baud rate).

7.7.2 Description of communication relevant Lenze codes

L-C1810:
Software code

Code	Subcode	Index	Possible settings		Data type
			Lenze	Selection	
L-C1810	-	22765_d = $58ED_h$	-	-	VS

When initialising the modules it can be determined with the help of the product code, which device is connected as participant.

L-C1811:
Software creation date

Code	Subcode	Index	Possible settings		Data type
			Lenze	Selection	
L-C1811	-	22764_d = $58EC_h$			VS

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

L-C1850/L-C2350:
Node address

Code	Subcode	Index	Possible settings		Data type
			Lenze	Selection	
L-C1850	-	$58C5_h$ = 22725_d	1	1 [1] 63	FIX32
L-C2350	-	$56D1_h$ = 22225_d	1	1 [1] 63	FIX32

The code serves to set the address of the 2175 module via CAN bus.

The code L-C1850 is an image of code L-C0009 of the basic device. Writing L-C1850 has a direct effect on L-C0009.

Note!

This code is only effective if the DIP switches S1-S6 are set to position OFF before mains switching.
Changing the node address only gets effective by renewed mains disconnection of the 2175 module or by sending the network manager command *Reset_Node* or *Reset_Communication* via CAN bus to the module.
With Servo PLC 9300 / Drive PLC this is also possible by assigning value 1 to the code C2120 (AIF control byte).

L-C1851/L-C2351: baud rate

Code	Subcode	Index	Possible settings			Data type
			Lenze	Selection		
L-C1851	-	$58C4_h$ = 22724_d	0	0 6	0: 500 kbits/s 1: 250 kbits/s 2: 125 kbits/s 3: 50 kbits/s 4: 1000 kbits/s 5: 20 kbits/s 6: 10 kbits/s	FIX32
L-C2351	-	$56D0_h$ = 22224_d		0 4		

The code serves to set the Baud rate of the 2175 module.

Note!

This code is only effective if the DIP switches S1-S6 are set to position OFF before mains switching.

Changing the baud rate only becomes effective by renewed mains disconnection of the 2175 module or by sending the network manager command *Reset_Node* or *Reset_Communication* via CAN bus to the module.

With Servo PLC 9300 / Drive PLC this is also possible by assigning value 1 to the code C2120 (AIF control byte).

The code L-C1851 is an image of code L-C0125 which is located in the basic device. This means that describing L-C1851 has a direct effect on L-C0125.

L-C1852/L-C2352: Master/slave operation

Code	Subcode	Index	Possible settings		Data type
			Lenze	Selection	
L-C1852	-	$58C3_h$ = 22723_d	0	0 = Slave operation 1 = Master operation	FIX32
L-C2352	-	$56CF_h$ = 22223_d			

After switch-on the module has the state PRE-OPERATIONAL. In this state only an exchange of parameter data (SDO's) is possible.

In slave operation, the module stays in this state until it is put by the network manager command *Start_Remote_Node* into the state OPERATIONAL.

In the state OPERATIONAL also process data (PDO's) are exchanged besides parameter data (SDO's).

In master operation, the network manager command *Start_Remote_Node* is transmitted after an adjustable boot-up time, which puts all nodes into the state OPERATIONAL.

Note:
The network manager command *Start_Remote_Node* is a "Broadcast" telegram which is directed to **all** other nodes.

Note!

The change of the master/slave operation only becomes effective by renewed mains disconnection of the 2175 fieldbus module or by sending one of the network manager commands "Reset_Node" or "Reset-Communication" via the CAN bus to the fieldbus module.

With Servo PLC 9300 / Drive PLC this is also possible by assigning value 1 to the AIF control byte.

L-C1853/L-C2353: Addressing CAN-INx/CAN-OUTx

Code	Subcode	Index	Possible settings		Data type	Explanation
			Lenze	Selection		
L-C1853	/1 CAN-IN1/OUT1 /2 CAN-IN2/OUT2 /3 CAN-IN3/OUT3	$58C2_h = 22722_d$	0	0 [1] 3	FIX32	0: Addressing to CANopen (Default identifier) 1: Addressing to L-C1854/ L-C2354 2 : Addressing to LENZE system bus 3 : Addressing to CANopen index $14XX_h/18XX_h$
L-C2353		$56CE_h = 22222_d$		0 [1] 2	FIX32	

Via this code, the source for the resulting addresses of the CAN-INx/OUTx process data objects (PDO's) on the CAN bus is selected.

Note!

Changing the source address in one or several subcodes only gets effective by renewed mains disconnection of the module or by sending the network manager command *Reset_Node* or *Reset_Communication* via CAN bus to the 2175 module.

With Servo PLC 9300 / Drive PLC this is also possible by assigning value 1 to the code C2120 (AIF control byte).

Relation to CANopen

The addressing of the corresponding PDO pair or subcode under code L-C1853 / L-C2353 is switched to CANopen indices $14XX_h/18XX_h$ (see above: value 3), if the CANopen indices 1400_h, 1401_h, 1402_h, 1800_h, 1801_h or 1802_h are assigned with a new value.

Calculation of the identifiers

PDO	Value L-C1853 / L - C2353				
	0	1		2	3
	to CANopen (default identifier)	via code L-1854/L-2354	Default setting	via Lenze system bus	to CANopen index
CAN-IN1	512 + node address	384 + C1854/1 or 384 + C2354/1	384 + 129	512 + node address	Index 1400_h, subindex 1
CAN-IN2	768 + node address	384 + C1854/3 or 384 + C2354/3	384 + 257	640 + node address	Index 1401_h, subindex 1
CAN-IN3	1024 + node address	384 + C1854/5 or 384 + C2354/5	384 + 385	768 + node address	Index 1402_h, subindex 1
CAN-OUT1	384 + node address	384 + C1854/2 or 384 + C1854/2	384 + 1	384 + node address	Index 1800_h, subindex 1
CAN-OUT2	640 + node address	384 + C1854/4 or 384 + C1854/4	384 + 258	641 + node address	Index 1801_h, subindex 1
CAN-OUT3	896 + node address	384 + C1854/6 or 384 + C1854/6	384 + 386	769 + node address	Index 1802_h, subindex 1

Addressing to CANopen (Default identifier)

This is the Lenze setting of the 2175 fieldbus module. The calculation consists of the basic identifier and the node address. The basic identifier corresponds to the preset value according to DS301 V4.01 (page 9-56, 9-85ff).

Addressing to L-C1854 / L-C2354

In case of this addressing the identifier is the sum of a fixed basic identifier 384 (180_{hex}) + the value of the corresponding subcode of L-C1854 / L-C2354. Here the node address has no influence anymore.

This configuration has been created in accordance with 8200 vector/motec controllers with E82ZAFCCxxx function module, system bus (CAN), and the integrated system bus interface of the 93XX controllers (code L-C0353 / L-C0354, selective System bus address).

Addressing to Lenze system bus

This setting corresponds to the preset calculation of the identifiers for 8200 vector/motec controllers in connection with a E82ZAFCCxxx function module, system bus (CAN), and the integrated system bus interface of 93XX controllers.

The identifier consists of a basic identifier and the node address.

Addressing to CANopen indices $14XX_{hex}$/$18XX_{hex}$

If the subcode has the value 3, this makes clear that the identifiers have been changed via the CANopen indices $14XX_{hex}$/$18XX_{hex}$. Now, the identifier is developed from the CANopen indices.

A change of the codes L-C1854/L-C2354 does not influence the current identifiers.

L-C1854/L-C2354: Selective addressing CAN-IN/CAN-OUT

Code	Subcode	Index	Possible settings		Data type
			Lenze	Selection	
L-C1854	/1 CAN-IN1 /2 CAN-OUT1 /3* CAN-IN2 /4* CAN-OUT2 /5* CAN-IN3 /6* CAN-OUT3	$58C1_h$ = 22721_d	/1: 129 /2: 1 /3: 257* /4: 258* /5: 385* /6: 386*	0 [1] 1663	FIX32
L-C2354		$56CD_h$ = 22221_d		0 [1] 513	

*) not effective when using 82XX, 8200 vector or 93XX controller

With code L-C1854 it is possible to set the addresses of the input and output PDOs individually via 6 subcodes (compare with previous chapter).

