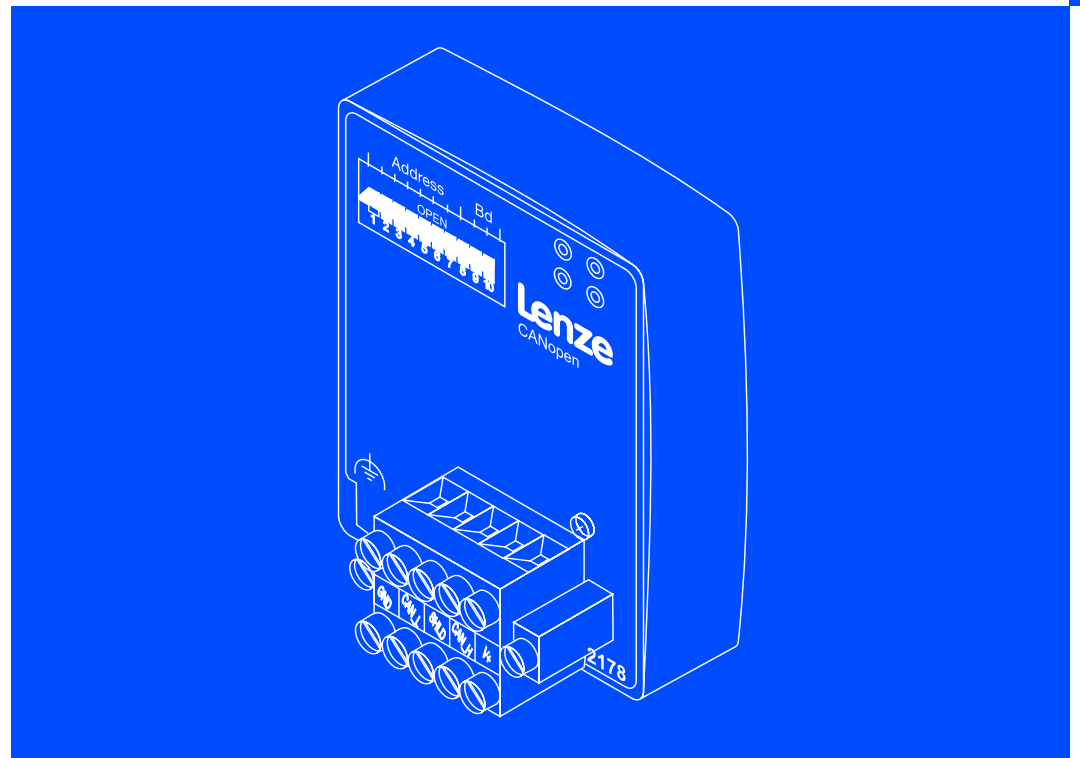


## CANopen



**EMF2178IB**

**Communication module**

|          |                                                            |           |
|----------|------------------------------------------------------------|-----------|
| <b>1</b> | <b>About this documentation .....</b>                      | <b>5</b>  |
| 1.1      | Document history .....                                     | 7         |
| 1.2      | Conventions used .....                                     | 8         |
| 1.3      | Terminology used .....                                     | 9         |
| 1.4      | Notes used .....                                           | 10        |
| <b>2</b> | <b>Safety instructions .....</b>                           | <b>11</b> |
| 2.1      | General safety information .....                           | 11        |
| 2.2      | Device- and application-specific safety instructions ..... | 12        |
| 2.3      | Residual hazards .....                                     | 12        |
| <b>3</b> | <b>Product description .....</b>                           | <b>13</b> |
| 3.1      | Application as directed .....                              | 13        |
| 3.2      | Identification .....                                       | 14        |
| 3.3      | Product features .....                                     | 15        |
| 3.4      | Connections and interfaces .....                           | 16        |
| <b>4</b> | <b>Technical data .....</b>                                | <b>17</b> |
| 4.1      | General data and operating conditions .....                | 17        |
| 4.2      | Protective insulation .....                                | 18        |
| 4.3      | Communication time .....                                   | 19        |
| 4.4      | Dimensions .....                                           | 20        |
| <b>5</b> | <b>Installation .....</b>                                  | <b>21</b> |
| 5.1      | Mechanical installation .....                              | 22        |
| 5.2      | Electrical installation .....                              | 23        |
| 5.2.1    | Wiring according to EMC (CE-typical drive system) .....    | 23        |
| 5.2.2    | Wiring with a host (master) .....                          | 24        |
| 5.2.3    | Wiring system bus (CAN) .....                              | 25        |
| 5.2.4    | Specification of the transmission cable .....              | 26        |
| 5.2.5    | Bus cable length .....                                     | 27        |
| 5.2.6    | Voltage supply .....                                       | 30        |
| <b>6</b> | <b>Commissioning .....</b>                                 | <b>32</b> |
| 6.1      | Before switching on .....                                  | 32        |
| 6.2      | Installing EDS files .....                                 | 32        |
| 6.3      | Setting node address and baud rate .....                   | 33        |
| 6.4      | Initial switch-on .....                                    | 36        |
| 6.5      | Enable drive via the communication module .....            | 37        |

|           |                                                                |           |
|-----------|----------------------------------------------------------------|-----------|
| <b>7</b>  | <b>Replacing the EMF2172IB communication module (CAN)</b>      | <b>39</b> |
| <b>8</b>  | <b>Data transfer</b>                                           | <b>42</b> |
| 8.1       | Structure of the CAN telegram                                  | 42        |
| 8.2       | CAN communication phases / network management (NMT)            | 45        |
| <b>9</b>  | <b>Process data transfer</b>                                   | <b>48</b> |
| 9.1       | Available process data objects                                 | 48        |
| 9.2       | Configuring process data channel                               | 50        |
| 9.3       | Cyclic process data objects                                    | 51        |
| 9.3.1     | Process data signals of Lenze controllers                      | 52        |
| 9.3.2     | Mapping in CANopen objects (I-160x, I-1A0x)                    | 65        |
| <b>10</b> | <b>Parameter data transfer</b>                                 | <b>69</b> |
| 10.1      | Access to the codes of the controller                          | 70        |
| 10.2      | Lenze parameter sets                                           | 71        |
| 10.2.1    | Parameter sets for 8200 vector controller                      | 71        |
| 10.2.2    | Parameter sets for controller 93XX                             | 72        |
| 10.3      | Structure of the parameter data telegram                       | 73        |
| 10.4      | Error codes                                                    | 76        |
| 10.5      | Examples of parameter data telegram                            | 77        |
| 10.6      | Special features for parameter setting of the drive controller | 81        |
| 10.6.1    | 8200 vector controller                                         | 81        |
| 10.6.2    | 9300 Servo PLC / Drive PLC / ECS                               | 81        |
| <b>11</b> | <b>Monitoring</b>                                              | <b>83</b> |
| 11.1      | Heartbeat Protocol                                             | 83        |
| 11.2      | Node Guarding Protocol                                         | 85        |
| 11.3      | Emergency telegram                                             | 87        |
| <b>12</b> | <b>Diagnostics</b>                                             | <b>88</b> |
| 12.1      | Measures in case of troubled communication                     | 88        |
| 12.2      | LED status displays                                            | 89        |

|           |                                                                |            |
|-----------|----------------------------------------------------------------|------------|
| <b>13</b> | <b>Implemented CANopen objects .....</b>                       | <b>91</b>  |
| 13.1      | Reference between CANopen object and Lenze code .....          | 91         |
| 13.2      | Overview .....                                                 | 92         |
| 13.2.1    | I-1000: Device type .....                                      | 96         |
| 13.2.2    | I-1001: Error register .....                                   | 96         |
| 13.2.3    | I-1003: Error history .....                                    | 97         |
| 13.2.4    | I-1005: COB-ID SYNC message .....                              | 98         |
| 13.2.5    | I-1006: Communication cycle period .....                       | 99         |
| 13.2.6    | I-1008: Manufacturer's device name .....                       | 99         |
| 13.2.7    | I-100A: Manufacturer software version .....                    | 100        |
| 13.2.8    | I-100C: Guard time .....                                       | 100        |
| 13.2.9    | I-100D: Life time factor .....                                 | 101        |
| 13.2.10   | I-1010: Store parameters .....                                 | 101        |
| 13.2.11   | I-1011: Restore default parameters .....                       | 102        |
| 13.2.12   | I-1014: COB-ID emergency object .....                          | 104        |
| 13.2.13   | I-1015: Emergency inhibit time .....                           | 105        |
| 13.2.14   | I-1016: Consumer heartbeat time .....                          | 105        |
| 13.2.15   | I-1017: Producer heartbeat time .....                          | 106        |
| 13.2.16   | I-1018: Module device description .....                        | 106        |
| 13.2.17   | I-1029: Error behaviour .....                                  | 106        |
| 13.2.18   | I-1200/I-1201: Server SDO parameters .....                     | 107        |
| 13.2.19   | I-1400 ... I-1402: Receive PDO communication parameters .....  | 109        |
| 13.2.20   | I-1600 ... I-1602: Receive PDO mapping parameters .....        | 111        |
| 13.2.21   | I-1800 ... I-1802: Transmit PDO communication parameters ..... | 112        |
| 13.2.22   | I-1A00 ... I-1A02: Transmit PDO mapping parameters .....       | 114        |
| <b>14</b> | <b>Codes .....</b>                                             | <b>115</b> |
| 14.1      | Overview .....                                                 | 115        |
| 14.2      | Communication-relevant Lenze codes .....                       | 119        |
| 14.3      | Important controller codes .....                               | 144        |
| <b>15</b> | <b>Index .....</b>                                             | <b>149</b> |

# 1 About this documentation

## Contents

This documentation exclusively contains descriptions of the EMF2178IB communication module (CANopen).



### Note!

This documentation supplements the **mounting instructions** supplied with the function/communication module and the **documentation of the used standard device**.

**The mounting instructions contain safety instructions which must be observed!**

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

Examples illustrate typical applications.

Furthermore this documentation contains the following:

- ▶ Safety instructions that must be observed.
- ▶ Key technical data relating to the communication module
- ▶ Information on versions of Lenze standard devices to be used.
- ▶ Notes on troubleshooting and fault elimination

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

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

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



### Tip!

For further information visit the homepage of the CAN user organisation CiA (CAN in Automation): [www.can-cia.org](http://www.can-cia.org).

**Target group**

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

**Tip!**

Information and auxiliary devices related to the Lenze products can be found in the download area at

**<http://www.Lenze.com>**

**Validity information**

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

- EMF2178IB communication modules (CANopen) as of version 1x.2x.

## 1.1 Document history

| Material no. | Version |         |      | Description                                                                                                                                     |
|--------------|---------|---------|------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| -            | 1.0     | 01/2008 | TD17 | First edition                                                                                                                                   |
| 13127634     | 2.0     | 07/2011 | TD17 | General revision                                                                                                                                |
| 13437291     | 3.0     | 06/2013 | TD17 | <ul style="list-style-type: none"><li>• New chapter "Replacing communication module EMF2172IB (CAN)" (📖 39)</li><li>• General updates</li></ul> |

### Your opinion is important to us!

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

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

[feedback-docu@Lenze.de](mailto:feedback-docu@Lenze.de)

Thank you for your support.

Your Lenze documentation team

# 1 About this documentation

## Conventions used

### 1.2 Conventions used

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

| Type of information     | Identification                                                                    | Examples/notes                                                                                                                                                                                      |
|-------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Spelling of numbers     |                                                                                   |                                                                                                                                                                                                     |
| Decimal separator       | Point                                                                             | In general, the decimal point is used.<br>For instance: 1234.56                                                                                                                                     |
| Decimal                 | Standard notation                                                                 | Example: 1234                                                                                                                                                                                       |
| Hexadecimal             | 0x[0 ... 9, A ... F]                                                              | Example: 0x60F4                                                                                                                                                                                     |
| Binary                  | 0b[0, 1]                                                                          | Example: '0b0110'                                                                                                                                                                                   |
| • Nibble                | Point                                                                             | Example: '0b0110.0100'                                                                                                                                                                              |
| Text                    |                                                                                   |                                                                                                                                                                                                     |
| Program name            | » «                                                                               | PC software<br>For example: »Engineer«, »Global Drive Control« (GDC)                                                                                                                                |
| Icons                   |                                                                                   |                                                                                                                                                                                                     |
| Page reference          |  | Reference to another page with additional information<br>For instance:  16 = see page 16                           |
| Documentation reference |  | Reference to another documentation with additional information<br>For example:  EDKxxx = see documentation EDKxxx |

### 1.3 Terminology used

| Term                           | Meaning                                                                                                                                                                                                                                                 |
|--------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Standard device                | Lenze controllers that can be used with the communication module.                                                                                                                                                                                       |
| Drive                          |  13                                                                                                                                                                    |
| »Global Drive Control« / »GDC« | PC software from Lenze which supports you in "engineering" (parameter setting, diagnosing, and configuring) during the entire life cycle, i.e. from planning to maintenance of the commissioned machine.                                                |
| Code                           | Parameter which serves to parameterise and monitor the controller. In normal speech, the term is usually called "Index".                                                                                                                                |
| Subcode                        | If a code contains more than one parameter, these parameters are stored in "subcodes".<br>In this documentation a slash "/" is used as a separator between code and subcode (e.g. "C00118/3").<br>In normal speech, the term is also called "Subindex". |
| Lenze setting                  | These are settings the device is preconfigured with ex works.                                                                                                                                                                                           |
| Basic setting                  |                                                                                                                                                                                                                                                         |
| HW                             | Hardware                                                                                                                                                                                                                                                |
| SW                             | Software                                                                                                                                                                                                                                                |
| PDO                            | Process data object                                                                                                                                                                                                                                     |
| SDO                            | Service data object                                                                                                                                                                                                                                     |

## 1.4 Notes used

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

### Safety instructions

Structure of safety instructions:



#### **Danger!**

(characterises the type and severity of danger)

#### **Note**

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

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

### Application notes

| Pictograph and signal word                                                                       | Meaning                                        |
|--------------------------------------------------------------------------------------------------|------------------------------------------------|
|  <b>Note!</b> | Important note to ensure troublefree operation |
|  <b>Tip!</b>  | Useful tip for simple handling                 |
|               | Reference to another documentation             |

## 2 Safety instructions



### **Note!**

It is absolutely vital that the stated safety measures are implemented in order to prevent serious injury to persons and damage to material assets.

Always keep this documentation to hand in the vicinity of the product during operation.

### 2.1 General safety information



### **Danger!**

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

- ▶ Lenze drive and automation components ...
  - ... must only be used for the intended purpose.
  - ... must never be operated if damaged.
  - ... must never be subjected to technical modifications.
  - ... must never be operated unless completely assembled.
  - ... must never be operated without the covers/guards.
  - ... can - depending on their degree of protection - have live, movable or rotating parts during or after operation. Surfaces can be hot.
- ▶ For Lenze drive components ...
  - ... only use permitted accessories.
  - ... only use original manufacturer spare parts.
- ▶ All specifications of the corresponding enclosed documentation must be observed.

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

The procedural notes and circuit details provided in this document are proposals which the user must check for suitability for his application. The manufacturer does not accept any liability for the suitability of the specified procedures and circuit proposals.
- ▶ Only qualified skilled personnel are permitted to work with or on Lenze drive and automation components.

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

  - ... who are familiar with the installation, assembly, commissioning and operation of the product,
  - ... possess the appropriate qualifications for their work,
  - ... and are acquainted with and can apply all the accident prevent regulations, directives and laws applicable at the place of use.

**2.2****Device- and application-specific safety instructions**

- ▶ During operation, the communication module must be securely connected to the standard device.
- ▶ With external voltage supply, always use a separate power supply unit, safely separated in accordance with EN 61800-5-1 in every control cabinet (SELV/PELV).
- ▶ Only use cables that meet the given specifications. (📖 26)

**Documentation of the standard device, control system, and plant/machine**

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

**2.3****Residual hazards****Protection of persons**

- ▶ If controllers are connected to phase-earthed system with a rated mains voltage  $\geq 400$  V, external measures need to be implemented to provide reliable protection against accidental contact. (see chapter "4.2", 📖 18)

**Device protection**

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

## 3 Product description

### 3.1 Application as directed

The communication module ...

- enables communication with Lenze controllers over the CAN bus with the CANopen communication profile.
- is a device intended for use in industrial power systems.
- can be used in conjunction with the following Lenze controllers:

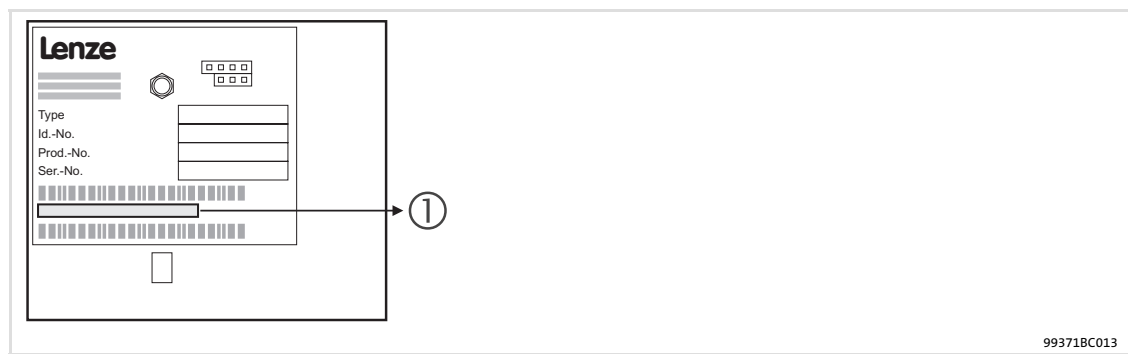
| Series           | Device type                 | Version |                  | Explanation                    |
|------------------|-----------------------------|---------|------------------|--------------------------------|
|                  |                             | HW      | SW <sup>1)</sup> |                                |
| 8200 vector      | E82xVxxxKxBxxxXX            | Vx      | 1x               | 8200 vector frequency inverter |
| 9300             | EVS9321-xS ... EVS9332-xS   | 2x      | 1x               | Servo inverter                 |
|                  | EVS9321-xK ... EVS9332-xK   | 2x      | 1x               | Servo cam profiler             |
|                  | EVS9321-xP ... EVS9332-xP   | 2x      | 1x               | Servo position controller      |
|                  | EVS9321-xR ... EVS9332-xR   | 2x      | 1x               | Servo register controller      |
|                  | EVS9321-xI ... EVS9332-xI   | 2x      | 8x               | 9300 servo PLC                 |
|                  | EVS9321-xT ... EVS9332-xT   | 2x      | 8x               |                                |
| 9300 vector      | EVF9321-xV ... EVF9333-xV   | 2x      | 1x               | 9300 vector frequency inverter |
|                  | EVF9335-xV ... EVF9338-xV   | 1x      | 0x               |                                |
|                  | EVF9381-xV ... EVF9383-xV   | 1x      | 0x               |                                |
| ECS servo system | ECSxSxxxC4xxxxXX            | 1A      | 6x               | "Speed and Torque"             |
|                  | ECSxPxxxC4xxxxXX            | 1A      | 6x               | "Posi and Shaft"               |
|                  | ECSxMxxxC4xxxxXX            | 1A      | 6x               | "Motion"                       |
|                  | ECSxAxxxC4xxxxXX            | 1A      | 6x               | "Application"                  |
| Drive PLC        | EPL10200-xI ... EPL10203-xI | 1x      | 8x               | Drive PLC                      |

1) Operating system software versions of the controllers

**Any other use shall be deemed inappropriate!**

3.2

Identification

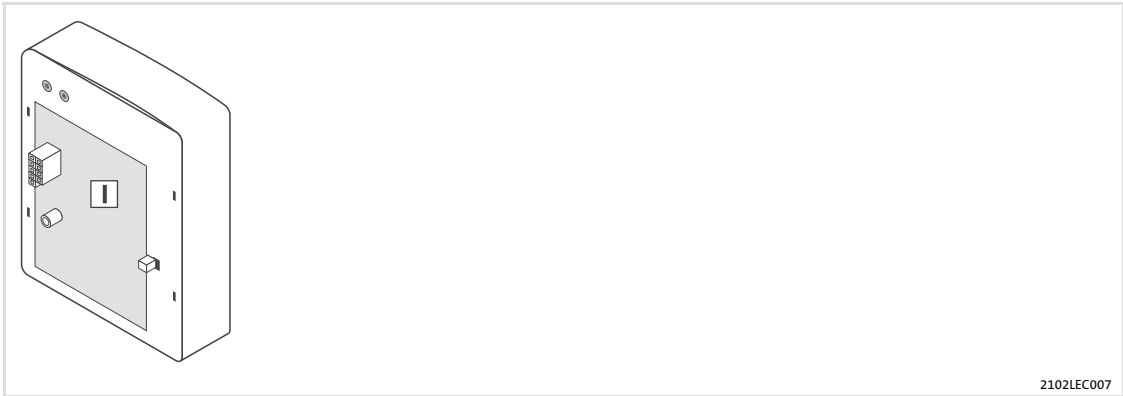
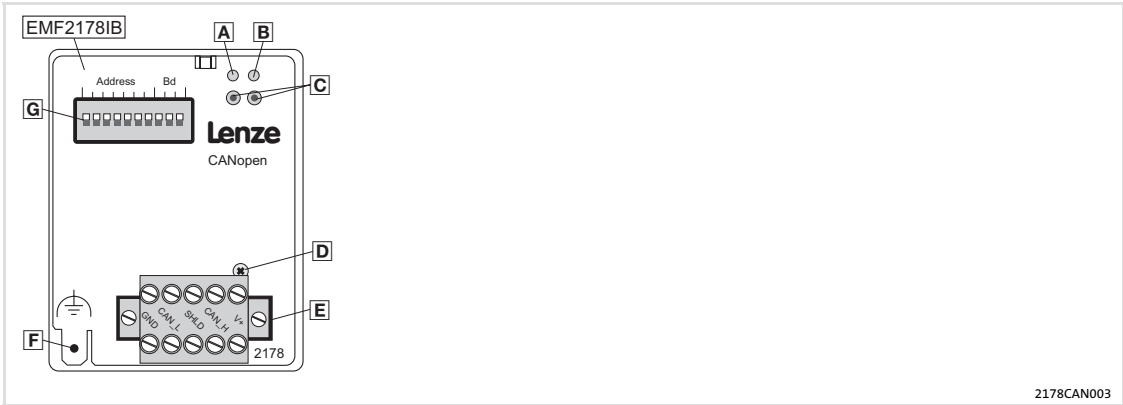


|                  |     |           |    |    |
|------------------|-----|-----------|----|----|
| Type code        | ① → | 33.2178IB | 1x | 2x |
| Series           |     |           |    |    |
| Hardware version |     |           |    |    |
| Software version |     |           |    |    |

### **3.3 Product features**

- ▶ Attachable communication module for the basic Lenze devices 8200 vector, 93XX, 9300 Servo PLC, Drive PLC, ECSXX
- ▶ Front DIP switches for setting the ...
  - CAN node address (max. 127 nodes)
  - baud rate (10, 20, 50, 125, 250, 500 and 1000 kbit/s)
- ▶ Bus expansion without repeater up to 7450 m
- ▶ Topology: Line terminated at both ends ( $R = 120\ \Omega$ )
- ▶ Simple connection through plug-on screw terminals
- ▶ Together with a 9300 Servo PLC, additional CANopen application profiles can be implemented.

3.4 Connections and interfaces



| Legend for fold-out page |                                                                                                                                                    |                      |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|
| Pos.                     | Description                                                                                                                                        | Detailed information |
| A                        | Connection status to standard device (two-colour LED)                                                                                              | 89                   |
| B                        | Connection status to fieldbus (two-colour LED)                                                                                                     |                      |
| C                        | Operating status of standard device (green and red Drive LED)                                                                                      |                      |
| D                        | Fixing screw                                                                                                                                       | 24                   |
| E                        | Plug connector with double screw connection, 5-pole                                                                                                |                      |
| F                        | PE shield cable connection                                                                                                                         |                      |
| G                        | DIP switches for setting the <ul style="list-style-type: none"><li>node address (switches 1 ... 7)</li><li>baud rate (switches 8 ... 10)</li></ul> | 33                   |
| I                        | Nameplate                                                                                                                                          | 14                   |

## 4 Technical data

### 4.1 General data and operating conditions

| Field                 | Values                                                                                                                                                                                                                                                                                                                                                                             |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Order designation     | EMF2178IB                                                                                                                                                                                                                                                                                                                                                                          |
| Communication media   | DIN ISO 11898                                                                                                                                                                                                                                                                                                                                                                      |
| Network topology      | Line terminated at both ends ( $R = 120 \Omega$ )                                                                                                                                                                                                                                                                                                                                  |
| Communication profile | CANopen, DS301 V4.01                                                                                                                                                                                                                                                                                                                                                               |
| Node addresses        | Max. 127                                                                                                                                                                                                                                                                                                                                                                           |
| Cable length          | Max. 7450 m (depending on the baud rate,  27)                                                                                                                                                                                                                                                   |
| Baud rate [kbit/s]    | 10, 20, 50, 125, 250, 500, 1000                                                                                                                                                                                                                                                                                                                                                    |
| Voltage supply        | Internal or external supply possible for basic devices: 8200 vector / 93XX / 9300 Servo PLC / Drive PLC / ECSXX<br>(also see  30)<br>External supply via separate power supply unit:<br>V+: $V = 24 \text{ V DC} \pm 10 \%$<br>I = 100 mA<br>GND: Reference potential for external voltage supply |



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

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

## 4.2

## Protective insulation

**Danger!****Dangerous electrical voltage**

If Lenze controllers are used on a phase earthed mains with a rated mains voltage  $\geq 400$  V, protection against accidental contact is not ensured without implementing external measures.

**Possible consequences:**

- Death or serious injury

**Protective measures:**

- If protection against accidental contact is required for the control terminals of the controller and the connections of the plugged device modules, ...
  - a double isolating distance must exist.
  - the components to be connected must be provided with the second isolating distance.

| Protective insulation between the bus and ...                                                                                                                                                                                            | Type of insulation according to EN 61800-5-1                                                             |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| Reference earth / PE                                                                                                                                                                                                                     | Functional insulation                                                                                    |
| External supply                                                                                                                                                                                                                          | No functional insulation                                                                                 |
| Power section <ul style="list-style-type: none"> <li>• 8200 vector</li> <li>• 9300 vector, Servo PLC</li> <li>• Drive PLC</li> <li>• ECSXX</li> </ul>                                                                                    | Double insulation<br>Double insulation<br>Double insulation<br>Double insulation                         |
| Control terminals <ul style="list-style-type: none"> <li>• 8200 vector (with internal supply,  30)</li> <li>• 8200 vector (with external supply,  30)</li> <li>• 9300 vector, Servo PLC</li> <li>• Drive PLC</li> <li>• ECSXX</li> </ul> | No functional insulation<br>Basic insulation<br>Basic insulation<br>Basic insulation<br>Basic insulation |

### 4.3 Communication time

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

The CAN bus communication times depend on ...

- ▶ the processing time in the controller (see documentation of the controller)
- ▶ Telegram runtime
  - baud rate
  - telegram length
- ▶ the data priority
- ▶ the bus load

#### Processing time in the controller



#### Documentation for the controller

Here you can find information on the processing times in the controller.

#### Telegram time

The telegram times depend on the baud rate and the telegram length:

| Baud rate<br>[kbit/s] | Telegram runtime<br>[ms] |         |         |
|-----------------------|--------------------------|---------|---------|
|                       | 0 bytes                  | 2 bytes | 8 bytes |
| 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    |

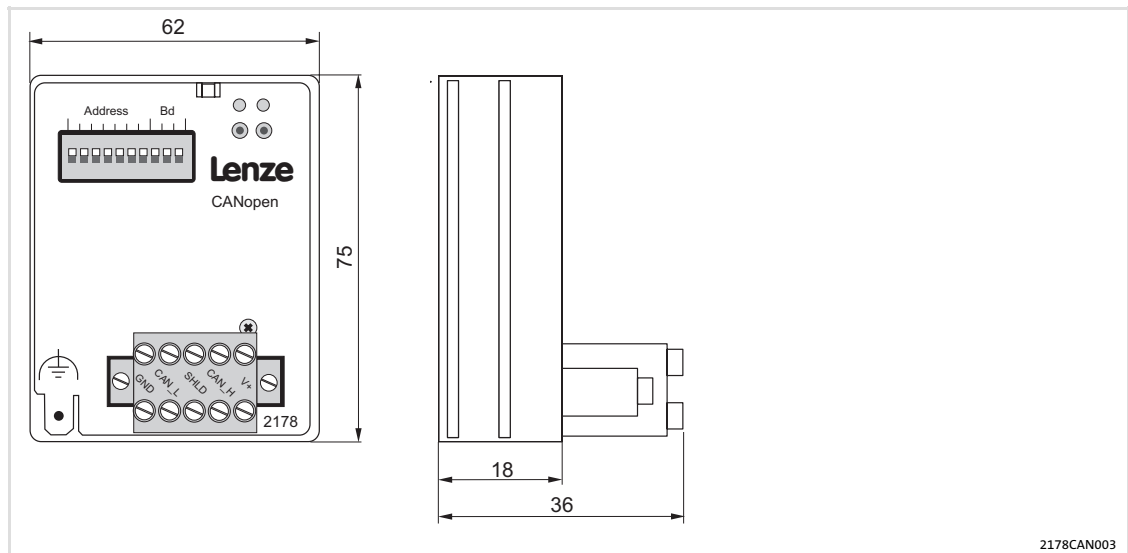
**4.4****Dimensions**

Fig. 4-1 Dimensions of the communication module (all dimensions in mm)

## 5

## Installation

**Danger!**

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

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

**Stop!**

The device contains components that can be destroyed by electrostatic discharge!

Before working on the device, the personnel must ensure that they are free of electrostatic charge by using appropriate measures.

### 5.1 Mechanical installation

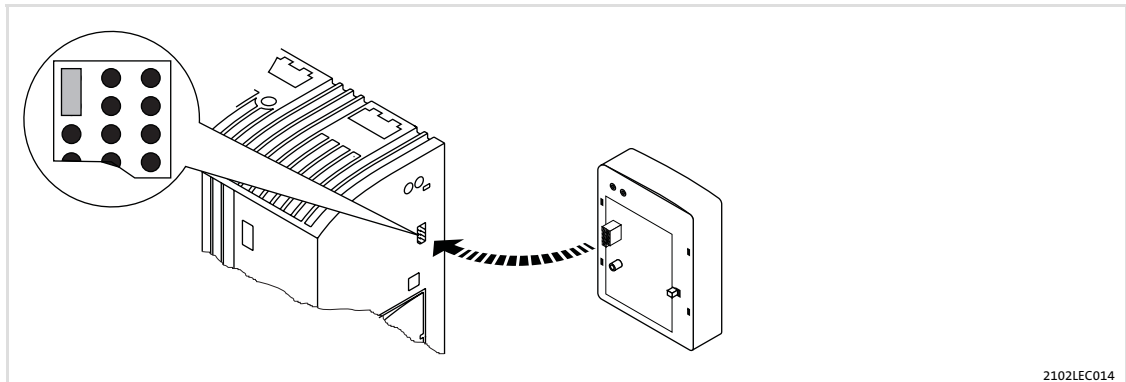


Fig. 5-1 Attaching the communication module

- ▶ Plug the communication module onto the standard device (here: 8200 vector).
- ▶ Tighten the communication module to the standard device using the fixing screw in order to ensure a good PE connection.



#### Note!

For the internal supply of the communication module by the 8200 vector frequency inverter the jumper has to be adjusted within the interface opening (see illustration above).

Observe the notes (📖 30).