The code becomes effective, if one of the subcodes of the code L-C1853/L-C2353 contains the value 1 (selective addressing).

Note!

Changing the address in one or several subcodes becomes effective

- by renewed mains disconnection of the module or
- by sending a network manager command via CAN bus to the module.
  – *Reset_Node* or L-C2120 = 1 or
  – *Reset_Communication*

L-C1855/L-C2355: Display of resulting identifiers CAN-IN / CAN-OUT

Code	Subcode	Index	Possible settings		Data type
			Lenze	Selection	
L-C1855	/1 CAN-IN1 /2 CAN-OUT1 /3* CAN-IN2	$58C0_h$ = 22720_d	-	0 [1] 2047	FIX32
L-C2355	/4* CAN-OUT2 /5* CAN-IN3 /6* CAN-OUT3	$56CC_h$ = 22220_d			

*) not effective when using 82XX, 8200 vector or 93XX controller

In six subcodes of this code the resulting identifiers of the PDOs can be read out.

Note! (for servo PLC 9300 / Drive PLC)

In case the addressing under L-C2353 or L-C2354 has been changed, the display will only be updated under L-C2355

- by renewed mains disconnection of the module or
- by sending a network manager command via CAN bus to the module.
  – *Reset_Node* or L-C2120 = 1 or
  – *Reset_Communication*

L-C1856/L-2356:
Boot up and cycle times

Code	Subcode	Index	Possible settings		Data type
			Lenze	Selection	
L-C1856	/1 Boot up time /2 cycle times CAN-OUT1 /3 cycle times CAN-OUT2 /4 cycle times CAN-OUT3 /5 Sync-Tx cycle times	$58BF_h$ = 22719_d	1: 3000 ms 2 .. 5: 0 ms	0 [1 ms] 65535	FIX32
L-C2356		$56CB_h$ = 22219_d			FIX32

In the event-controlled/cyclic operation it is possible to define the cycle times with which the single PDOs will be sent via the CAN bus (also see L-C1875/L-C2375, “Tx mode” for CAN-OUT1..3).

Note!

In case of servo PLC 9300 / drive PLC the change only becomes effective by renewed mains disconnection of the fieldbus module or by sending one of the network manager commands “Reset_Node” or “Reset_Communication” via CAN bus to the fieldbus module. It is also possible to carry out the update by assigning value 1 to the AIF control byte.

The change will be immediately valid for the controller 82XX and 93XX.

The value 0 deactivates the cyclic sending of the PDO.

Sync-Tx cycle time (C1856/5 or C2356/5)

“Sync-Tx cycle time” describes the interval time (time basis: ms) that is required for sending a SYNC telegram to the CAN bus.

Note!

Changing the Sync-Tx cycle time gets immediately effective.

The value 0 in code L-C1856/5 or L-C2356/5 deactivates the cyclic sending of the sync telegram.

Relation to CANopen

The CANopen index 1006_h “Communication Cycle Period” reflects the contents of code L-C1856/5 or L-C2356/5 (time basis: ms).

As the data processing speed of the 2175 module amounts to 1000 µs, the entry via CANopen 1006_h is rounded to an integral multiple of 1000 µs. If the CANopen index 1006_h is read, the contents of this code in [µs] will be sent as response.

In addition, the bit 30 of the CANopen index 1005_h (COB-ID Sync message) is automatically set by describing the codes L-C1856/5 or L-C2356/5.

L-C1857/L-C2357: Monitoring time

Code	Subcode	Index	Possible settings		Data type
			Lenze	Selection	
L-C1857	/1 CAN-IN1 /2 CAN-IN2 /3 CAN-IN3 /4 BUS-OFF monitoring time	$58BE_h$ = 22718_d	3000 ms	0 [1 ms] 65535	FIX32
L-C2357		$56CA_h$ = 22218_d			FIX32

Note!

The value 0 deactivates the monitoring.

A change of monitoring times becomes immediately effective for 93XX and 82XX controllers.

With Servo PLC 9300 / Drive PLC the change only becomes effective by renewed mains disconnection of the fieldbus module or by sending one of the network manager commands "Reset_Node" or "Reset_Communication" via CAN bus to the field bus module. It is also possible to carry out the update by assigning value 1 to the AIF control byte.

The monitoring time starts with the arrival of the first telegram.

The monitoring time describes the period of time in which new process input data must arrive with the CAN-IN1..3-identifiers. If the time entered is exceeded, it is possible to set a corresponding reaction under code L-C1882.

BUS-OFF (C1857/4 or C2357/4)

Even if the controller changes to the state BUS-OFF, it is possible to set a time (time basis: ms) in which a reaction can take place.
Monitoring reaction: see code L-C1882 or L-C2382.

L-C1859/L-C2359: Display of DIP switch position

Code	Subcode	Index	Possible settings		Data type
			Lenze	Selection	
L-C1859	-	$58BC_h$ = 22716_d	-	0 [1] 1023	U16
L-C2359	-	$56C8_h$ = 22216_d			U16

The DIP switch position is indicated with the initialisation of the module.

The following table shows the valency:

Switch	---	---	---	---	---	---	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
Valency							512	256	128	64	32	16	8	4	2	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Example for DIP switch position:

- S10 = ON (CANopen communication profile)
- S4, S5 = ON (address 6)
- S7, 8, 9 = OFF (baud rate 500kbits/s)

From the table above the sum of the corresponding valencies amounts to: 61_h (97_d), which is displayed when reading the code L-C1859 or L-C2359.

L-C1860:
Display of the current DIP switch position

Code	Subcode	Index	Possible settings			Data type
			Lenze	Selection		
L-C1860	-	$58BB_h$ = 22715_d	-	0 [1] 1023		U16

By displaying the current DIP switch position it is possible to find out if the switch position for address, baud rate and communication profile setting has changed since the last initialisation. Valency see code L-C1859.

L-C1867/L-C2367:
Sync Rx identifier

Code	Subcode	Index	Possible settings		Data type
			Lenze	Selection	
L-C1867	-	$58B4_h$ = 22708_d	128	0 [1] 2047	FIX32
L-C2367	-	$56C0_h$ = 22208_d			FIX32

The code contains the identifier with which the sync telegram is received.

By receiving the sync telegram it is possible for the module to e.g. send its process data objects to the CAN bus. Also see L-C1875 / L-C2375.

Note!

A change of the identifier will immediately become effective for the 93XX and 82XX controller.

With Servo PLC 9300 / Drive PLC the change only becomes effective by renewed mains disconnection of the fieldbus module or by sending one of the network manager commands “Reset_Node” or “Reset_Communication” via CAN bus to the fieldbus module. It is also possible to carry out the update by assigning value 1 to the AIF control byte.

Relation to CANopen

The CANopen index 1005_h “COB-ID SYNC message” directly influences this code. The identifier of a value re-entered into the index 1005_h will also be taken over from the code L-C1867/L-C2367.

When reading the CANopen index 1005_{hex} (COB-ID Sync message) the value saved under code L-C1868/L-C2368 is displayed.

L-C1868/L-C2368: Sync Tx identifier

Code	Subcode	Index	Possible settings		Data type
			Lenze	Selection	
L-C1868	-	$58B3_h$ = 22707_d	128	0 [1] 2047	FIX32
L-C2368	-	$56BF_h$= 22207_d			FIX32

The code contains the identifier with which the sync telegram is received.

With the identifier set under L-C1868/L-C2368 the sync is to the CAN bus (concerning this also see L-C1856 or L-C2356, subcode 5).

Note!

Changing the identifier will immediately get effective for the 93XX and 82XX controller.

In case of servo PLC 9300 / drive PLC the change only becomes effective by renewed mains disconnection of the field bus module or by sending one of the network manager commands “Reset_Node” or “Reset_Communication” via CAN bus to the field bus module. It is also possible to carry out the update by assigning the AIF control byte with the value 1.

Relation to CANopen

The CANopen index 1005_h “COB-ID SYNC message” directly influences this code. The identifier of a value that has been reentered into the index 1005_h will also be taken over from the code L-C1868/L-C2368. When reading the index 1005_h the value saved here will be displayed.