## 5.2 Electrical installation

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

For wiring according to EMC requirements observe the following points:



#### Note!

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

#### Procedure for wiring

1. Observe the bus topology, i.e. do not use stubs.
2. Observe notes and wiring instructions in the documents for the control system.
3. Only use cables corresponding to the listed specifications (📖 26).
4. Observe the permissible bus cable length (📖 27).
5. Connect bus terminating resistors of 120  $\Omega$  each (scope of supply):
  - only to the physically first and last node
  - between the terminals CAN-LOW and CAN-HIGH

## 5.2.2

## Wiring with a host (master)

**Danger!**

An additional electrical isolation is required if a safe electrical isolation (reinforced insulation) to EN 61800-5-1 is necessary.

For this purpose for instance an interface module for the master computer with an additional electrical isolation can be used (see respective manufacturer information).

For wiring observe the electrical isolation of the supply voltage. The supply voltage is on the same potential as the data bus.

The 5-pole plug connector with double screw connection serves to

- ▶ connect the bus (📖 25);
- ▶ effect the external voltage supply (📖 30).

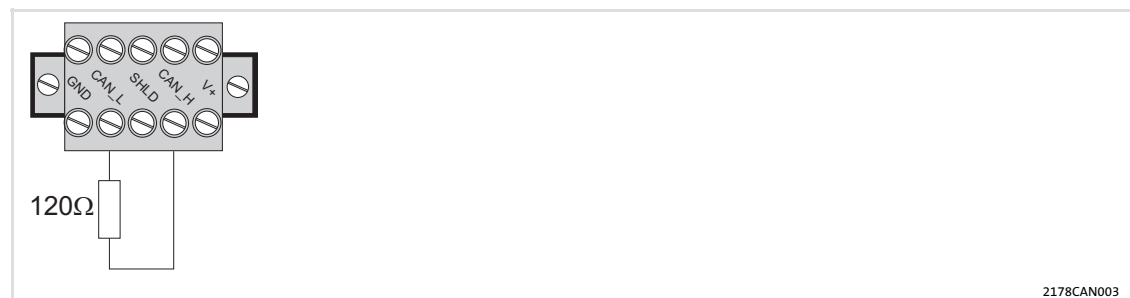


Fig. 5-2 5-pole plug connector with double screw connection

| Designation | Explanation                                                           |
|-------------|-----------------------------------------------------------------------|
| GND         | Reference potential for external voltage supply<br>CAN-GND connection |
| CAN_L       | Data line / input for terminating resistor 120 Ω                      |
| SHIELD      | Shielding                                                             |
| CAN_H       | Data line / input for terminating resistor 120 Ω                      |
| V+          | External voltage supply                                               |

### 5.2.3 Wiring system bus (CAN)

#### Structure of a CAN bus system (example)

The CAN bus system is designed as a 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. three process data channels (PDO = Process Data Object)
  - Process data are sent via the process data channel and are used for high-speed and high-priority control tasks. Typical process data are, for instance, control words, status words, setpoints and actual values of a standard device.
- ▶ Two parameter data channels (SDO = Service Data Object)
  - The parameters are transferred at lower priority than the process data and are set or changed e.g. during commissioning or product change.
  - The parameters are accessed via the parameter data channels of the EMF2178IB communication module to the codes of the basic Lenze device or the corresponding CANopen objects.
  - With both parameter data channels, two masters can be connected to a standard device. A PC (e.g. with the Lenze software "Global Drive Control") or an operator terminal serve to change parameters directly at the standard device during operation of a system connected to PLC. The second parameter data channel can be reached under the set address (via DIP switch or C0009) with an offset of "64". If, e.g., a PLC addresses the standard device with the address "1" and a second commanding device the address "65", always the same standard device is addressed.
  - The second parameter channel is deactivated in the default state.



#### Note!

- ▶ The last telegram determines the parameter when a parameter is accessed by two units (see CANopen objects 1200 and 1201 "Server SDO Parameters". (📖 107)).
- ▶ Please observe the notes in the chapter 6 Commissioning (📖 33), if you do not select the baud rate and address via the front DIP switches.

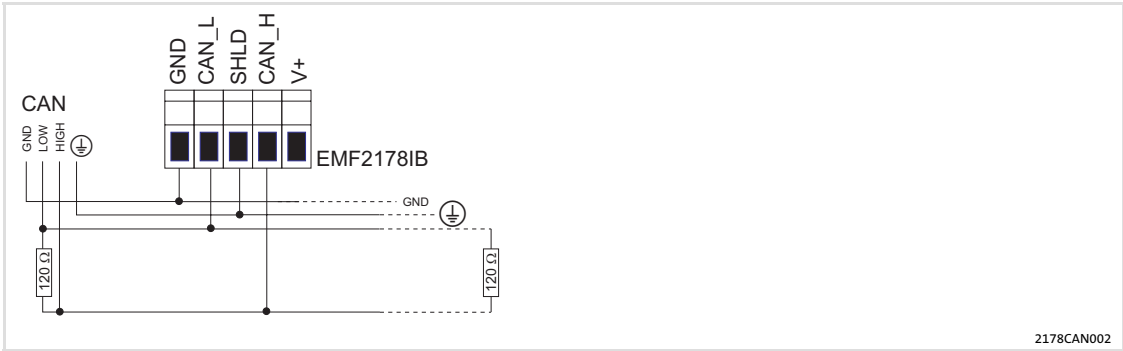


Fig. 5-3 Connection to the plug connector

5.2.4 Specification of the transmission cable

We recommend the use of CAN cables in accordance with ISO 11898-2:

| CAN cable in accordance with ISO 11898-2 |                                                   |
|------------------------------------------|---------------------------------------------------|
| Cable type                               | Paired with shielding                             |
| Impedance                                | 120 Ω (95 ... 140 Ω)                              |
| Cable resistance/cross-section           |                                                   |
| Cable length ≤ 300 m                     | ≤ 70 mΩ/m / 0.25 ... 0.34 mm <sup>2</sup> (AWG22) |
| Cable length 301 ... 1000 m              | ≤ 40 mΩ/m / 0.5 mm <sup>2</sup> (AWG20)           |
| Signal propagation delay                 | ≤ 5 ns/m                                          |

## 5.2.5 Bus cable length



### Note!

- ▶ It is absolutely necessary to comply with the permissible cable lengths.
- ▶ If the total cable lengths of the CAN nodes differ for the same baud rate, the smaller value must be used to determine the max. cable length.
- ▶ Observe the reduction of the total cable length due to the signal delay of the repeater.

### 5.2.5.1 Total cable length

The baud rate determines the total cable length.

| Baud rate [kbps] | Max. bus length [m] |
|------------------|---------------------|
| 10               | 7450                |
| 20               | 3950                |
| 50               | 1550                |
| 125              | 630                 |
| 250              | 290                 |
| 500              | 120                 |
| 1000             | 25                  |

Tab. 5-1 Total cable length

## 5.2.5.2 Segment cable length

The segment cable length is determined by the cable cross-section used and by the number of nodes. Repeaters divide the total cable length into segments. If no repeaters are used, the segment cable length is identical to the total cable length.

| Max. number of nodes per segment | Cable cross-section (can be interpolated) |                             |                              |                             |
|----------------------------------|-------------------------------------------|-----------------------------|------------------------------|-----------------------------|
|                                  | 0.25 mm <sup>2</sup> (AWG24)              | 0.5 mm <sup>2</sup> (AWG21) | 0.75 mm <sup>2</sup> (AWG19) | 1.0 mm <sup>2</sup> (AWG18) |
| 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                       |
| 100                              | 150 m                                     | 270 m                       | 410 m                        | 600 m                       |

Tab. 5-2 Segment cable length

## Example: Selection help

| Given                                       |                                                                                                                                                                        |
|---------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Total cable length to be implemented        | 200 m                                                                                                                                                                  |
| Number of nodes                             | 63                                                                                                                                                                     |
| Results                                     |                                                                                                                                                                        |
| Max. possible baud rate                     | 250 kbps<br>(derived from Tab. 5-1 Total cable length)                                                                                                                 |
| Required cable cross-section (interpolated) | 0.30 mm <sup>2</sup> (AWG23)<br>(derived from Tab. 5-2 Segment cable length)                                                                                           |
| Cable cross-section of standard CAN cable   | 0.34 mm <sup>2</sup> (AWG22)<br>(see specification of the transmission cable  26) |

### 5.2.5.3 Checking the use of repeaters

Compare the values derived from Tab. 5-1 Total cable length (📖 27) and Tab. 5-2 Segment cable length (📖 28).

- ▶ If the sum of the segment cable lengths is smaller than the total cable length to be implemented, either repeaters must be used or the cable cross-section must be increased.
- ▶ If the use of repeaters reduces the max. possible total cable length so much that it is smaller than the total cable length to be implemented, then the cable cross-section must be increased or less repeaters must be used or the baud rate must be decreased.
- ▶ The use of a further repeater is recommended as ...
  - service interface  
Advantage: Trouble-free connection during bus operation is possible.
  - calibration interface  
Advantage: The calibration/programming unit remains electrically isolated.

#### Example

| Given                                                             |                               |
|-------------------------------------------------------------------|-------------------------------|
| Total cable length to be implemented                              | 450 m                         |
| Number of nodes                                                   | 32                            |
| Cable cross-section                                               | 0.50 mm <sup>2</sup> (AWG 20) |
| Baud rate                                                         | 125 kbps                      |
| Repeater used                                                     | Lenze repeater EMF2176IB      |
| Reduction of the max. total cable length per repeater (EMF2176IB) | 30 m                          |

| Results                   |                                                                                         |
|---------------------------|-----------------------------------------------------------------------------------------|
| Max. total cable length   | 630 m<br>(see Tab. 5-1 Total cable length (📖 27))                                       |
| Max. segment cable length | 360 m<br>(see Tab. 5-2 Segment cable length (📖 28))                                     |
| Comparison                | The max. segment cable length is smaller than the total cable length to be implemented. |
| Conclusion                | A repeater must be installed at the determined max. segment cable length of 360 m.      |

| Results with 1 repeater   |                                                                                                                              |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------|
| Max. total cable length   | 600 m<br>(Reduction of the total cable length (📖 27) by 30 m)                                                                |
| Max. segment cable length | 720 m                                                                                                                        |
| Comparison                | Both the possible total cable length and the segment cable lengths are larger than the total cable length to be implemented. |
| Conclusion                | 1 repeater is sufficient to implement the total cable length of 450 m.                                                       |

## 5.2.6

## Voltage supply

## Internal voltage supply

**Note!**

Internal voltage supply has been selected in the case of standard devices with an extended AIF interface opening (e.g. front of 8200 vector). The area shown on a grey background in the graphic marks the jumper position.

- ▶ By default, this is **not** supplied internally in the standard device.
- ▶ For internal voltage supply place the jumper on the position indicated below.

In the case of all other device series (9300, ECS), voltage is always supplied from the standard device.

| Lenze setting<br>(Only external voltage supply possible.) | Internal voltage supply |
|-----------------------------------------------------------|-------------------------|
|                                                           |                         |

## External voltage supply

**Note!**

In the case of an external voltage supply and for greater distances between the control cabinets, always use a separate power supply unit (SELV/PELV) that is safely separated in accordance with EN 61800-5-1 in each control cabinet.

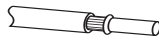
The external voltage supply of the communication module ...

- ▶ is required if communication via the fieldbus is to be continued in case the supply of the standard device fails.
- ▶ is carried out via the 2-pole plug connector with screw connection (24 V DC):

| Terminal | Description                                                                        |
|----------|------------------------------------------------------------------------------------|
| V+       | External voltage supply<br>$V = 24 \text{ V DC} \pm 10 \%$<br>$I = 100 \text{ mA}$ |
| GND      | Reference potential for external voltage supply                                    |

- ▶ The parameters of a basic device disconnected from the mains cannot be accessed.

### Terminal data

| Area                  | Values                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Electrical connection | Plug connector with screw connection                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Possible connections  | <div>rigid:</div> <div>  1.5 mm<sup>2</sup> (AWG 16) </div> <div>flexible:</div> <div>  without wire end ferrule<br/>1.5 mm<sup>2</sup> (AWG 16) </div> <div>  with wire end ferrule, without plastic sleeve<br/>1.5 mm<sup>2</sup> (AWG 16) </div> <div>  with wire end ferrule, with plastic sleeve<br/>1.5 mm<sup>2</sup> (AWG 16) </div> |
| Tightening torque     | 0.5 ... 0.6 Nm (4.4 ... 5.3 lb-in)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Stripping length      | 6 mm                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

**6****Commissioning**

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

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

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

**6.1****Before switching on****Stop!**

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

- ▶ the entire wiring with regard to completeness, short circuit, and earth fault.
- ▶ whether the bus system is terminated by a bus terminating resistor at the physically first and last node.

**6.2****Installing EDS files**

The EDS files serve to implement the Lenze communication modules for the AIF and FIF interfaces into the CANopen configuration software.

The single EDS files describe the implemented CANopen functions of the respective communication module and the "on board" Lenze system bus (CAN).

**Tip!**

The current EDS file required for configuring the EMF2178IB (CANopen) communication module can be found in the download area on:

**[www.Lenze.com](http://www.Lenze.com)**

### 6.3 Setting node address and baud rate

The node address and the baud rate can be set via codes or via the DIP switches arranged at the front:

- ▶ Node address: Switches **1 ... 7** / code **C1850/C2350**
- ▶ Baud rate: Switches **8 ... 10** / code **C1851/C2351**

The Lenze setting of all DIP switches is OFF.



#### Note!

##### Settings via codes

- ▶ In the Lenze setting – Address switches **1 ... 7 = OFF** –, the values are accepted from the codes **C1850/C2350** (node address) and **C1851/C2351** (baud rate).
  - Node addresses > 99 can only be set via DIP switch.
  - The baud rates 10 kbps and 20 kbps can only be selected via DIP switch.
- ▶ Writing the codes (e.g. with GDC via CAN) has a direct effect on the standard device codes C0009 and C0126.
- ▶ Acceptance of code changes by:
  - Switching off and then on again the voltage supply;
  - "Reset node" with C0358 = 1;
  - Network management command "Reset Communication";
  - Set C2120 (AIF control byte) = 1.
- ▶ The codes are inactive if at least one address switch (1 ... 7) has been set to ON position before a renewed mains connection.

Node address setting

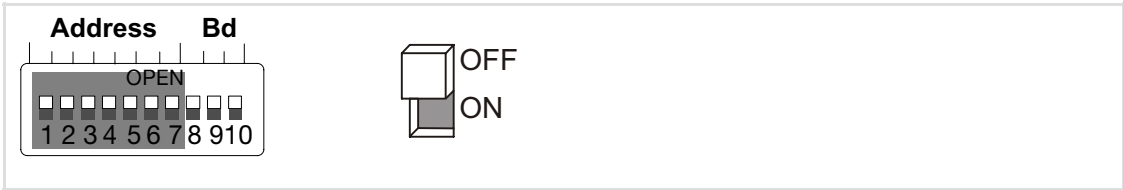


Fig. 6-1 Address assignment via DIP switch

- If several devices are connected to the CAN network, the node addresses must differ from each other.
- The desired node address results from the sum of the values of switches (1 ... 7) in ON position.

| Switch | Value | Example         |                            |
|--------|-------|-----------------|----------------------------|
|        |       | Switch position | Node address               |
| 1      | 64    | OFF             | 16 + 4 + 2 + 1 = <b>23</b> |
| 2      | 32    | OFF             |                            |
| 3      | 16    | <b>ON</b>       |                            |
| 4      | 8     | OFF             |                            |
| 5      | 4     | <b>ON</b>       |                            |
| 6      | 2     | <b>ON</b>       |                            |
| 7      | 1     | <b>ON</b>       |                            |



**Note!**

Switch off the voltage supply of the communication module, and then switch it on again to activate the changed settings.

Baud rate setting

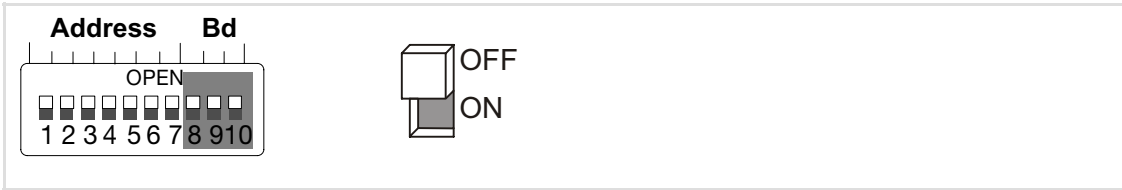


Fig. 6-2 Baud rate setting

- The baud rate must be the same for all CAN nodes.
- The following baud rates can be set:

| Baud rate [kbps] | Switch |     |     |
|------------------|--------|-----|-----|
|                  | 8      | 9   | 10  |
| 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 |



Note!

Switch off the voltage supply of the communication module, and then switch it on again to activate the changed settings.

## 6.4

## Initial switch-on

**Note!**

Do not change the setting sequence.

1. Switch on the standard device and, if necessary, the external voltage supply of the communication module.
  - The green LED **A** on the front of the communication module is lit.
  - The status LED of the standard device (Drive LED) **C** must be lit or blinking. The meaning of the signalling can be found in the standard device documentation.
2. You can now communicate with the drive, i. e.
  - all parameters (SDO) can be read;
  - all writable parameters (SDO) can be overwritten.
3. After a state change ("Operational") process data can be exchanged with the drive.

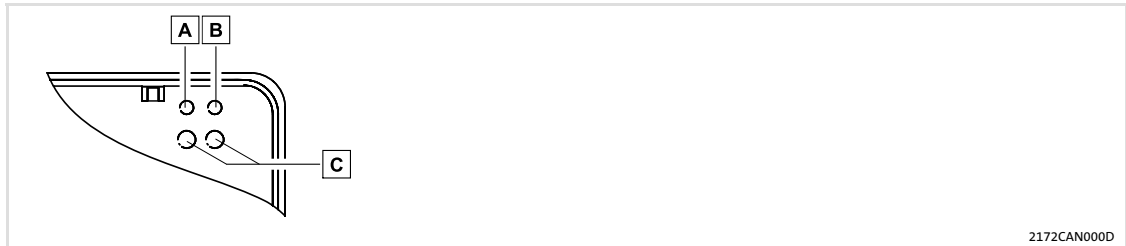


Fig. 6-3 LEDs of the communication module

## 6.5 Enable drive via the communication module



### Note!

- ▶ During operation, the plugging of the communication module onto a different controller may cause undefined operating states.
- ▶ Observe the information on the CAN configuration and controller enable in the documentation of the corresponding controller.

### Enable the 8200 vector via the communication module

| Step | Procedure                                                   | Comments                                                                                                                                                                                                                                                                                                                                                                         |
|------|-------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.   | Set C0001 from "0" to "3"                                   | The Lenze parameter C0001 (operating mode) can be set with the GDC, keypad XT or directly via CANopen.<br>Example for direct setting via CANopen:<br>Write (C0001 = 3)<br><ul style="list-style-type: none"> <li>• Index = 0x5FFE (resulting from: 0x5FFF - (C0001)<sub>hex</sub>)</li> <li>• Subindex: 0</li> <li>• Value: 30000 (resulting from: C0001 = 3 x 10000)</li> </ul> |
| 2.   | Assign 28 to HIGH level                                     | The terminal 28 (controller enable) is always active and must be assigned to HIGH level during CANopen operation. Otherwise the controller cannot be enabled via CANopen.                                                                                                                                                                                                        |
| 3.   | Assign input terminal for QSP to HIGH level                 | The quick stop function (QSP) is always active. If QSP is configured to an input terminal (Lenze setting: Not assigned), it must be assigned to HIGH level during CANopen operation.                                                                                                                                                                                             |
| 4.   | The controller now accepts parameter data and process data. |                                                                                                                                                                                                                                                                                                                                                                                  |

### Enable 93XX via the communication module

| Step | Procedure                                                                                                   | Comments                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------|-------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.   | Set C0005 to the value "xxx3"                                                                               | The value "xxx3" of the Lenze parameter C0005 (control of the controller via CANopen) can be set with the GDC, keypad XT or directly via CANopen.<br>Example for the first commissioning with the signal configuration "1013":<br>Write (C0005 = 1013)<br><ul style="list-style-type: none"> <li>• Index = 0x5FFA (resulting from: 0x5FFF - (C0005)<sub>hex</sub>)</li> <li>• Subindex: 0</li> <li>• Value: 10130000 (resulting from: C0005 = 1013 x 10000)</li> </ul> |
| 2.   | Set C0142 = 0                                                                                               | See "Protection against uncontrolled restart" (□ 38).                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 3.   | Assign 28 to HIGH level                                                                                     | The terminal 28 (controller enable) is always active and must be assigned to HIGH level during CANopen operation. Otherwise the controller cannot be enabled via CANopen.                                                                                                                                                                                                                                                                                              |
| 4.   | Assign terminal E1 to HIGH level                                                                            | If the signal configuration C0005 = 1013, the quick stop function (QSP) is assigned to the digital input terminals E1 and E2 in connection with the right/left change-over and thus always active.                                                                                                                                                                                                                                                                     |
| 5.   | Connect terminal X5/A1 to<br><ul style="list-style-type: none"> <li>• X5/28 and</li> <li>• X5/E1</li> </ul> | Only affects the signal configuration C0005 = xx13<br>With this signal configuration the terminal A1 is switched as voltage output.                                                                                                                                                                                                                                                                                                                                    |
| 6.   | The controller now accepts parameter data and process data.                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |

**Enable ECSXX via the communication module**

| Step | Procedure                                                   | Comments                                                                                                                                                                                                                   |
|------|-------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.   | Select control interface "AIF" via code.                    | See documentation of the corresponding ECS controller.                                                                                                                                                                     |
| 2.   | Set C0142 = 0                                               | See "Protection against uncontrolled restart" (□ 38).                                                                                                                                                                      |
| 3.   | Assign terminals X6/SI1 and X6/SI2 to HIGH level            | The terminals X6/SI1 (controller enable/inhibit) and X6/SI2 (pulse enable/inhibit) are always active and must be assigned to HIGH level during CANopen operation. Otherwise, the controller cannot be enabled via CANopen. |
| 4.   | The controller now accepts parameter data and process data. |                                                                                                                                                                                                                            |

**Protection against uncontrolled restart****Note!****Establishing communication**

If communication is to be established via an externally supplied communication module, initially the standard device must also be switched on.

After communication has been established, the externally supplied module is independent of the power on/off state of the standard device.

**Protection against uncontrolled restart**

After a fault (e.g. short-term mains failure), a restart of the drive is not always wanted and - in some cases - even not allowed.

The restart behaviour of the controller can be set in C0142:

- ▶ C0142 = 0 (Lenze setting)
  - The controller remains inhibited (even if the fault is no longer active).
  - The drive starts up in a controlled manner by explicit controller enable:
    - 93XX: Set terminal 28 to HIGH level.
    - ECSXX: Set terminals X6/SI1 and X6/SI2 to HIGH level.
- ▶ C0142 = 1
  - An uncontrolled restart of the drive is possible.

## 7

## Replacing the EMF2172IB communication module (CAN)

Observe the following information when replacing the **EMF2172IB (CAN)** communication module by **EMF2178IB (CANopen)**:

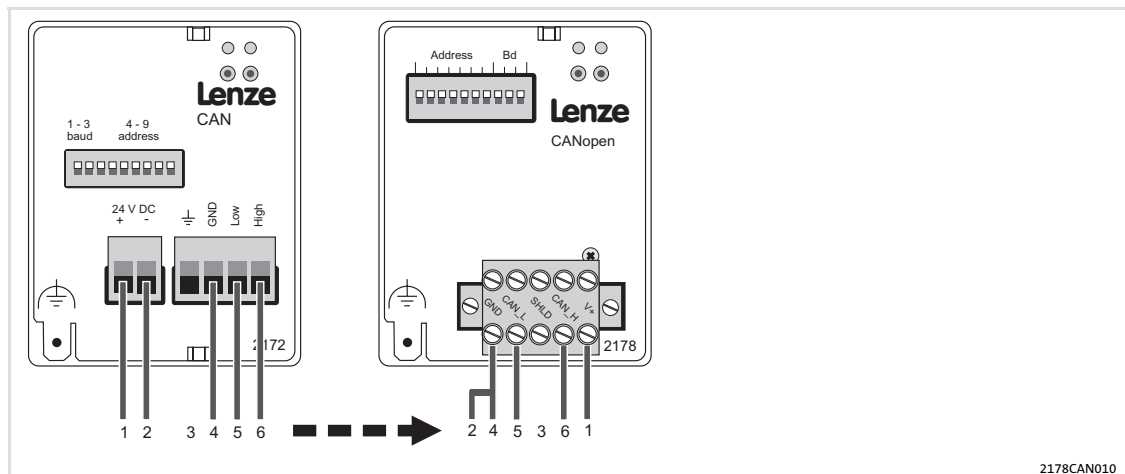
- The communication modules feature different plug connectors for connecting the voltage supply and the system bus (CAN).
- The DIP switches of the communication modules feature different assignments for setting the CAN node address and the baud rate (▢ 40). Furthermore there are different adjustable CAN address ranges:

| Communication module | CAN address range | DIP switch |
|----------------------|-------------------|------------|
| EMF2172IB (CAN)      | 1 ... 63          | S4 ... S9  |
| EMF2178IB (CANopen)  | 1 ... 127         | S1 ... S7  |

- Depending on the standard device used, the 2. SDO channel is activated via code **C1865/1** or **C2365/1** (▢ 41).

## Changing the wiring

The following illustration shows how you must carry out the previous wiring for the EMF2178IB communication module now.



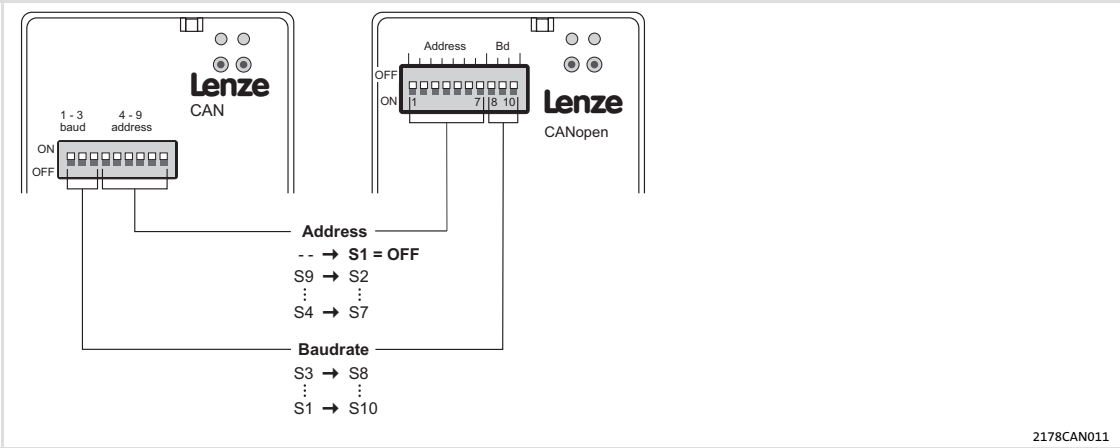
2178CAN010

DIP switch settings



**Note!**

On the **EMF2178IB** communication module, the switch **S1 = OFF**(Lenze setting) must remain set. (An address setting > 63 was not possible for EMF2172IB.)



| Switch    |           | Value | Example         |                            |
|-----------|-----------|-------|-----------------|----------------------------|
| EMF2172IB | EMF2178IB |       | Switch position | CAN node address           |
| -         | S1        | 64    | OFF             | 16 + 4 + 2 + 1 = <b>23</b> |
| S9        | S2        | 32    | OFF             |                            |
| S8        | S3        | 16    | ON              |                            |
| S7        | S4        | 8     | OFF             |                            |
| S6        | S5        | 4     | ON              |                            |
| S5        | S6        | 2     | ON              |                            |
| S4        | S7        | 1     | ON              |                            |

## Activate the 2. SDO channel

Depending on the standard device used, set code **C1865/1** or **C2365/1** to activate the 2. SDO channel.

| Settings via the GDC/XT keypad for these standard devices:                                                                                                                                                                                                                                                              |                                                                                                                                                                                                                                                                                                                      |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| EVS 93xx-ES<br>EVS 93xx-EP<br>EVS 93xx-EK<br>EVS 93xx-ER<br>EVF 93xx-EV<br>E82EVxxx                                                                                                                                                                                                                                     | EVS 93xx-EI<br>EVS 93xx-ET<br>EPL-10200<br>ECSxS/P/M/A                                                                                                                                                                                                                                                               |
| 1. Plug the EMF2178IB communication module onto the standard device.<br>2. Use the GDC to set code <b>C1865/1 = 1</b> .<br>The setting is saved with mains failure protection directly in the communication module.<br>When using the standard devices specified above, the setting can also be carried out beforehand. | Use the GDC or XT keypad to set standard device code <b>C2365/1 = 1</b> and save it with mains failure protection.<br>After mains connection of one of the standard devices specified above, or after plugging on the EMF2178IB communication module, the contents of C2365 are written to the communication module. |

## 8 Data transfer

Master and controller communicate with each other by exchanging data telegrams via the CAN bus.

The user data area of the CAN telegram either contains *network management data*, *process data* or *parameter data* (44).

Different communication channels are assigned to parameter and process data in the controller:

- ▶ Process data are transferred via the *process data channel*.
- ▶ Parameter data are transferred via the *parameter data channel*.

### 8.1 Structure of the CAN telegram

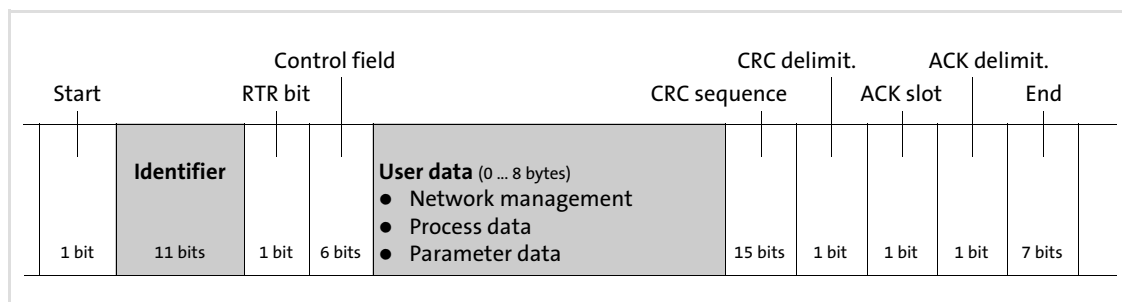


Fig. 8-1 Basic structure of the CAN telegram



### Tip!

The identifier and the user data will be explained in more detail in this documentation.

The other signals refer to the transfer characteristics of the CAN telegram which are not described in this documentation.

For more information please refer to the homepage of the CAN user organisation CiA (CAN in Automation):

<http://www.can-cia.org>

### Identifier (COB-ID)

The principle of the CAN communication is based on a message-oriented data exchange between a transmitter and various receivers. Here, all nodes can quasi simultaneously transmit and receive messages. In case of CANopen, a node orientation is achieved by having only one transmitter for each message.

The control which node is to receive a transmitted message is executed via the *identifier* in the CAN telegram, also called *COB-ID* (*Communication Object Identifier*). In addition to the addressing, the identifier contains information on the priority of the message and the type of user data.

With the exception of the network management and the sync telegram, the identifier consists of a basic identifier and the node address of the node to be addressed:

**Identifier (COB-ID) = basic identifier + adjustable node address (Node-ID)**

The identifier assignment is defined in the CANopen protocol.