L-C1873/L-C2373: Sync rate CAN-IN1 ... CAN-IN3

The input process data (CAN-INx) are only transferred to the controller after a certain number of SYNC telegrams have been received.

Code	Subcode	Index	Possible settings		Data type
			Lenze	Selection	
L-C1873	/1 CAN-IN1 /2* CAN-IN2	$58AE_h$ = 22702_d	1	0 [1] 240	FIX32
L-C2373	/3* CAN-IN3	$56BA_h$ = 22202_d		1 [1] 240	FIX32

*) not effective when using 82XX, 8200 vector or 93XX controller

Example:

Selection n = 23. Acceptance of input PDO (to CAN-IN1... CAN-IN3) into the controller after the arrival of the 23. Sync telegram.

The sync rate can be set individually for each input PDO.

Note!

Only sync telegrams are counted, that have been received with the identifier set under L-C1867 / L-C2367 of the 2175 module.

A change of the sync rate will immediately become effective for the 82XX and 93XX controller.

With Servo PLC 9300 / Drive PLC the change only becomes effective by renewed mains disconnection of the fieldbus module or by sending one of the network manager commands "Reset_Node" or "Reset_Communication" via CAN bus to the fieldbus module. It is also possible to carry out the update by assigning value 1 to the AIF control byte.

Relation to CANopen

Die CANopen indices 1400_h, 1401_h and 1402_h (receive PDO communication parameter), subindex 2 each (transmission type), are directly mapped on the subcodes of code L-C1873 / L-C2373.

Index $14XX_h$, subindex 2 = 1-240	=	Code L-C1873, subcode X = 1-240
Index $14XX_h$, subindex 2 = 254		Code L-C1873, subcode 0

An exception is the value 0, which is not directly mapped on the CANopen indices 1400_{hex}, 1401_{hex} and 1402_{hex}. The value 0 is mapped under CANopen index $14Xx_{hex}$, subindex 2, with the value = 254 (vendor-specific). On the other hand, with the entry of 254 in subindex 2 the corresponding subcode of code L-C1873 is described with the value 0.

L-C1874/L-C2374: Sync rate CAN-OUT1 ... CAN-OUT3

Code	Subcode	Index	Possible settings		Data type
			Lenze	Selection	
L-C1874	/1 = CAN-OUT1 /2* = CAN-OUT2 /3* = CAN-OUT3	$58AD_h$ = 22701_d	1	1 [1] 240	FIX32
L-C2374		$56B9_h$ = 22201_d			FIX32

*) not effective when using 82XX, 8200 vector or 93XX controller

The output process data (CAN-OUTx) are only transferred after a certain number of SYNC telegrams have been received.

The sync rate can be set individually for each process output data object.

Note!

Only those sync telegrams are counted, that have been received with the identifier set under L-C1867 / L-C2367 of the 2175 module.

A change of the sync rate will immediately become effective for the 82XX and 93XX controller.

With Servo PLC 9300 / Drive PLC the change only becomes effective by renewed mains disconnection of the fieldbus module or by sending one of the network manager commands "Reset_Node" or "Reset_Communication" via CAN bus to the fieldbus module. It is also possible to carry out the update by assigning the AIF control byte with the value 1.

Relation to CANopen

The CANopen indices "Transmit PDO communication parameter" 1800_h, 1801_h and 1802_h (with subindex 2, "transmission type") are directly mapped on the subcodes of code L-C1874 or L-C2374.

Index $18XX_h$, subindex 2 = 1-240	=	Code L-C1874 or L-C2374, subcode X = 1-240 Code L-C1875, subcode X = 0
Index $18XX_h$, subindex 2 =252		Code L-C1875, subcode X = 1
Index $18XX_h$, subindex 2 = 254		Code L-C1875, subcode X = 2

If for instance the index 1802_h is read, the value in code L-C1874 / L-C2374, subcode 3 comes back as an answer. If the index 1801_h is described with a new value, the code L-C1874, subcode 2, is described with this value, too.

L-C1875/L-C2375: Tx mode CAN-OUT1 ... CAN-OUT3

This code contains a selection which indicates at which event the output PDOs CAN-OUT1 .. CAN-OUT3 are to be sent. It is possible to make an individual selection for each output PDO by dividing the code in subcodes.

Code	Subcode	Index	Possible settings		Data type
			Lenze	Selection	
L-C1875	/1 CAN-OUT1 /2* CAN-OUT2 /3* CAN-OUT3	$58AC_h$ = 22700_d	/1: 0 /2: 1 /3: 1	0 [1] 3	FIX32
L-C2375		$56B8_h$ = 22200_d	/1: 0 /2: 0 /3: 0		FIX32

*) not effective when using 82XX, 8200 vector or 93XX controller

- Value = 0
  - The output PDOs are sent when a sync telegram has been received.
  - Only those sync telegrams are counted, that have been received with the identifier set under L-C1867 / L-C2367. It is possible to set that the output PDOs are only to be sent after the nth sync telegram (adjustabel with n = 1..240) under L-C1874 / L-C2374.
- Value
  - Output PDOs are not sent.
  - With this selection the transmission of CAN-OUT1..CAN-OUT3 is deactivated.
    This is useful for 82XX, 8200 vector and 9300 drives, which are only able to exchange 4 words as a maximum of control and status information via the AIF. For this purpose the use of an input and output PDO is sufficient, as each PDO contains 4 words of information (factory setting).
    In this case an unnecessary load of the CAN can be avoided.
- Value = 2
  - The PDO is either transmitted event-controlled or cyclically if this value is entered in one of the three available subcodes.
  - The output PDO is transmitted cyclically, if a cycle time is defined for the same CAN-OUT1..3 under code L-C1856 or (dependent on drive) L-C2356. If the cycle time is zero, this is sent in case of an event, i. e. bit change within the CAN-OUT object.
- Value = 3
  - The output PDO is event-controlled and cyclically transmitted. This means that the object is transmitted with the cycle time defined under code L-C1856 or L-C2356.
  - In addition, the object will be transmitted if one or several bits are changed within the CAN-OUT object.

Note!

A change of the Tx mode will immediately become effective for the 82XX and 93XX controller for a process data output object.

With Servo PLC 9300 / Drive PLC the update must take place by assigning value 1 to the AIF control byte.

If the transmission is event-controlled (also with cyclic superpositions) some bits can be hidden by masking the object using codes L-C1876 to L-C1878 or with the Servo PLC 9300 / Drive PLC L-C2376 up to L-C2378. That means that the CAN-OUT object will **not** be sent when a bit is being changed.

If the value under code L-C1875/L-2375 = 0 , subcodes 1..3, the value of code L-C1874/L-C2374 is reflected under CANopen index 1800_{hex}, 1801_{hex} or 1802_{hex}, subindex 2 each.

The value = 1 under code L-C1875/L-C2375 is displayed under CANopen index 1800_{hex}, 1801_{hex} or 1802_{hex} with the value = 252.

The value = 2 or 3 under code L-C1875/L-C2375 is displayed under CANopen index 1800_{hex}, 1801_{hex} or 1802_{hex} with the value = 254.

When describing CANopen index 1800_{hex}, 1801_{hex} or 1802_{hex} the same relation prevails, see the table below or under the description of the CANopen indices $18XX_{hex}$.

Code L-C1875 / L-C2375, subcode 1, 2 or 3	CANopen index 1800hex, 1801hex or 1802hex, subindex 2
0	Code L-C1874 / L-C2374, subcode 1, 2 or 3 (Wert = 1 .... 240)
1	252
2 or 3	254

L-C1876/L-C2376: Masks CAN-OUT1

One or several bits of the output PDO CAN-OUT1 can be extracted by the mask.

Code	Subcode	Index	Possible settings		Data type
			Lenze	Selection	
L-C1876	/1 CAN-OUT1.W1 /2 CAN-OUT1.W2	$58AB_h$ = 22699_d	65535	0 [1] 65535	FIX32
L-C2376	/3 CAN-OUT1.W3 /4 CAN-OUT1.W4	$56B7_h$ = 22199_d			FIX32

The event-controlled transmission of the CAN-OUT object can be e.g. dependent on only one bit. Also see code L-C1875 / L-C2375.