### Basic identifier

The basic identifier is preset as default with the following values according to the CANopen specification:

| Object                     |       | Direction      |              | Basic identifier |       |
|----------------------------|-------|----------------|--------------|------------------|-------|
|                            |       | from the drive | to the drive | dec              | hex   |
| NMT                        |       |                |              | 0                | 0x000 |
| Sync                       |       |                |              | 128              | 0x080 |
| Emergency                  |       | X              |              | 128              | 0x080 |
| PDO1                       | TPDO1 | X              |              | 384              | 0x180 |
| (Process data channel 1)   | RPDO1 |                | X            | 512              | 0x200 |
| PDO2                       | TPDO2 | X              |              | 640              | 0x280 |
| (Process data channel 2)   | RPDO2 |                | X            | 768              | 0x300 |
| PDO3                       | TPDO3 | X              |              | 896              | 0x380 |
| (process data channel 3)   | RPDO3 |                | X            | 1024             | 0x400 |
| SDO1                       |       | X              |              | 1408             | 0x580 |
| (Parameter data channel 1) |       |                | X            | 1536             | 0x600 |
| SDO2                       |       | X              |              | 1472             | 0x5C0 |
| (parameter data channel 2) |       |                | X            | 1600             | 0x640 |
| Lenze setting: not active. |       |                |              |                  |       |
| Node guarding / heartbeat  |       | X              |              | 1792             | 0x700 |

### Node address (node ID)

Each node of the CAN network must be assigned with a node address (also called *node ID*) within the valid address range for unambiguous identification.

- ▶ A node address must not be assigned more than once within a network.
- ▶ The node address of the controller is configured under code **C1850/C2350** or via **DIP switch** (33).

**User data**

The user data area of the CAN telegram either contains *network management data*, *process data* or *parameter data*:

- ▶ Network management data (NMT data)
  - Network management data contain information on the establishment of communication via the CAN network (📖 45).
- ▶ Process data (PDO, Process Data Objects)
  - Process data are transferred via the *process data channel* (see also chapter "Process data transfer", 📖 48).
  - Process data serve to control the controller (slave).
  - Process data are transferred between the host and the controllers to ensure a continuous exchange of current input and output data.
  - The host has direct access to process data. In the PLC, the data are, for instance, directly assigned to the I/O area. An exchange between host and controller is required as fast as possible. Small amounts of data can be transferred cyclically.
  - Process data are not saved in the controller.
  - Process data are, for instance, control words, status words, setpoints and actual values.
- ▶ Parameter data (SDO, Service Data Objects)
  - Parameter data are transmitted as SDOs via the *parameter data channel* and acknowledged by the receiver, i.e. the sender gets a feedback about the transmission being successful or not (see also chapter "parameter data transfer", 📖 69).
  - The parameter data channel enables access to all Lenze codes and CANopen indices.
  - The parameters for instance are set for the initial system set-up during commissioning or when material is changed on the production machine.
  - In general, the parameter data transfer is not time-critical.
  - Parameter changes are automatically stored in the controller.
  - Parameter data for example are operating parameters, diagnostics information, and motor data.

## 8.2 CAN communication phases / network management (NMT)

Regarding communication, the controller knows the following statuses:

| Status            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| "Initialisation"  | <p>After the controller is switched on, the initialisation phase is run through. During this phase, the controller is not involved in the data exchange on the bus.</p> <p>A part of the initialisation or the complete initialisation can be run through again in every NMT status by transmitting different telegrams (see "state transition"). Here, all parameters are written with their set values.</p> <p>After completing the initialisation, the controller automatically adopts the "Pre-Operational" status.</p> |
| "Pre-Operational" | <p>The controller can receive parameter data.</p> <p>The process data are ignored.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| "Operational"     | The controller can receive parameter data and process data.                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| "Stopped"         | Only network management telegrams can be received.                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

### Network management (NMT)

The telegram structure used for the network management contains the identifier and the command included in the user data which consists of the command byte and the node address.

Telegrams with the identifier "0" and 2 byte user data are used to change between the different communication phases.

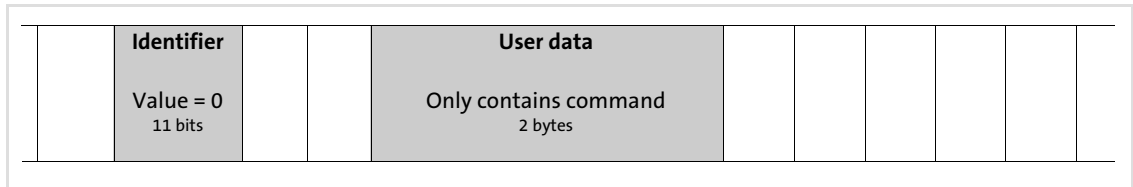


Fig. 8-2 Telegram for switching over the communication phases

A bus node, the network master, carries out the changes between the communication phases for the whole network.

With code **C1852 / C2352** (📖 125) you can set up the communication module for master or slave operation.

After an adjustable boot-up time in **C1856/1 / C2356/1** in master operation, (📖 130) the NMT command *Start\_Remote\_Node* is transmitted, which puts all nodes into the "Operational" status.



#### Note!

Only a change to "Operational" status enables communication via the process data!

#### Example:

If all nodes connected to the bus are to be switched from the "Pre-Operational" communication status to the "Operational" communication status via the CAN master, the identifier and the user data must have the following values in the transmission telegram:

- ▶ Identifier: 0x00 (broadcast telegram)
- ▶ User data: 0x0100

### State transitions

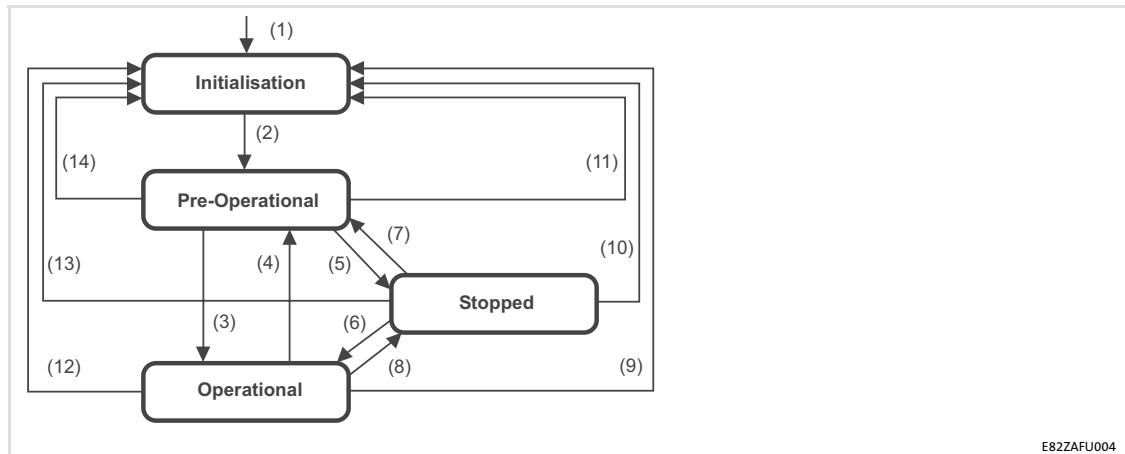


Fig. 8-3 State transitions in CAN network (NMT)

| Status transition                                                                                                                                    | Command (hex) | Network status after change | Effects on process and parameter data after the status change                                                                                                                                                                                                                                                                        |
|------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (1)                                                                                                                                                  | -             | Initialisation              | Initialisation starts automatically when the mains is switched on.<br>During initialisation, the drive does not take part in the data transfer.<br>After initialisation has been completed, the node sends a boot-up message with an individual identifier on the CAN bus and changes automatically to the status "Pre-operational". |
| (2)                                                                                                                                                  | -             | Pre-operational             | In this phase, the master determines the way in which the controller takes part in the communication.                                                                                                                                                                                                                                |
| From that moment on, the master changes the states for the whole network. A target address, which is part of the command, specifies the receiver(s). |               |                             |                                                                                                                                                                                                                                                                                                                                      |
| (3), (6)                                                                                                                                             | 0x01yy        | Operational                 | Network management telegrams, sync, emergency, process data (PDOs) and parameter data (SDOs) active (like "Start remote node")<br>Optional:<br>When the status is changed, event and time-controlled process data (PDOs) will be sent once.                                                                                          |
| (4), (7)                                                                                                                                             | 0x80yy        | Pre-operational             | Network management telegrams, sync, emergency and parameter data (SDOs) active (like "Enter pre-operational state")                                                                                                                                                                                                                  |
| (5), (8)                                                                                                                                             | 0x02yy        | Stopped                     | Only network management telegrams can be received.                                                                                                                                                                                                                                                                                   |
| (9)                                                                                                                                                  | 0x81yy        | Initialisation              | Initialises all parameters in the communication module with the stored values (corresponds to "Reset node")                                                                                                                                                                                                                          |
| (10)                                                                                                                                                 |               |                             |                                                                                                                                                                                                                                                                                                                                      |
| (11)                                                                                                                                                 |               |                             |                                                                                                                                                                                                                                                                                                                                      |
| (12)                                                                                                                                                 |               |                             |                                                                                                                                                                                                                                                                                                                                      |
| (13)                                                                                                                                                 | 0x82yy        | Initialisation              | Initialises communication-relevant parameters (CIA DS 301) in the communication module with the stored values (corresponds to "Reset communication")                                                                                                                                                                                 |
| (14)                                                                                                                                                 |               |                             |                                                                                                                                                                                                                                                                                                                                      |

yy = 00

yy = node ID

In case of this assignment, the telegram addresses all devices connected. The state of all devices can be changed at the same time.  
If a node address is given, only the state of the device with the corresponding address will be changed.

## Process data transfer

### Agreements

- ▶ Process data telegrams between host (master) and controller (slave) are distinguished as follows with regard to their direction:
  - Process data telegrams **to** the controller
  - Process data telegrams **from** the controller
- ▶ In CANopen, the process data objects are named from the node's view:
  - RPDOx: A process data object received by a node
  - TPDOx: A process data object transmitted by a node

### 9.1

#### Available process data objects

Depending on the basic device used, up to 3 RPDOs and TPDOs can be available.



#### Tip!

From the masters's view, the following terms are used for process data transfer with the standard device and the plugged communication module:

- ▶ The master sends the process data output words (POWs) as process output data to the standard device
- ▶ The master receives the process data input words (PIWs) as process input data from the standard device.

### Process data telegram to the controller (RPDO)

The identifier of the process data telegram includes the node address of the controller. The telegram has a maximum user data length of 8 bytes. This chapter describes which user data will be evaluated for the controllers.

The CAN bus is connected to the automation interface X1 via the communication module. X1 is connected to the function block AIF-IN. Here, the user data is transformed into corresponding signal types in order to use them for further function blocks. The control word is especially important for the controller. It contains the drive setpoint in user data bytes 1 and 2.

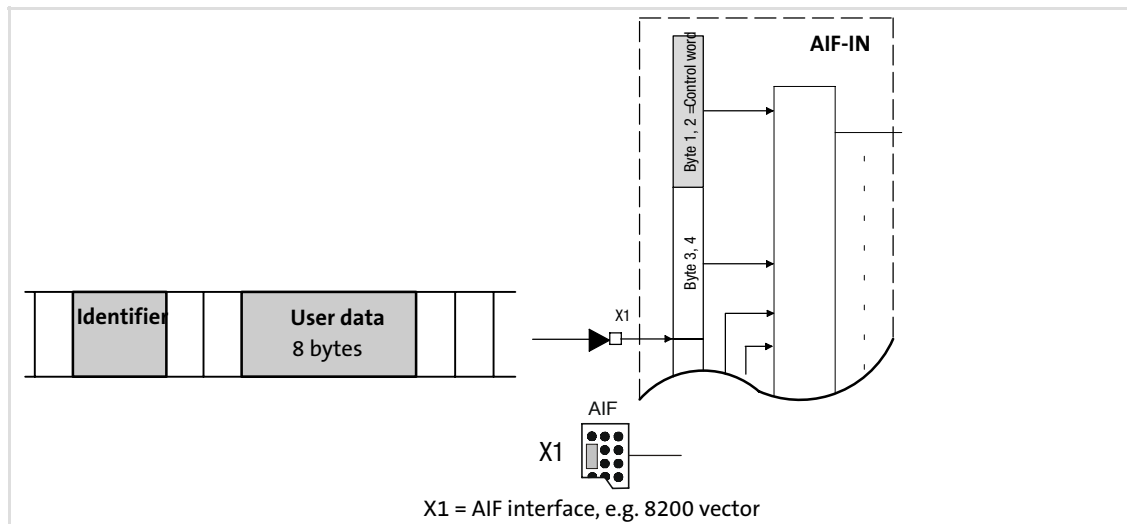


Fig. 9-1 Example: Process data telegram to controller

### Process data telegram from the controller (TPDO)

For the cyclic process data telegrams **from** the controller, the function block to be used is called AIF-OUT. The status word (bytes 1 and 2) included in the process data telegram is transmitted on the CAN bus to the master via this function block.

## 9.2

**Configuring process data channel****Selection of the setpoint source****Drive controller 8200 vector**

The selection of the setpoint source for these controllers is determined with code C0001 (index: 0x5FFE). For process data evaluation the code C0001 must be set to 3 when operating the controller with the communication module.

The setpoint source is the process channel which overwrites the frequency setpoint (C0046) and the control word (C0135) (see documentation for 8200 vector).

**Note!**

Please observe that the setpoint source selection (C0001) must be set equally in all parameter sets.

**Drive controller 93XX**

The 9300 controllers do 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 himself 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 (code C0005) define which source (terminal, keyboard, communication module) overwrites the frequency setpoint and the control word.

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

**9300 system manual**

Here, you can find more detailed information.

**PLC devices: 9300 Servo PLC / Drive PLC / ECSxA**

For communicating via a communication module (e.g. EMF2178IB), it is necessary that the system blocks AIF-IN/OUT1 ... 3 and, if required, the AIF management are integrated into the control configuration of the IEC61131 project.

**Axis modules of the ECS series****Operating instructions for the axis modules of the ECS series**

Here you can find detailed information on the process data configuration.

### 9.3 Cyclic process data objects

#### Synchronisation of cyclic process data

The "sync telegram" is used to ensure that the process data can be cyclically read by the controller and will be accepted by the controller.

The sync telegram is the trigger point for accepting data in the controller and activates the sending process from the controller. For cyclic process data processing, the sync telegram must be generated accordingly.

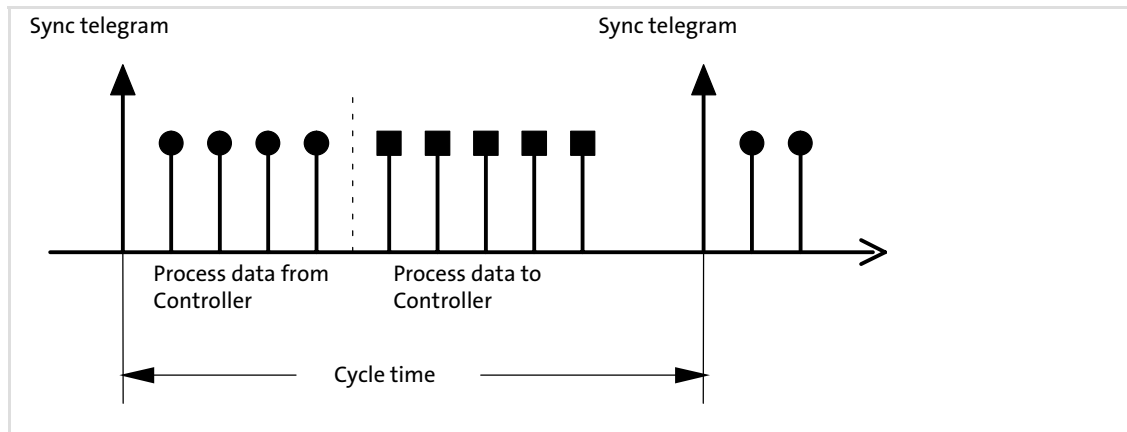


Fig. 9-2 Sync telegram

After the sync telegram has been received, the controllers send the cyclic process data to the master. The master reads them as process input data.

When the sending process has been completed, the controllers receive the process output data (from the master).

All other telegrams (e. g. parameters or event-controlled process data) are acyclically accepted by the controllers after transmission has been completed.

Acyclic data have not been taken into account for the above figure. When selecting the cycle time, they must be considered.

### 9.3.1 Process data signals of Lenze controllers

#### 9.3.1.1 Process data signals for 8200 vector frequency inverters

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



#### Note!

Frequency and speed values are normalised with  $\pm 24000 \equiv \pm 480 \text{ Hz}$ .

#### Process data telegram to drive

| User data (up to 8 bytes) |                           |                       |                        |                       |                        |        |        |
|---------------------------|---------------------------|-----------------------|------------------------|-----------------------|------------------------|--------|--------|
| Byte 1                    | Byte 2                    | Byte 3                | Byte 4                 | Byte 5                | Byte 6                 | Byte 7 | Byte 8 |
| Control word<br>Low byte  | Control word<br>High byte | AIF-IN.W1<br>Low byte | AIF-IN.W1<br>High byte | AIF-IN.W2<br>Low byte | AIF-IN.W2<br>High byte | xx     | xx     |

Byte description:

| Byte  | Content                          | Description                                                             |
|-------|----------------------------------|-------------------------------------------------------------------------|
| 1     | C0135<br>Control word, low byte  | Bits 0 ... 7 of the control word under C0135 (□ 145) are entered here.  |
| 2     | C0135<br>Control word, high byte | Bits 8 ... 15 of the control word under C0135 (□ 145) are entered here. |
| 3     | AIF-IN.W1, low byte              | AIF-IN.Wx is parameterised under code C0412.                            |
| 4     | AIF-IN.W1, high byte             |                                                                         |
| 5     | AIF-IN.W2, low byte              |                                                                         |
| 6     | AIF-IN.W2, high byte             |                                                                         |
| 7 / 8 | xx                               | No evaluation of these data, any content possible                       |

#### Process data telegram from drive

| User data (up to 8 bytes) |                          |                        |                         |                        |                         |        |        |
|---------------------------|--------------------------|------------------------|-------------------------|------------------------|-------------------------|--------|--------|
| Byte 1                    | Byte 2                   | Byte 3                 | Byte 4                  | Byte 5                 | Byte 6                  | Byte 7 | Byte 8 |
| Status word<br>Low byte   | Status word<br>High byte | AIF-OUT.W1<br>Low byte | AIF-OUT.W1<br>High byte | AIF-OUT.W2<br>Low byte | AIF-OUT.W2<br>High byte | xx     | xx     |

Byte description:

| Byte  | Content                         | Explanation                                                            |
|-------|---------------------------------|------------------------------------------------------------------------|
| 1     | C0150<br>Status word, low byte  | Bits 0 ... 7 of the status word are entered here under C0150 (□ 145).  |
| 2     | C0150<br>Status word, high byte | Bits 8 ... 15 of the status word are entered here under C0150 (□ 145). |
| 3     | AIF-OUT.W1<br>Low byte          | AIF-OUT.Wx is parameterised under code C0421.                          |
| 4     | AIF-OUT.W1<br>High byte         |                                                                        |
| 5     | AIF-OUT.W2<br>Low byte          |                                                                        |
| 6     | AIF-OUT.W2<br>High byte         |                                                                        |
| 7 / 8 | xx                              | No evaluation of these data, any content possible                      |

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

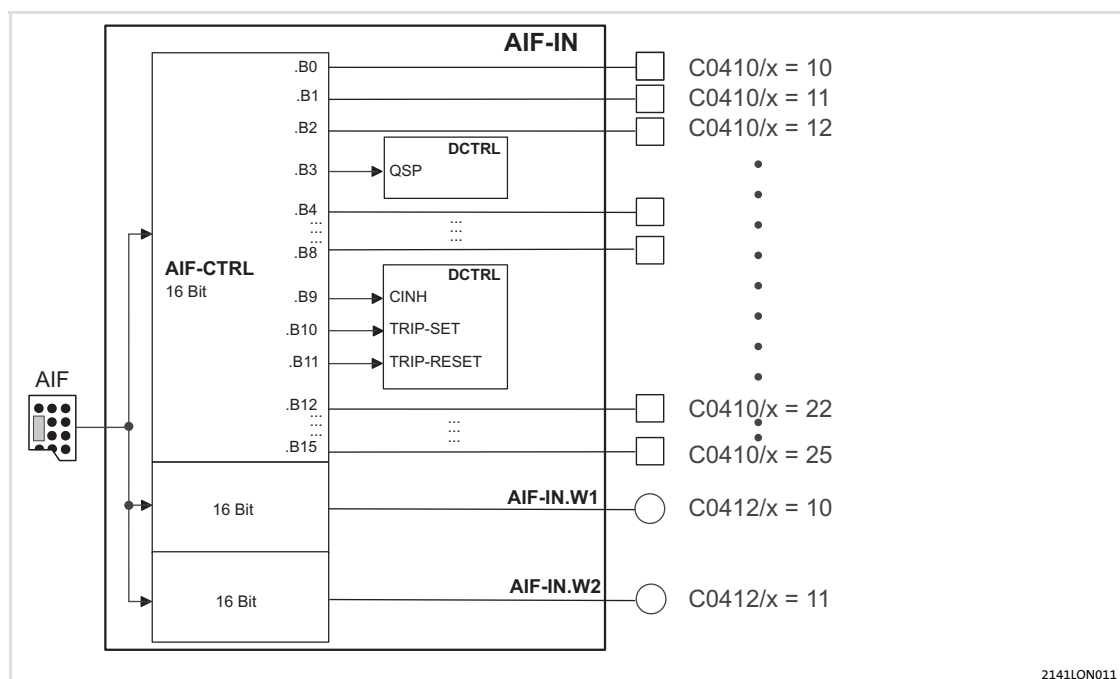


Fig. 9-3 System block AIF-IN in 8200 vector (freely configurable assignment)

### Device status word AIF-STAT for 8200 vector (C0150, index 0x5F69)

| Bit      | Assignment (Lenze setting)                                                                  | Set under C0417/... |    |   |                                              |                               |
|----------|---------------------------------------------------------------------------------------------|---------------------|----|---|----------------------------------------------|-------------------------------|
| 0        | <b>Current parameter set</b> (DCTRL1-PAR-B0)                                                | 1                   |    |   |                                              |                               |
| 1        | <b>Pulse inhibit</b> (DCTRL1-IMP)                                                           | 2                   |    |   |                                              |                               |
| 2        | <b>I<sub>max</sub> limit</b> (MCTRL1-IMAX)                                                  | 3                   |    |   |                                              |                               |
| 3        | <b>Output frequency = frequency setpoint</b> (MCTRL1-RFG1=NOUT)                             | 4                   |    |   |                                              |                               |
| 4        | <b>Ramp function generator input = ramp function generator output 1</b><br>(NSET1-RFG1-I=0) | 5                   |    |   |                                              |                               |
| 5        | <b>Q<sub>min</sub> threshold</b> (PCTRL1-QMIN)                                              | 6                   |    |   |                                              |                               |
| 6        | <b>Output frequency = 0</b> (DCTRL1-NOUT=0)                                                 | 7                   |    |   |                                              |                               |
| 7        | <b>Controller inhibit</b> (DCTRL1-CINH)                                                     | 8                   |    |   |                                              |                               |
| 8 ... 11 | <b>Device status</b> (DCTRL1-Stat*1 ... STAT*8)                                             | Reserved            |    |   |                                              |                               |
| Bit      | 11                                                                                          |                     | 10 | 9 | 8                                            |                               |
|          | 0                                                                                           |                     | 0  | 0 | 0                                            | Device initialisation         |
|          | 0                                                                                           |                     | 0  | 1 | 0                                            | Switch-on inhibit             |
|          | 0                                                                                           |                     | 0  | 1 | 1                                            | Operation inhibited           |
|          | 0                                                                                           |                     | 1  | 0 | 0                                            | Flying restart circuit active |
|          | 0                                                                                           |                     | 1  | 0 | 1                                            | DC injection brake active     |
|          | 0                                                                                           |                     | 1  | 1 | 0                                            | Operation enabled             |
|          | 0                                                                                           |                     | 1  | 1 | 1                                            | Message active                |
|          | 1                                                                                           |                     | 0  | 0 | 0                                            | Fault active                  |
|          | 1                                                                                           | 1                   | 1  | 1 | Communication with basic device not possible |                               |
| 12       | <b>Overtemperature warning</b> (DCTRL1-OH-WARN)                                             | 13                  |    |   |                                              |                               |
| 13       | <b>DC-bus overvoltage</b> (DCTRL1-OV)                                                       | 14                  |    |   |                                              |                               |
| 14       | <b>Direction of rotation</b> (DCTRL1-CCW)                                                   | 15                  |    |   |                                              |                               |
| 15       | <b>Ready for operation</b> (DCTRL1-RDY)                                                     | 16                  |    |   |                                              |                               |

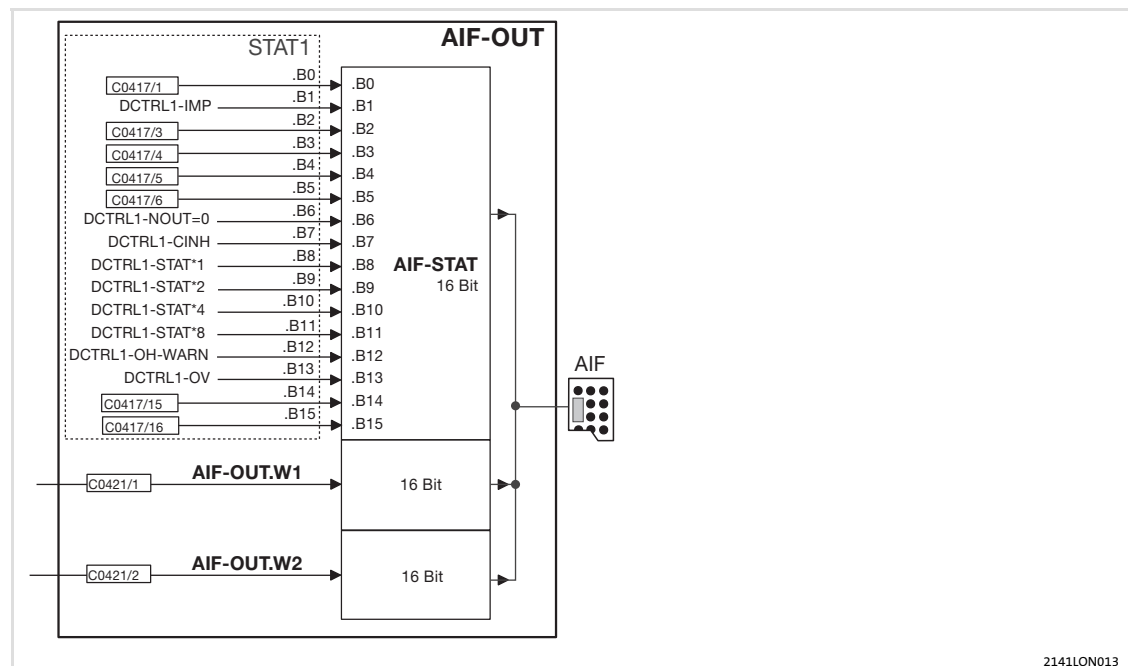


Fig. 9-4 System block AIF-OUT in 8200 vector (freely programmable assignment)

### 9.3.1.2 Process data signals for 9300 servo inverter

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



#### 9300 system manuals

Here you can find detailed information on the function blocks and signal configurations.

#### Function block AIF-IN

The function block AIF-IN determines the input data of the controller as data interface for the EMF2178IB communication module.

#### Process data telegram to drive

| User data (up to 8 bytes)       |                                  |                              |                               |                              |                               |                              |                               |
|---------------------------------|----------------------------------|------------------------------|-------------------------------|------------------------------|-------------------------------|------------------------------|-------------------------------|
| Byte 1                          | 2nd byte                         | 3rd byte                     | Byte 4                        | Byte 5                       | Byte 6                        | 7th byte                     | Byte 8                        |
| <b>Control word</b><br>Low byte | <b>Control word</b><br>High byte | <b>AIF-IN.W1</b><br>Low byte | <b>AIF-IN.W1</b><br>High byte | <b>AIF-IN.W2</b><br>Low byte | <b>AIF-IN.W2</b><br>High byte | <b>AIF-IN.W3</b><br>Low byte | <b>AIF-IN.W3</b><br>High byte |
| 57                              |                                  |                              |                               |                              |                               |                              |                               |

AIF-IN.W1 to AIF-IN.W3 depend on the signal configuration selected under code C0005. Under code C0005 you can preconfigure the signals of the control word and the status word.

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 93XX System Manual is used.

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

### Control word for 93XX

| 9300  | 9300 servo inverter |                     |                     |                     | 9300 position controller | 9300 cam profiler   | 9300 vector                          |                     |                     |
|-------|---------------------|---------------------|---------------------|---------------------|--------------------------|---------------------|--------------------------------------|---------------------|---------------------|
| C0005 | 1xx3                | 4xx3                | 5xx3                | 6xx3, 7xx3          | 2xx3                     | 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    | DCTRL-PAR-LOAD      | DCTRL-PAR-LOAD      | DCTRL-PAR-LOAD      | DCTRL-PAR-LOAD      | POS-PARAM-RD             | Not assigned        | DCTRL-PAR-LOAD                       | DCTRL-PAR-LOAD      | DCTRL-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 individual bit-control commands of the control word depend on other bit positions.

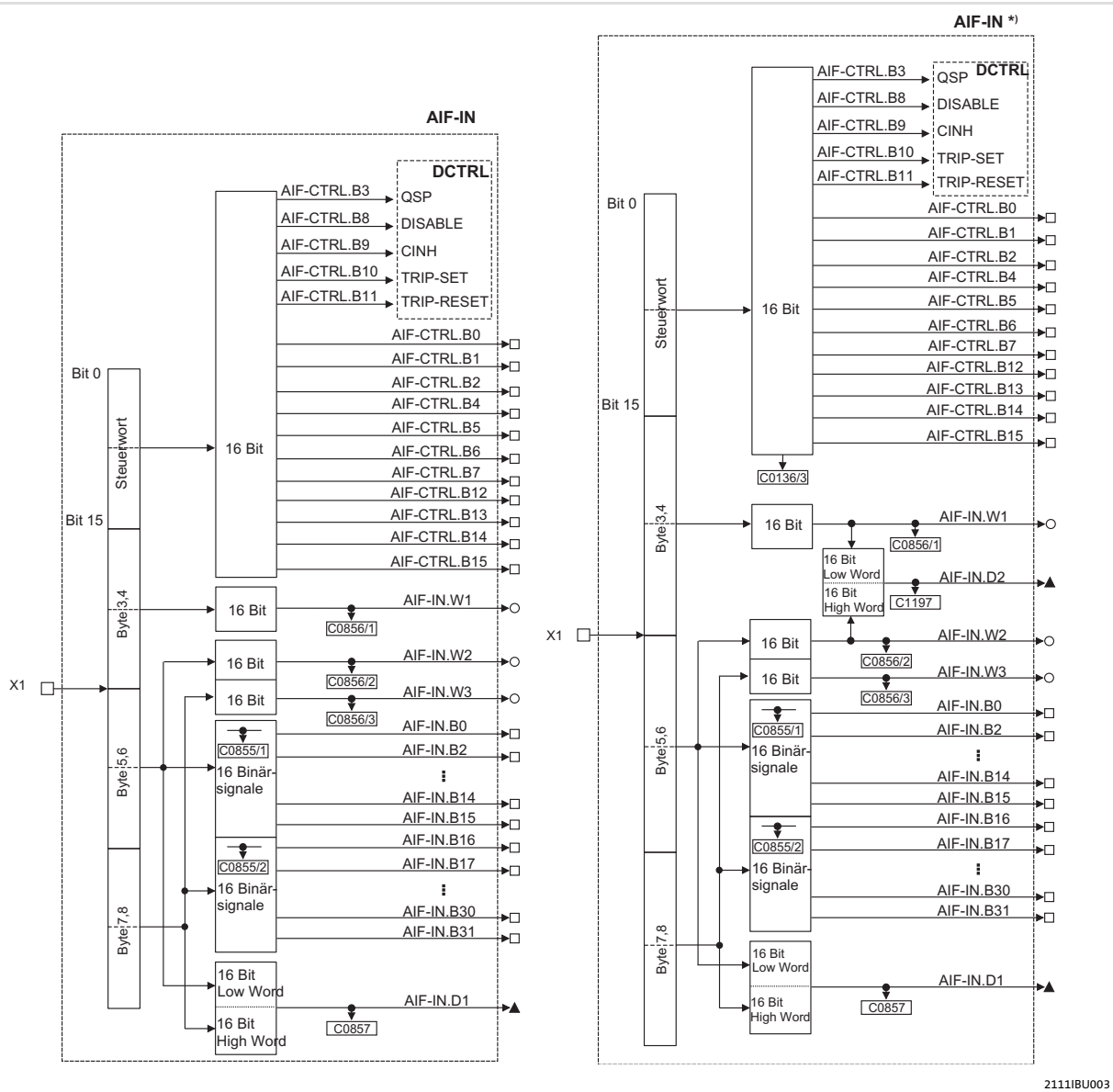


Fig. 9-5

Function blocks AIF-IN and AIF-IN<sup>\*</sup>)

AIF-IN<sup>\*</sup>) is available for the 9300 technology variants servo inverter, positioning controller and cam profiler as of software version 2.0. AIF-IN.D2 is new.