Example:

The mask in word 3 of the process data object CAN-OUT 1 is set through code L-C1876/3 with the value 20_{hex} (see “MASK“). Please note the field marked in grey.

1. cycle

Result after 1. cycle: The PDO is transmitted

	CAN-OUT 1.W3															
MASK	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Data	1	1	1	1	1	1	1	1	0	1	1	0	0	0	1	0
Result	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**

2. cycle

After the 2. cycle new data have been written into CAN-OUT 1.

Result after 2. cycle: The PDO is not transmitted due to bit change

	CAN-OUT 1.W3															
MASK	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Data	1	1	1	1	1	1	1	1	0	1	0	0	0	0	1	0
Result	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Note!

A change of the mask will immediately become effective for the 82XX and 93XX controller.

With Servo PLC 9300 / Drive PLC the change only becomes effective by renewed mains disconnection of the fieldbus module or by sending one of the network manager commands “Reset_Node” or “Reset_Communication” via CAN bus to the fieldbus module. It is also possible to carry out the update by assigning value 1 to the AIF control byte.

L-C1877/L-C2377: Masks CAN-OUT2

One or several bits of the output PDO CAN-OUT2 can be extracted by the mask.

Code	Subcode	Index	Possible settings				Data type
			Lenze	Selection			
L-C1877	/1 CAN-OUT2.W1 /2 CAN-OUT2.W2	$58AA_h$ = 22698_d	65535	0	[1]	65535	FIX32
L-C2377	/3 CAN-OUT2.W3 /4 CAN-OUT2.W4	$56B6_h$ = 22198_d					

See L-C1876 / L-C2376.

L-C1878/L-C2378: Masks CAN-OUT3

One or several bits of the output PDO CAN-OUT3 can be extracted by the mask.

Code	Subcode	Index	Possible settings				Data type
			Lenze	Selection			
L-C1878	/1 CAN-OUT3.W1 /2 CAN-OUT3.W2	$58A9_h$ = 22697_d	65535	0	[1]	65535	FIX32
L-C2378	/3 CAN-OUT3.W3 /4 CAN-OUT3.W4	$56B5_h$ = 22197_d					

See L-C1876 / L-C2376.

L-C1882/L-C2382: Monitoring reaction

Code	Subcode	Index	Possible settings				Data type
			Lenze	Selection			
L-C1882	/1 response CAN-IN1 /2 response CAN-IN2 /3 response CAN-IN3 /4 response BUS-OFF /5 response Life Guarding Event	$58A5_h$ = 22693_d	0	0 0: no response 1: Controller inhibit 2: Quickstop	[1]	2	FIX32
L-C2382		$56B1_h$ = 22193_d					FIX32

Defintion of the reaction, when the monitoring times (see L-C1857/L-C2357) have been exceeded.

Note!

A change of the monitoring response will immediately become effective for the 82XX and 93XX controllers.

With Servo PLC 9300 / Drive PLC the change only becomes effective by renewed mains disconnection of the fieldbus module or by sending one of the network manager commands "Reset_Node" or "Reset_Communication" via CAN bus to the fieldbus module. It is also possible to carry out the update by assigning value 1 to the AIF control byte.

Relation to CANopen

Under the CANopen indices $100C_h$ "guard time" and $100D_h$ "life time factor" it is possible to set a time for the node guarding protocol. The node guarding protocol has been developed in order to monitor the connection of master and slave (in this case 2175IB). Under CANopen index $100C_h$ "guard time" it is possible to enter a time in milliseconds. Under index $100D_h$ "life time factor" a factor is stored. The product of both indices results in a monitoring time in which the master must send the slave 2175IB a specific telegram. If one of the two indices is set to zero, the monitoring time is also zero and thus deactivated. The slave transmits a telegram with its current NMT state to the master. These states can be pre-operational, operational or stopped.

If the monitoring time is exceeded, the slave reacts with the life guarding event and the master with the node guarding event.

The sequence of the node guarding protocol must be programmed and started in the master. The 2175IB module supports the node guarding protocol, it is only possible to enter a response under this code.

L-C2120: AIF control byte

Code	Subcode	Index	Possible settings		Data type
			Lenze	Selection	
L-C2120	-	22455_d = $57B7_h$	0	0 = No command 1 = Update codes L-23XX and CAN re-initialisation ≡ Reset node 2 = Update codes L-C23XX 10 = L-C2356/1...4 re-read 11 = L-C2357 re-read 12 = L-C2375 re-read 13 = L-C2376 ... L-C2378 re-read 14 = L-C2382 re-read	FIX32

Wit the AIF control byte it is possible to read the codes L-C23XX saved in the servo PLC 9300 / Drive PLC into the 2175 field bus module. By writing a value indicated in the table into the AIF control byte this process can be initiated per command.

By writing the value = 2 into the AIF control byte all L-C23XX codes are re-read. For some codes however it is necessary to carry out a CAN re-initialisation so that new values or the functions derived from them will become effective. Here is a list:

Code	Value L-C2120	Function
L-C2350	1	Activation of the new identifiers for SDO, PDO or Emergency, if these are dependent on the node address
L-C2351	1	Activation of new Baud rate
L-C2352	1	Activation of changed function (master/slave)
L-C2353	1	Activation of new addressing
L-C2354	1	Activation of new addressing (selective)
L-C2355	1	Representation of the new identifiers
L-C2356, subcode 1..4	1, 2 or 10	Activation of new times (boot-up, cycle AIF-XCAN1..3)
L-C2356, subcode 5	1 or 2	Activation of sync cycle time
L-C2357, subcode 1..4	1, 2 or 11	Activation of monitoring times
L-C2359	1	Representation of changed switch position
L-C2367	1	Activation of new receiving identifiers for sync telegrams
L-C2368	1	Activation of new transmitting identifiers for sync telegrams
L-C2373	1 or 2	Activation of new sync rate receiving POD
L-C2374	1 or 2	Activation of new sync rate transmitting PDO
L-C2375	1, 2 or 12	Activation of new mode transmitting PDO
L-C2376 to L-C2378	1, 2 or 13	Activation of new mask transmitting PDO
L-C2382	1, 2 or 14	Activation of new fault response PDO, bus OFF and life guarding event

L-C2121: AIF status byte

Code	Subcode	Index	Possible settings				Data type
			Lenze	Selection			
L-C2121	-	22454_d = $57B6_h$	0	0	1	255	FIX32

The AIF status byte provides information of the 2175 fieldbus module of the Servo PLC 9300 and Drive PLC. By reading the status bit the Servo PLC 9300 and Drive PLC can monitor the status of the 2175 fieldbus module. Depending on this it is possible for the user to take corresponding countermeasures.

AIF status byte	Description
Bit 0	CE11 fault, monitoring time CAN-IN1 exceeded
Bit 1	CE12 fault, monitoring time CAN-IN2 exceeded
Bit 2	CE13 fault, monitoring time CAN-IN3 exceeded
Bit 3	CE14 fault, module in BUS-OFF state
Bit 4	Operational state
Bit 5	Pre-Operational state
Bit 6	Warning state
Bit 7	internally assigned

7.7.3 Implemented CANopen objects

Lenze devices can be parameterised either with Lenze codes (see (📖 7.9-1)) or with the vendor-independent "CANopen objects". In order to achieve a completely CANopen-conform communication, only the CANopen objects are allowed to be used for parameter setting. The CANopen objects described in these Instructions are defined in the "CiA Draft Standard 301/Version 4.01".

All CANopen objects can be mapped on Lenze codes. In section "**Relation to CANopen**" the effects on Lenze codes through changing the CANopen objects are described.

Note!

Some of the terms used have their origin in the CANopen protocol which is written in English. The translation of these terms is only partly allowed.

1000_{hex}: Device type

The CANopen index 1000_{hex} describes the profile for this device. Furthermore, it is possible to enter additional information, being defined in the device itself. If no special device profile is considered, the contents are 0000_{hex} (2175IB).