### Function block AIF-OUT

The function block AIF-OUT determines the output data of the controller as data interface for the EMF2178IB communication module.

### Process data telegram from drive

| User data (up to 8 bytes) |                          |                        |                         |                        |                         |                        |                         |
|---------------------------|--------------------------|------------------------|-------------------------|------------------------|-------------------------|------------------------|-------------------------|
| Byte 1                    | Byte 2                   | Byte 3                 | Byte 4                  | Byte 5                 | Byte 6                  | Byte 7                 | Byte 8                  |
| Status word<br>Low byte   | Status word<br>High byte | AIF-OUT.W1<br>Low byte | AIF-OUT.W1<br>High byte | AIF-OUT.W2<br>Low byte | AIF-OUT.W2<br>High byte | AIF-OUT.W3<br>Low byte | AIF-OUT.W3<br>High byte |
| 60                        |                          |                        |                         |                        |                         |                        |                         |

AIF-OUT.W1 to AIF-OUT.W3 depend on the signal configuration selected under code C0005. Under code C0005 you can preconfigure the signals of the control word and the status word.

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 93XX System Manual is used.

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

### Status word for 93XX

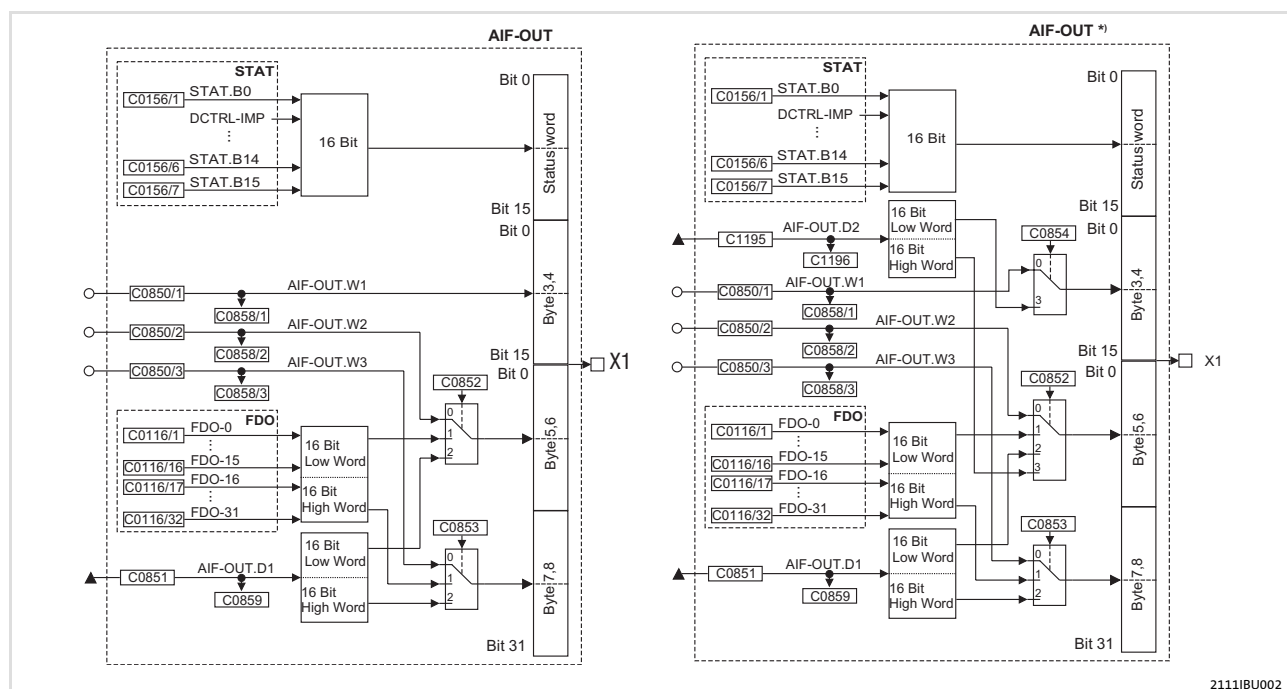
[illegible]

Fig. 9-6 Function blocks AIF-OUT and AIF-OUT\*)  
AIF-OUT\*) is available for the 9300 technology variants servo inverter, positioning controller and cam profiler as of software version 2.0. AIF-OUT.D2 is new.

### 9.3.1.3 Process data signals for 9300 Servo PLC, Drive PLC, ECSxA



For more information ...

please see the chapter "System blocks, AIF1\_IO\_AutomationInterface" in the Manuals and Operating Instructions for the 9300 Servo PLC, Drive PLC and ECSxA axis module.



#### **Note!**

##### **Only 9300 Servo PLC & ECSxA:**

Perform the following linkings in the PLC program of the drive controller:

- ▶ AIF1\_wDctrlCtrl → DCTRL\_wAIF1Ctrl
- ▶ DCTRL\_wStat → AIF1\_wDctrlStat

## Process data telegram to drive

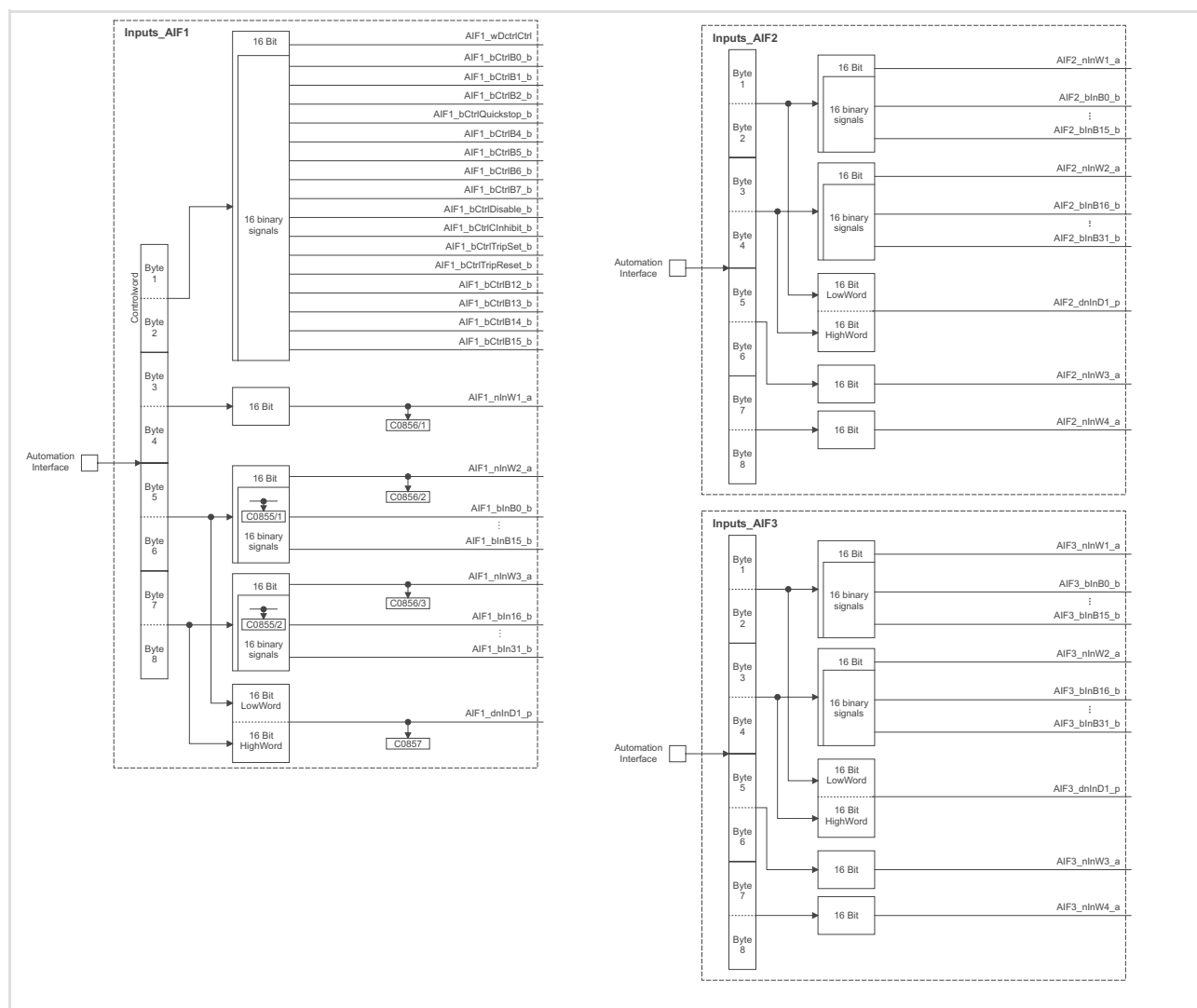


Fig. 9-7 Function blocks AIF1-IN, AIF2-IN and AIF3-IN

## Process data telegram from drive

The following data can be assigned to the PIW data (TPDOs):

| Designation / variable name         | Explanation       |
|-------------------------------------|-------------------|
| Device status word (AIF1_DctrlStat) | AIF word 1        |
| AIF_nOutW1_a                        | AIF word 2        |
| AIF_nOutW2_a                        | AIF word 3        |
| AIF_nOutW3_a                        | AIF word 4        |
| AIF2_nOutW1_a                       | AIF word 5        |
| AIF2_nOutW2_a                       | AIF word 6        |
| AIF2_nOutW3_a                       | AIF word 7        |
| AIF2_nOutW4_a                       | AIF word 8        |
| AIF3_nOutW1_a                       | AIF word 9        |
| AIF3_nOutW2_a                       | AIF word 10       |
| AIF3_nOutW3_a                       | AIF word 11       |
| AIF3_nOutW4_a                       | AIF word 12       |
| AIF1_dnOutD1_p                      | AIF double word 1 |

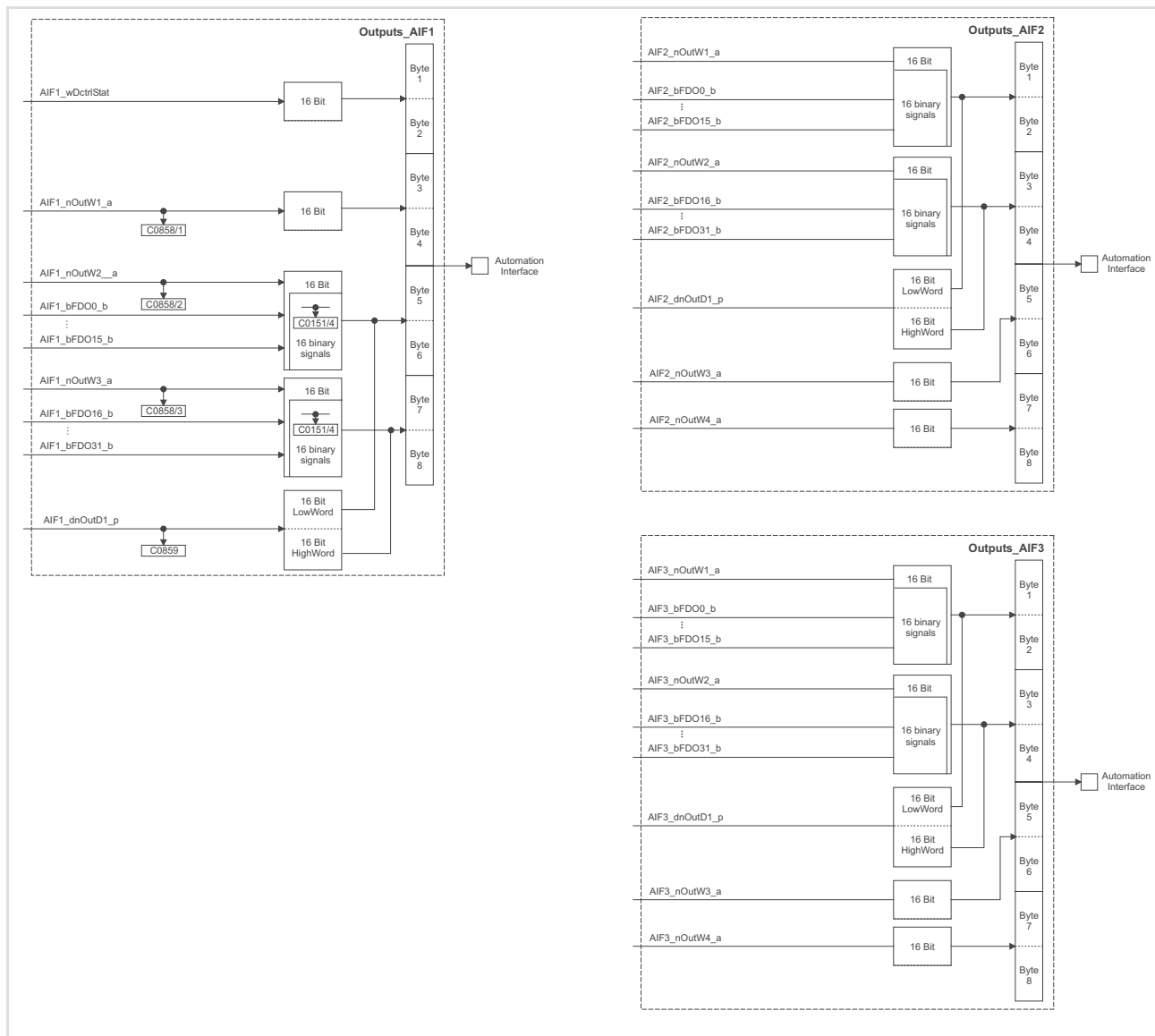


Fig. 9-8 Function blocks AIF1-OUT, AIF2-OUT and AIF3-OUT

## Axis modules of the ECS series



## Operating instructions for the axis modules of the ECS series

Here you can find detailed information on the process data configuration.

### 9.3.2 Mapping in CANopen objects (I-160x, I-1A0x)

#### General



#### Note!

Mapping is not possible in communication modules used together with old devices of the 8200 series.

The CANopen objects I-160x and I-1A0x are used to assign the CANopen process data to the internal data. In the EMF2178IB communication module an assignment to the process data of the internal AIF interface is possible. The process data volume and type depends on the standard device.

The process data length depends on the AIF mode (68):

| Basic device   | AIF mode                                               |
|----------------|--------------------------------------------------------|
| 8200 vector    | Mode 3: 3 process data words with variable assignment  |
| 93XX           | Mode 4: 4 process data words with variable assignment  |
| 9300 servo PLC | Mode 5: 12 process data words with variable assignment |
| ECSXX          | depends on the variant (mode 4 or 5)                   |

For mapping in the module, the master enters indices and data lengths.