Index [hex]	Subindex	Name	Data type	Value range	Rights
1000	0	Device type	U32	0 ... (2^{32} - 1)	ro

Bit assignment in the telegram data

5th byte	6th byte	7th byte	8th byte
U32			
LSB Device profile number		Additional information MSB	

Reading the fault register

1001_{hex}: Error Register

Index [hex]	Subindex	Name	Data type	Value range	Rights
1001	0	Error register	U8	0 ... 255	ro

Bit assignment in the data byte (U8) of the telegram

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	0	0	0: no TRIP 1: device TRIP/ fault CANopen module Fault analysis: see L-C0161 in the Instructions of the corresponding controller

Fault history

1003_{hex}: Pre-defined Error Field

Index [hex]	Subindex	Name	Data type	Value range	Rights
1003	0	Recording numbers of error	U8	0 ... 255	rw
	1	Standard error field	U32	0 .... (2^{32} - 1)	r0

This object serves to detect faults in the module and the basic device:

- Subindex 0: Number of the stored fault messages.
- Subindex 1: Display of the fault list.
  The fault messages (U32) consist of a 16-bit error code and a vendor-specific information field comprising 16 bits.

Note!

The value in "Standard error field" under subindex 1 is deleted, if subindex "recording number of errors" is 0.

1005_{hex}:
Identifier sync message

With this object it is possible

- to create sync telegrams for the module
- to describe the value of the identifier.

Index [hex]	Subindex	Name	Data type	Value range	Rights
1005	0	Identifier sync message	U32	0 .... (2^{32} - 1)	rw

Sync telegram creation

For creating sync telegrams, bit 30 (see below) must be set to the value 1.

The intervals of the sync telegrams can react to another object (index 1006_{hex}).

Identifier description

For the receipt of PDOs the value 80_{hex} is entered as default setting (and in accordance with CANopen specification) in the 11 bit identifier. This means, that all modules are preset to the same sync telegram.
If sync telegrams are only to be received from specific modules, the corresponding identifiers can be entered with a value up to $7FF_{hex}$. The identifier may only be changed, if the 2175 fieldbus module does not send any sync (bit 30 = 0).

5th byte	6th byte	7th byte	8th byte
U32			

0 … 10	11 - 28																		29	30	31
11-bit identifier	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	X

Bit no.	Value	Explanation
0 - 10	X	Identifier (also see chapter 36)
(11 - 28)*	0	*) The extended identifier (29 bits) is not supported. Each bit in this area must have the value 0.
29*	0	
30	0 1	The device does not create any SYNC telegrams The device creates SYNC telegrams
31	X	optional

1006_{hex}:
Communication Cycle Period

Cycle time setting of sync telegrams

Index [hex]	Subindex	Name	Data type	Value range	Rights
1006	0	Communication cycle period	U32	0 ... (2^{32} - 1)	rw

With the preset value (default) of t = 0 no sync telegrams are created.

The cycle time can be selected with the entry 1000 or the integral multiple of this figure. The unit of the entered time is [μs]. The maximum value to be set of the 2175 fieldbus module is 65535000 [μs].

1008_{hex}:
Manufacturer Device Name

Announcement of the controller and module names

Index [hex]	Subindex	Name	Data type	Rights
1008	0	Manufacturer device name	Visible string {9 digits}	const

$100A_{hex}$:
Manufacturer software version

Software version of the controller and module

Index [hex]	Subindex	Name	Data type	Rights
100A	0	Manufacturer software version	Vis. string {11 digits}	const

$100C_{hex}$:
Guard Time

Monitoring time

Index [hex]	Subindex	Name	Data type	Value range	Rights
100C	0	Guard time	U16	0 ... 65535	rw

The monitoring time is indicated in [ms].

If the monitoring shall not be supported, the default entry of 0 is to be maintained.

100D$_{hex}$: Life Time Factor

Index [hex]	Subindex	Name	Data type	Value range	Rights
100D	0	Life time factor	U8	0 ... 255	rw

If the monitoring shall not be supported, the default entry of 0 is to be maintained.

1010$_{hex}$: Store Parameters

Storage of parameters in the EEPROM.

Index [hex]	Subindex	Name	Data type	Value range	Rights
1010	0	store parameters	U32	0 ... (2^{32} - 1)	ro/rw
	1 ... 3*				

The subindices 1, 2 and 3 are not supported at the moment

Store parameters

Fault message in case of

- faulty storing (in the bytes 5 ...8): 0606 0000$_{hex}$
- false signature: 0800 0020$_{hex}$

Note!

For storing module parameters the signature “**save**” must be included in the telegram data.

Assignment of the telegram data words to store parameters

Signature	MSB			LSB
ISO 8859 (ASCII)	**e**	**v**	**a**	**s**
hex	65	76	61	73

Bit assignment for write authorisation

U32								
0	1	2 - 31						
0/1	0/1	0	0	0	..	0	0	0

<table>
<tr><th rowspan="2">Subindex</th><th rowspan="2">Rights</th><th colspan="3">Explanation</th></tr>
<tr><th>Writing</th><th colspan="2">Reading</th></tr>
<tr><td>0</td><td>ro</td><td>• If you attempt to write the following fault message occurs: 0601 0002</td><td colspan="2">Supported subindex = 3</td></tr>
<tr><td>1</td><td rowspan="3">rw</td><td rowspan="3">• This function is not yet supported at the moment.
• If you attempt to write the following fault message occurs: 0800 0020</td><td>Read memory functionality of all parameters</td><td rowspan="3">The following functionalities are possible in dependence on the controller and are described by reading the values of the bit positions 0 and 1:
Value 0: Storing is not carried out
Value 1: Storing on command
Value 2: Automatic storing
Value 3: Automatic storing and storing on command</td></tr>
<tr><td>2</td><td>Only read memory functionality of the communication parameters of the objects.</td></tr>
<tr><td>3</td><td>Only read memory functionality of the vendor-specific parameters

(Memory area: 6000$_{hex}$ - 9FFF$_{hex}$)</td></tr>
</table>

1011_{hex}: Restore Default Parameters

Loading of the default setting.

> **Note!**
> If this function is applied, the subindices used depend on the controller type.

Index [hex]	Subindex	Name	Data type	Value range	Rights
1011	0 ... 7	restore default parameters	U32	0 ... (2^{32} - 1)	rw/ro

Besides index and subindex the signature "**load**" must be included in the telegram data, so that the parameters can be loaded (see table).

Signature	MSB			LSB
ISO 8859 (ASCII)	**d**	**a**	**o**	**l**
hex	64	61	6F	6C

Bit assignment for write authorisation

U32						
0	1 - 31					
0: Loading not possible 1: Loading possible	0	0	0 ..	0	0	0

Subindex	Rights	Writing	Reading
0	ro	• If you attempt to write the following fault message occurs: 0601 0002_{hex}	Maximally available subindex dependent on the controller type: 7: 8200 vector/motec frequency inverter
1	rw	• This function is not yet supported at the moment. • If you attempt to write the following fault message occurs: 0800 0020_{hex}	Loading of all parameters is possible
2			Only loading of the communication parameters of the objects is possible
3			Only loading of the vendor-specific parameters (index 6000_{hex} - $9FFF_{hex}$)
4			Loading of parameter set 1 possible
5			Loading of parameter set 2 possible
6			Loading of parameter set 3 possible
7			Loading of parameter set 4 possible

1014_{hex}:
COB-ID emergency object

If an internal fault of the bus module or the controller occurs or is accepted (e.g. TRIP), a fault message is sent via the CAN bus. The telegram is transmitted once at each fault.

Index [hex]	Subindex	Name	Data type	Value range	Rights
1014	0	COB-ID emergency object	U32	0 ... (2^{32} - 1)	rw

Assignment of the data telegram

5th byte	6th byte	7th byte	8th byte
U32			

0 10	11 - 28																		29	30	31
11-bit identifier	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1

Explanation

Bit no.	Value	Explanation
0 - 10	0/1	Identifier (also see chapter 36)
(11 - 28)*	0	*) The extended identifier (29 bits) is not supported. Each bit in this area must have the value 0.
29*	0	
30	0	Reserved
31	0/1	0: valid emergency object 1: invalid emergency object

The “Emergency” telegram sent via CAN bus is structured as follows:

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
Emergency error code LENZE: Error code “10XX”		Error register Object 1001_{hex}	Field for vendor-specific fault message				

1015_{hex}:
Inhibit time emergency

This object serves to select the interval between the occurence of an internal fault and the sending of the fault message to the bus (“COB-ID Emergency object”, code: 1014_{hex}).

Only integral multiples of 10 are processed as entered values. The entered value multiplied by 100 results in the period of time in [μs].

Index [hex]	Subindex	Name	Data type	Value range	Rights
1015	0	Inhibit time emergency	U16	0 ... 65535	rw

1018_{hex}:
Identity object

Entry of the vendor ID

Index [hex]	Subindex	Name	Data type	Rights
1018	0 ... 4	Identity object	Identity	ro

The identification number for Lenze allocated by the “Organisation CAN in Automation e. V.” can be read out via this object:

Subindex	Meaning
0	Highest subindex
1	Vendor ID
2	Product code
3	Version number
4	Serial number

1200_{hex} / 1201_{hex}:
Server SDO parameters

Two objects (CAN parameter data channel 1 = 1200_{hex} and CAN parameter data channel 2 = 1201_{hex}) are available for parameter setting of the servers SDOs.

With index 1201 the identifier can be described in receiving direction and sending direction while index 1200 only possesses read rights. The server SDO parameter is only valid when the bit 31 contains the value 0 in both transmission directions (subindes 1 and 2).

Index [hex]	Subindex	Name	Data type		Rights	Explanation of subindex
1200	0	Server SDO Parameter	SDO	U 8	ro	0: Max. supported subindex = 2 1: Specification of receiving identifier 2: Specification of sending identifier
	1	Identifier client → Server (rx)	Param	U 32	ro	
	2	Identifier server → Client (tx)		U 32	ro	
1201	0	Server SDO Parameter		U 8	ro	
	1	Identifier client → Server (rx)		U 32	rw	
	2	Identifier server → Client (tx)		U 32	rw	

Assignment of the data telegram

5th byte	6th byte	7th byte	8th byte
U32			

0 … 10	11 - 28																		29	30	31
11-bit identifier	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1

Explanation

Bit no.	Value	Explanation
0 - 10	0/1	Identifier (also see (🕮 7.6-3))
(11 - 28)*	0	*) The extended identifier (29 bits) is not supported. Each bit in this area must have the value 0.
29*	0	
30	0	Reserved
31	0/1	0: SDO valid 1: SDO invalid

Example:

The CAN parameter data channel 2 of the drive with the controller address 4 is to be switched off.

This command must be sent from the master to the drive via the parameter data channel 1 (SDO1). The table on (🕮 7.6-3) includes the basic identifier of the SDO1 with 1536_{dec}.

(resulting) identifier = basic identifier + controller address = 1540_{dec} = 0604_{hex}

In order to switch off the parameter data channel (= invalid), the bit 31 must be set to “1”. The value results in 80 00 00 00_{hex} + 604_{hex} = 80 00 06 04_{hex} (see above).

Byte:	1	2	3	4	5	6	7	8
Name:	Command	Index (Low byte)	Index (High byte)	Subindex	U 32			
Read request:	23 (Write request)	01	12	01 Client → Server (rx)	04_{hex}	06_{hex}	0000 0000	80_{hex}

1400_{hex}: Receive PDO1 communication parameter

Receipt of communication parameters of PDO 1

Index [hex]	Subindex	Name	Data type		Rights	Explanation
1400	0	Number of entries	PDO comm.	U 8	ro	Max. supported subindex = 2
	1	COB-ID used by PDO		U 32	rw	Setting of the identifier for this PDO (200_{hex} + node ID) 🕮 7.6-3
	2	Transmission type		U 8	rw	Setting of the transmission type (see table)

Assignment of the data telegram

U32																				
0 ... 10	11 - 28																	29	30	31
11-bit identifier	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1

Explanation subindex 1:

Bit no.	Value	Meaning
0 - 10 (LSB)	X	contains the identifier (basic + controller address)
(11 - 28)*	0	*) The extended identifier (29 bits) is not supported. Each single bit must adopt the state 0!
29*	0	*) The extended identifier (29 bits) is not supported. Each single bit must adopt the state 0!
30	0 1	RTR to this PDO is permitted (Lenze) RTR to this PDO is not permitted (not adjustable) RTR = remote transmission request
31 (MSB)	0 1	0: PDO active 1: PDO not active

Explanation subindex 2:

PDO transmission			Transmission type	Explanation
cyclic	synchronous	event-controlled		
X	X		n = 1 ... 240	If a value n is entered, this PDO is accepted at every nth SYNC.
		X	n = 254	PDO is immediately accepted, see L-C1873 / L-C2373

Note!

The value n = 254 will only be supported as of version 1.0 of the field bus module.

1401_{hex}: Receive PDO2* Communication Parameter

Note!

*) The object is not available for 82XX, 8200 vector/motec and 93XX controllers.

Receipt of communication parameters of PDO 2

Index [hex]	Subindex	Name	Data type	Value range	Rights	Explanation
1401	0	Number of entries	PDO comm.	U 8	ro	Max. supported subindex = 2
	1	COB-ID used by PDO		U 32	rw	Setting of the identifier for this PDO (300_{hex} + node ID) 🕮 7.6-3
	2	Transmission type		U 8	rw	Setting of the transmission type (see table)

For further description regarding this object please see the description for 1400_{hex}.

1402_{hex}: Receive PDO3* Communication Parameter

Receipt of communication parameters of PDO 3

Note!

*) The object is not available for the 82XX, 8200 vector/motec and 93XX controllers.

Index [hex]	Subindex	Name	Data type	Value range	Rights	Explanation
1402	0	Number of entries	PDO comm.	U 8	ro	Max. supported subindex = 2
	1	COB-ID used by PDO		U 32	rw	Setting of the identifier for this PDO (400_{hex} + node ID) 🕮 7.6-3
	2	Transmission type		U 8	rw	Setting of the transmission type (see table)

For further description regarding this object please see the description for 1400_{hex}.

1600_{hex}:
Receive PDO1 Mapping Parameter

This object serves to receive parameter data as PDO1.

Note!

The functionality of this object is not yet available at the moment. On the attempt to read the object, the following values are sent back in dependence of the subindices:

Subind. 0: value 4_{hex}

Subind. 1 - 4: value 10_{hex}

Index [hex]	Subindex	Name	Data type	Value range	Rights	Explanation
1600	0	Number of mapped objects in PDO's	PDO Mapping	U 8	ro	Subindex 0: Max. supported subindex = 4
	1	PDO mapping 1		U 32		In case of read requests regarding this object the value 10_{hex} will be sent back.
	2	PDO mapping 2		U 32		
	3	PDO mapping 3		U 32		
	4	PDO mapping 4		U 32		

1601_{hex}:
Receive PDO2* mapping parameter

This object serves to receive parameter data as PDO2.

Note!

The functionality of this object is not yet available at the moment. On the attempt to read the object, the following values are sent back in dependence of the subindices:

Subind. 0: value 4_{hex}

Subind. 1 - 4: value 10_{hex}

*) The object is not available for the 82XX, 8200 vector/motec and 93XX controllers.

Index [hex]	Subindex	Name	Data type	Value range	Rights	Explanation
1601	0	Number of mapped objects in PDO's	PDO Mapping	U 8	ro	Subindex 0: Max. supported subindex = 4
	1	PDO mapping 1		U 32		In case of read requests regarding this object the value 10_{hex} will be sent back.
	2	PDO mapping 2		U 32		
	3	PDO mapping 3		U 32		
	4	PDO mapping 4		U 32		

1602$_{hex}$:
Receive PDO3* mapping parameter

This object serves to receive parameter data as PDO3.

Note!

The functionality of this object is not yet available at the moment. On the attempt to read the object, the following values are sent back in dependence of the subindices:

Subind. 0: value 4$_{hex}$

Subind. 1 - 4: value 10$_{hex}$

*) The object is not available for the 82XX, 8200 vector/motec and 93XX controllers.

Index [hex]	Subindex	Name	Data type		Rights	Explanation
1602	0	Number of mapped objects in PDO's	PDO Mapping	U 8	ro	Subindex 0: Max. supported subindex = 4
	1	PDO mapping 1		U 32		In case of read requests regarding this object the value 10$_{hex}$ will be sent back.
	2	PDO mapping 2		U 32		
	3	PDO mapping 3		U 32		
	4	PDO mapping 4		U 32		

1800_{hex}:
Transmit PDO1 Communication Parameter

Process data transmission

Index [hex]	Subindex	Name	Data type		Rights	Explanation
1800	0	Number of subindices supported	PDO Comm.	U 8	ro	Max. supported subindex = 2
	1	Identifier of the PDO		U 32	rw	Setting of the identifier for this PDO (180_{hex} + node ID)
	2	Transmission type		U 8	rw	Setting of the transmission type (see table)

Explanation subindex 1:

Bit no.	Value	Meaning
0 - 10 (LSB)	X	contains the identifier (basic + controller address)
(11 - 28)*	0	*) The extended identifier (29 bits) is not supported. Each single bit must adopt the state 0!
29*	0	*) The extended identifier (29 bits) is not supported. Each single bit must adopt the state 0!
30	0 1	RTR to this PDO is permitted (Lenze) RTR to this PDO is not permitted (not adjustable) RTR = remote transmission request
31 (MSB)	0 1	0: PDO active 1: PDO not active

Explanation subindex 2:

PDO transmission			Transmission type	Explanation
cyclic	synchronous	event-controlled		
x	X		n = 1 ... 240	If a value n is entered, this PDO is accepted at every nth SYNC.
	X		n = 252	PDO is filled with new data but only transmitted to RTR.
		X	n = 254	vendor-specific, see L-C1875 / L-C2375

Note!

The value n = 252 will only be supported as of version 1.0 of the fieldbus module.

1801_{hex}:
Transmit PDO2* Communication Parameter

Process data transmission

Note!

*) The object is not available for the 82XX, 8200 vector/motec and 93XX controllers.

Index [hex]	Subindex	Name	Data type		Rights	Explanation
1801	0	Number of subindices supported	PDO Comm.	U 8	ro	Max. supported subindex = 2
	1	Identifier of the PDO		U 32	rw	Setting of the identifier for this PDO (280_{hex} + node ID)
	2	Transmission type		U 8	rw	Setting of the transmission type (see table)

Explanation subindex 1:

Bit no.	Value	Meaning
0 - 10 (LSB)	X	contains the identifier (basic + controller address)
(11 - 28)*	0	*) The extended identifier (29 bits) is not supported. Each single bit must adopt the state 0!
29*	0	*) The extended identifier (29 bits) is not supported. Each single bit must adopt the state 0!
30	0 1	RTR to this PDO is permitted (Lenze) RTR to this PDO is not permitted (not adjustable) RTR = remote transmission request
31 (MSB)	0 1	0: PDO active 1: PDO not active

Explanation subindex 2:

PDO transmission			Transmission type	Explanation
cyclic	synchronous	event-controlled		
X	X		n = 1 ... 240	If a value n is entered, this PDO is accepted at every nth SYNC.
	X		n = 252	PDO is filled with new data but only transmitted to RTR.
		X	n = 254	vendor-specific, see L-C1875 / L-C2375

1802$_{hex}$:
Transmit PDO3* Communication Parameter

Process data transmission

Note!

*) The object is not available for the 82XX, 8200 vector/motec and 93XX controllers.

Index [hex]	Subindex	Name	Data type		Rights	Explanation
1802	0	Number of subindices supported	PDO Comm.	U 8	ro	Max. supported subindex = 2
	1	Identifier of the PDO		U 32	rw	Setting of the identifier for this PDO (380$_{hex}$ + node ID)
	2	Transmission type		U 8	rw	Setting of the transmission type (see table)

Explanation subindex 1:

Bit no.	Value	Meaning
0 - 10 (LSB)	X	contains the identifier (basic + controller address)
(11 - 28)*	0	*) The extended identifier (29 bits) is not supported. Each single bit must adopt the state 0!
29*	0	*) The extended identifier (29 bits) is not supported. Each single bit must adopt the state 0!
30	0 1	RTR to this PDO is permitted (Lenze) RTR to this PDO is not permitted (not adjustable) RTR = remote transmission request
31 (MSB)	0 1	0: PDO active 1: PDO not active

Explanation subindex 2:

PDO transmission			Transmission type	Explanation
cyclic	synchronous	event-controlled		
X	X		n = 1 ... 240	If a value n is entered, this PDO is accepted at every nth SYNC.
	X		n = 252	PDO is filled with new data but only transmitted to RTR.
		X	n = 254	Vendor-specific, see L-C1875 / L-C2375

1A00$_{hex}$: Transmit PDO1 Mapping Parameter

This object serves to send parameter data as PDO1.

Note!

The functionality of this object is not yet available at the moment. On the attempt to read the object, the following values are sent back in dependence of the subindices:

Subind. 0: value 4_{hex} (max. supported subindex)

Subind. 1 - 4: In case of read requests regarding this object the value 10_{hex} will be sent back.

Index [hex]	Subindex	Name	Data type		Rights
1A00	0	Number of mapped objects in PDO's	PDO Mapping	U 8	ro
	1	PDO mapping 1		U 32	
	2	PDO mapping 2		U 32	
	3	PDO mapping 3		U 32	
	4	PDO mapping 4		U 32	

1A01$_{hex}$: Transmit PDO2* mapping parameter

This object serves to send parameter data as PDO2.

Note!

The functionality of this object is not yet available at the moment. On the attempt to read the object, the following values are sent back in dependence of the subindices:

Subind. 0: value 4_{hex} (max. supported subindex)

Subind. 1 - 4: In case of read requests regarding this object the value 10_{hex} will be sent back.

*) The object is not available for the 82XX, 8200 vector/motec and 93XX controllers.

Index [hex]	Subindex	Name	Data type		Rights
1A01	0	Number of mapped objects in PDO's	PDO Mapping	U 8	ro
	1	PDO mapping 1		U 32	
	2	PDO mapping 2		U 32	
	3	PDO mapping 3		U 32	
	4	PDO mapping 4		U 32	

$1A02_{hex}$: Transmit PDO3* mapping parameter

This object serves to send parameter data as PDO3.

Note!

The functionality of this object is not yet available at the moment. On the attempt to read the object, the following values are sent back in dependence of the subindices:

Subind. 0: value 4_{hex} (max. supported subindex)

Subind. 1 - 4: In case of read requests regarding this object the value 10_{hex} will be sent back.

*) The object is not available for the 82XX, 8200 vector/motec and 93XX controllers.

Index [hex]	Subindex	Name	Data type		Rights
1A02	0	Number of mapped objects in PDO's	PDO Mapping	U 8	ro
	1	PDO mapping 1		U 32	
	2	PDO mapping 2		U 32	
	3	PDO mapping 3		U 32	
	4	PDO mapping 4		U 32	

7.8 Troubleshooting

No communication with the controller

Possible causes	Diagnostics	Remedy
Is the controller switched on?	The operation status LED of the basic unit must be on 🕮 7.4-1 .	Supply controller with voltage (see Operating Instructions for the basic unit)
Is the fieldbus module supplied with voltage?	The green LED "Status controller connection" at the fieldbus module 🕮 7.4-1 must be on (remedy 1) or blinking (remedy 2)	In case of supply from the basic unit check correct connection. With external supply check the 24 V voltage at terminals 39 and 59. A voltage of 24 V +10 % must be applied.
		The fieldbus module has not been initialised with the controller yet. Possibility 1: Controller not switched on (see fault possibility 1). Possibility 2: Check the connection to the controller
Does the controller receive telegrams?	The LED "Status bus connection" at the fieldbus module 🕮 7.4-1 must blink green when communicating with host.	Check wether the connection corresponds to the instruciton given in chapter "electrical installation". Check whether host sends telegrams and uses the appropriate interface.
		Has the available device address already been assigned? Check the setting of the other participants in the DeviceNet.

7.9 Appendix

7.9.1 Code table

Overview

How to read the table

Column	Abbreviation	Meaning
Code	L-C1853	(Lenze) code C1853
Subcode	1 2 .. 4	Subcode 1 of code C1853 Subcode 2 to 4 of code C1853
Index	-	The indicated value must be entered into byte 2 and byte 3 of the parameter telegram.
Lenze		Default setting of code
Selection	1 [1] 99	minimum value [minimum increment/unit] maximum value
Data type	-	VS: Visible String, character string with given length FIX32: Fixed value 4 bytes (= 32 bits) U16: Unsigned Integer 2 bytes (= 16 bits)

L-C18xx
(for 82xx, 8200 vector and 93XX)

Code	Subcode	Index	Possible settings: Lenze	Possible settings: Selection	Data type	Name
L-C1810	-	22765_d = $58ED_h$	-	-	VS	Software product code
L-C1811	-	22764_d = $58EC_h$			VS	Software creation date
L-C1850	-	$58C5_h$ = 22725_d	1	1 [1] 63	FIX32	Node address
L-C1851	-	$58C4_h$ = 22724_d	0	0 = 500 kbits/s, 1 = 250 kbits/s 2 = 125 kbits/s, 3 = 50 kbits/s 4 = 1000 kbits/s, 5 = 20 kbits/s 6 = 10 kbits/s	FIX32	baud rate
L-C1852	-	$58C3_h$ = 22723_d	0	0 = Slave operation 1 = Master operation	FIX32	Master/slave operation
L-C1853	/1 ... /3	$58C2_h$ = 22722_d	0	0 = Addressing to CANopen 1 = Addressing to L-C1854/L-C2354 2 = Addressing to LENZE system bus 3 = Addressing to CANopen index $14XX_h$/$18XX_h$	FIX32	Addressing CAN-INx/ CAN-OUTx
L-C1854	/1 ... /2 /3* ... /6*	$58C1_h$ = 22721_d	/1: 129 /2: 1 /3: 257* /4: 258* /5: 385* /6: 386*	0 [1] 1663	FIX32	Selective addressing CAN-IN/ CAN-OUT
L-C1855	/1 ... /2 /3* ... /6*	$58C0_h$ = 22720_d	0	0 [1] 2047	FIX32	Display of resulting identifiers
L-C1856	/1 ... /5	$58BF_h$ = 22719_d	/1: 3000 ms /2 .. /5: 0 ms	0 [1 ms] 655 35	FIX32	Boot up and cycle times
L-C1857	/1 ... /4	$58BE_h$ = 22718_d	3000 ms	0 [1 ms] 655 35	FIX32	Monitoring time
L-C1859	-	$58BC_h$ = 22716_d	-	0 [1] 1023	U16	Display of DIP switch position

Code	Subcode	Index	Possible settings		Data type	Name
			Lenze	Selection		
L-C1860	-	$58BB_h$ = 22715_d	-	0 [1] 1023	U16	Display of the current DIP switch position
L-C1867	-	$58B4_h$ = 22708_d	128	0 [1] 2047	FIX32	Sync Rx identifier
L-C1868	-	$58B3_h$ = 22707_d	128	0 [1] 2047	FIX32	Sync Tx identifier
L-C1873	/1 /2*, /3*	$58AE_h$ = 22702_d	1	0 [1] 240	FIX32	Sync rate CAN-IN1 ... CAN-IN3
L-C1874	/1 /2*, /3*	$58AD_h$ = 22701_d	1	1 [1] 240	FIX32	Sync rate CAN-OUT1 ... CAN-OUT3
L-C1875	/1 /2*, /3*	$58AC_h$ = 22700_d	/1: 0 /2: 1* /3: 1*	0 [1] 3	FIX32	Tx mode CAN-OUT1 ... CAN-OUT3
L-C1876	/1 ... /4	$58AB_h$ = 22699_d	65535	0 [1] 65535	FIX32	Masks CAN-OUT1
L-C1877	/1 ... /4	$58AA_h$ = 22698_d	65535	0 [1] 65535	FIX32	Masks CAN-OUT2
L-C1878	/1 ... /4	$58A9_h$ = 22697_d	65535	0 [1] 65535	FIX32	Masks CAN-OUT3
L-C1882	/1 ... /5	$58A5_h$ = 22693_d	0	0 [1] 2 0: no response 1: Controller inhibit 2: Quickstop	FIX32	Monitoring reactions

*) not effective when using 82XX, 8200 vector or 93XX controller

L-C23xx
(for servo PLC 9300 / Drive PLC)

Code	Subcode	Index	Possible settings		Data type	Name
			Lenze	Selection		
L-C1810	-	22765_d = $58ED_h$	-	-	VS	Software product code
L-C1811	-	22764_d = $58EC_h$			VS	Software creation date
L-C2120	-	22455_d = $57B7_h$	0	0 = No command 1 = Update codes L-23XX and CAN re-initialisation ≡ Reset node 2 = Update codes L-C23XX 10 = L-C2356/1...4 re-read 11 = L-C2357 re-read 12 = L-C2375 re-read 13 = L-C2376 ... L-C2378 re-read 14 = L-C2382 re-read 15 = not assigned	FIX32	AIF control byte
L-C2121	-	22454_d = $57B6_h$	0	0 [1] 255	FIX32	AIF status byte
L-C2350	-	$56D1_h$ = 22225_d	1	1 [1] 63	FIX32	Node address
L-C2351	-	$56D0_h$ = 22224_d	0	0 = 500 kbits/s, 1 = 250 kbits/s 2 = 125 kbits/s, 3 = 50 kbits/s 4 = 1000 kbits/s	FIX32	baud rate
L-C2352	-	$56CF_h$ = 22223_d	0	0 = Slave operation 1 = Master operation	FIX32	Master/slave operation
L-C2353	/1 ... /3	$56CE_h$ = 22222_d	0	0 = Addressing to CANopen 1 = Addressing to L-C1854/L-C2354 2 = Addressing to LENZE system bus	FIX32	Addressing CAN-INx/ CAN-OUTx

Code	Subcode	Index	Possible settings		Data type	Name
			Lenze	Selection		
L-C2354	/1 ... /6	56CD$_h$ = 22221$_d$	/1: 129 /2: 1 /3: 257 /4: 258 /5: 385 /6: 386	0 [1] 513	FIX32	Selective addressing CAN-IN/ CAN-OUT
L-C2355	/1 ... /6	56CC$_h$= 22220$_d$	-	0 [1] 2047	FIX32	Display of resulting identifiers
L-C2356	/1 ... /5	56CB$_h$ = 22219$_d$	1: 3000 ms 2 .. 5: 0 ms	0 [1 ms] 6553 5	FIX32	Boot up and cycle times
L-C2357	/1 ... /4	56CA$_h$ = 22218$_d$	3000 ms	0 [1 ms] 6553 5	FIX32	Monitoring time
L-C2359	-	56C8$_h$ = 22216$_d$	-	0 [1] 1023	U16	Display of DIP switch position
L-C2367	-	56C0$_h$ = 22208$_d$	128	0 [1] 2047	FIX32	Sync Rx identifier
L-C2368	-	56BF$_h$ = 22207$_d$	128	0 [1] 2047	FIX32	Sync Tx identifier
L-C2373	/1 ... /3	56BA$_h$ = 22202$_d$	1	1 [1] 240	FIX32	Sync rate CAN-IN1 ... CAN-IN3
L-C2374	/1 ... /3	56B9$_h$ = 22201$_d$	1	1 [1] 240	FIX32	Sync rate CAN-OUT1 ... CAN-OUT3
L-C2375	/1 ... /3	56B8$_h$ = 22200$_d$	/1: 0 /2: 1 /3: 1	0 [1] 3	FIX32	Tx mode CAN-OUT1 ... CAN-OUT3
L-C2376	/1 ... /4	56B7$_h$ = 22199$_d$	65535	0 [1] 65535	FIX32	Masks CAN-OUT1
L-C2377	/1 ... /4	56B6$_h$ = 22198$_d$	65535	0 [1] 65535	FIX32	Masks CAN-OUT2
L-C2378	/1 ... /4	56B5$_h$ = 22197$_d$	65535	0 [1] 65535	FIX32	Masks CAN-OUT3
L-C2382	/1 ... /5	56B1$_h$ = 22193$_d$	0	0 [1] 2 0: no response 1: Controller inhibit 2: Quickstop	FIX32	Monitoring reactions

7.10 Index