#### Mapping structure

| MSB   |     |    |    |          |   |   |                                                | LSB |  |
|-------|-----|----|----|----------|---|---|------------------------------------------------|-----|--|
| 31    | ... | 16 | 15 | ...      | 8 | 7 | ...                                            | 0   |  |
| Index |     |    |    | Subindex |   |   | Object length<br>DWord 0 = 0x20<br>Word = 0x10 |     |  |

The EDS file for the EMF2178IB communication module supports the mapping.



#### Tip!

The current EDS file required for configuring the EMF2178IB (CANopen) communication module can be found in the download area on:

**[www.Lenze.com](http://www.Lenze.com)**

**AIF data image in codes**

The AIF process data images are mapped to the following codes:

| AIF process data                             | Code           | Index [hex] |
|----------------------------------------------|----------------|-------------|
| Process input data AIF-IN as 16-bit values   | C1822/1 ... 12 | 0x58E1      |
| Process output data AIF-OUT as 16-bit values | C1823/1 ... 12 | 0x58E0      |
| Process input data AIF-IN as 32-bit values   | C1824/1 ... 11 | 0x58DF      |
| Process output data AIF-OUT as 32-bit values | C1825/1 ... 11 | 0x58DE      |

The codes represent the values at the AIF interface. Converted into indices, mapping refers to these codes.

| Process data AIF-IN |                                   | Represented as |                             |
|---------------------|-----------------------------------|----------------|-----------------------------|
|                     |                                   | 16-bit values  | 32-bit values <sup>1)</sup> |
| Byte 1              | <b>AIF1-IN</b><br>(bytes 1 ... 8) | C1822/1        | [C1824/1]                   |
| 2nd byte            |                                   |                |                             |
| 3rd byte            |                                   | C1822/2        |                             |
| Byte 4              |                                   |                |                             |
| Byte 5              |                                   | C1822/3        | C1824/2                     |
| Byte 6              |                                   |                |                             |
| 7th byte            |                                   | C1822/4        |                             |
| Byte 8              |                                   |                |                             |
| Byte 9              | <b>AIF2-IN</b><br>(bytes 1 ... 8) | C1822/5        | [C1824/4]                   |
| Byte 10             |                                   |                |                             |
| Byte 11             |                                   | C1822/6        |                             |
| Byte 12             |                                   |                |                             |
| Byte 13             |                                   | C1822/7        | [C1824/6]                   |
| Byte 14             |                                   |                |                             |
| Byte 15             |                                   | C1822/8        |                             |
| Byte 16             |                                   |                |                             |
| Byte 17             | <b>AIF3-IN</b><br>(bytes 1 ... 8) | C1822/9        | [C1824/8]                   |
| Byte 18             |                                   |                |                             |
| Byte 19             |                                   | C1822/10       |                             |
| Byte 20             |                                   |                |                             |
| Byte 21             |                                   | C1822/11       | [C1824/10]                  |
| Byte 22             |                                   |                |                             |
| Byte 23             |                                   | C1822/12       |                             |
| Byte 24             |                                   |                |                             |

<sup>1)</sup> [Cxxx/y] = Codes not relevant for the AIF modes (📖 68)

| Process data AIF-OUT |                                    | Represented as |                             |
|----------------------|------------------------------------|----------------|-----------------------------|
|                      |                                    | 16-bit values  | 32-bit values <sup>1)</sup> |
| Byte 1               | <b>AIF1-OUT</b><br>(bytes 1 ... 8) | C1823/1        | [C1825/1]                   |
| 2nd byte             |                                    |                |                             |
| 3rd byte             |                                    | C1823/2        |                             |
| Byte 4               |                                    |                |                             |
| Byte 5               |                                    | C1823/3        | C1825/2                     |
| Byte 6               |                                    |                |                             |
| 7th byte             |                                    | C1823/4        | C1825/3                     |
| Byte 8               |                                    |                |                             |
| Byte 9               | <b>AIF2-OUT</b><br>(bytes 1 ... 8) | C1823/5        | [C1825/4]                   |
| Byte 10              |                                    |                |                             |
| Byte 11              |                                    | C1823/6        |                             |
| Byte 12              |                                    |                |                             |
| Byte 13              |                                    | C1823/7        | [C1825/6]                   |
| Byte 14              |                                    |                |                             |
| Byte 15              |                                    | C1823/8        | [C1825/7]                   |
| Byte 16              |                                    |                |                             |
| Byte 17              | <b>AIF3-OUT</b><br>(bytes 1 ... 8) | C1823/9        | [C1825/8]                   |
| Byte 18              |                                    |                |                             |
| Byte 19              |                                    | C1823/10       |                             |
| Byte 20              |                                    |                |                             |
| Byte 21              |                                    | C1823/11       | [C1825/10]                  |
| Byte 22              |                                    |                |                             |
| Byte 23              |                                    | C1823/12       | [C1825/11]                  |
| Byte 24              |                                    |                |                             |

<sup>1)</sup> [Cxxx/y] = Codes not relevant for the AIF modes (📖 68)

### CANopen indices for mapping

According to the CANopen specifications, the indices I-160x and I-1A0x are available for mapping the CAN PDOs. During mapping, the process output data of the master are mapped via the indices I-160x to the AIF input data of the basic device. The indices I-160x refer to codes C1822 and C1824. The entry contains the index of the corresponding code and the corresponding subindex for the assignment of a word or double word. The object length must be indicated accordingly.

Under the indices I-1A0x, the indices of code C1823 must be entered for words or the indices of code C1825 for double words.

### Device-internal mechanisms

The user must know the AIF interface assignment to configure the mapping accordingly in the communication module. All data of the AIF interface can be placed on any position of the CAN PDOs. The maximum scope of the AIF interface is shown in the codes. The available values depend on the basic device and on the application in the basic device.

## AIF interface assignment / AIF modes

| Process data |                                | Mode 3                        | Mode 4                    |                           |                                 | Mode 5                             |                                   |  |
|--------------|--------------------------------|-------------------------------|---------------------------|---------------------------|---------------------------------|------------------------------------|-----------------------------------|--|
| Byte 1       | AIF1-IN/OUT<br>(bytes 1 ... 8) | Control word /<br>status word | AIF-CTRL /<br>AIF-STAT    |                           |                                 | AIF_wDctrlCtrl /<br>AIF_wDctrlStat |                                   |  |
| 2nd byte     |                                |                               |                           |                           |                                 |                                    |                                   |  |
| 3rd byte     |                                |                               | AIF-IN.W1 /<br>AIF-OUT.W1 | AIF-IN.W1 /<br>AIF-OUT.W1 |                                 | AIF-IN.D2 /<br>AIF-OUT.D2          | AIF1_nInW1_a /<br>AIF1_nOutW1_a   |  |
| Byte 4       |                                |                               |                           |                           |                                 |                                    |                                   |  |
| Byte 5       |                                | AIF-IN.W2 /<br>AIF-OUT.W2     | AIF-IN.W2 /<br>AIF-OUT.W2 | AIF-IN.D1 /<br>AIF-OUT.D1 | AIF1_nInW2_a /<br>AIF1_nOutW2_a |                                    | AIF1_dnInD1_p /<br>AIF1_dnOutD1_p |  |
| Byte 6       |                                |                               |                           |                           |                                 |                                    |                                   |  |
| 7th byte     |                                |                               |                           |                           | AIF1_nInW3_a /<br>AIF1_nOutW3_a |                                    |                                   |  |
| Byte 8       |                                |                               |                           |                           |                                 |                                    |                                   |  |
| Byte 9       | AIF2-IN/OUT<br>(bytes 1 ... 8) |                               |                           |                           |                                 | AIF2_nInW1_a /<br>AIF2_nOutW1_a    | AIF2_dnInD1_p /<br>AIF2_dnOutD1_p |  |
| Byte 10      |                                |                               |                           |                           |                                 |                                    |                                   |  |
| Byte 11      |                                |                               |                           |                           |                                 | AIF2_nInW2_a /<br>AIF2_nOutW2_a    |                                   |  |
| Byte 12      |                                |                               |                           |                           |                                 |                                    |                                   |  |
| Byte 13      |                                |                               |                           |                           |                                 | AIF2_nInW3_a /<br>AIF2_nOutW3_a    |                                   |  |
| Byte 14      |                                |                               |                           |                           |                                 |                                    |                                   |  |
| Byte 15      |                                |                               |                           |                           |                                 | AIF2_nInW4_a /<br>AIF2_nOutW4_a    |                                   |  |
| Byte 16      |                                |                               |                           |                           |                                 |                                    |                                   |  |
| Byte 17      | AIF3-IN/OUT<br>(bytes 1 ... 8) |                               |                           |                           |                                 | AIF3_nInW1_a /<br>AIF3_nOutW1_a    | AIF3_dnInD1_p /<br>AIF3_dnOutD1_p |  |
| Byte 18      |                                |                               |                           |                           |                                 |                                    |                                   |  |
| Byte 19      |                                |                               |                           |                           |                                 | AIF3_nInW2_a /<br>AIF3_nOutW2_a    |                                   |  |
| Byte 20      |                                |                               |                           |                           |                                 |                                    |                                   |  |
| Byte 21      |                                |                               |                           |                           |                                 | AIF3_nInW3_a /<br>AIF3_nOutW3_a    |                                   |  |
| Byte 22      |                                |                               |                           |                           |                                 |                                    |                                   |  |
| Byte 23      |                                |                               |                           |                           |                                 | AIF3_nInW4_a /<br>AIF3_nOutW4_a    |                                   |  |
| Byte 24      |                                |                               |                           |                           |                                 |                                    |                                   |  |

- For mode 3 (e.g. for the 8200 vector), an assignment of the first three words is useful. For the mapping, the index of code C1822 should accordingly be used with the subindices 1 to 3.
- For mode 4 (e.g. for the 93XX), an assignment of the first four words or the double words 2 and 3 is possible. Here, the representation of the double words differs from the text description in the basic device.
- For mode 5 (e.g. for the 9300 Servo PLC), all 12 words are possible and the use of double words 3, 5 and 9.
- The user is responsible for the corresponding assignment. The mapping is not checked once again during the entry.
- For a more detailed description of the entries, see indices I-160x (111) and I-1A0x (114).

## 10

## Parameter data transfer

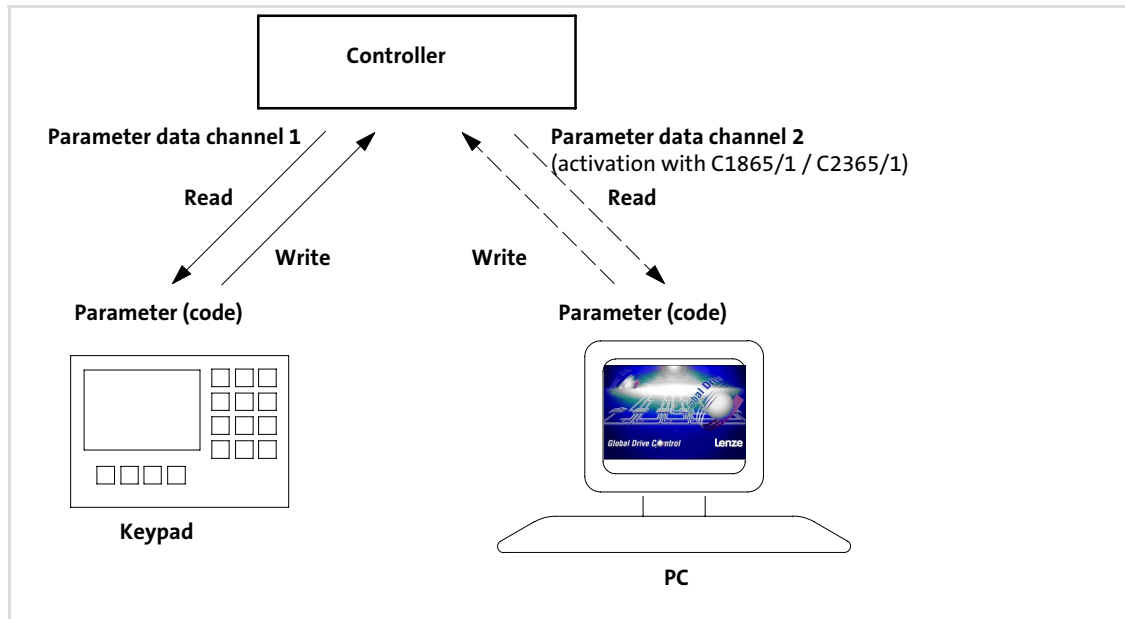


Fig. 10-1 Connection of devices for parameter setting via two parameter data channels

#### Parameters ...

- ▶ are values which are stored under codes in Lenze controllers.
- ▶ are, for instance, used for one-off plant settings or a change of material in a machine.
- ▶ are transferred with low priority.

Parameter data are transmitted via the system bus as SDOs (Service Data Objects) and acknowledged by the receiver. The SDOs enable read and write access to the object directory.

In the Lenze setting, one parameter data channel is available for parameter setting.

A second parameter data channel can be activated via code C1865/1 / C2365/1 or the implemented CANopen object I-1201 in order to enable the simultaneous connection of various devices. The parameter data channel 2 is **deactivated** as default.



#### Note!

When the second parameter data channel is activated the possible number of bus nodes is reduced.

### 10.1 Access to the codes of the controller

When using Lenze communication modules, you can change the features and response characteristics of any controller connected to the network using a higher-level host system (PLC, master).

In Lenze controllers, parameters to be changed are listed under codes.

Controller codes are addressed via the index when accessing the code through the communication module.

The index for Lenze code numbers is between 16576 (0x40C0) and 24575 (0x5FFF).



#### Documentation for the controller

Here you can find a detailed description of the codes.

#### Indexing of Lenze codes

| Conversion formula |                                      |
|--------------------|--------------------------------------|
| Index (dec)        | Index (hex)                          |
| 24575 - Lenze code | 0x5FFF - (Lenze code) <sub>hex</sub> |

| Example for C0001 (operating mode) |                     |
|------------------------------------|---------------------|
| Index (dec)                        | Index (hex)         |
| 24575 - 1 = 24574                  | 0x5FFF - 1 = 0x5FFE |

The parameter value is part of the telegram user data (see examples,  77).

## 10.2 Lenze parameter sets

Parameter sets are used for separately storing codes based on the need for different configurations of application processes.



### Stop!

#### 8200 vector with EMF2178IBcommunication module

- ▶ The cyclic writing of parameter data to the EEPROM is not permissible.
- ▶ If you still want to change parameter data, code C0003 must be set to "0" after every mains switching. The parameter data are not stored in the EEPROM but as volatile data.

### 10.2.1 Parameter sets for 8200 vector controller

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

#### Addressing

Addressing is carried out with a code offset:

- ▶ Offset "0" addresses the parameter set 1 with the codes C0000 ... C1999.
- ▶ Offset "2000" addresses the parameter set 2 with the codes C2000 ... C3999.
- ▶ Offset "4000" addresses the parameter set 3 with the codes C4000 ... C5999.
- ▶ Offset "6000" addresses the parameter set 4 with the codes C6000 ... C7999.

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

#### Example

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

- ▶ C0011 in parameter set 1: Code no. = 11
- ▶ C0011 in parameter set 2: Code no. = 2011
- ▶ C0011 in parameter set 3: Code no. = 4011
- ▶ C0011 in parameter set 4: Code no. = 6011



### Note!

Automatic saving of the changed parameter data is activated (Lenze basic setting, can be switched off via C0003).

**10.2.2 Parameter sets for controller 93XX**

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

**Addressing**

- ▶ Only the current parameter set can be directly addressed via the bus.
- ▶ After power-on, parameter set 1 is automatically loaded into the current parameter set.
- ▶ Before changing the parameters of parameter sets 2 ... 4, the parameter sets must be activated.

**Note!**

Changes in the current parameter set are lost after switching off the drive controller.

Code C0003 is used to store the current parameter set (see documentation of the controller).

### 10.3 Structure of the parameter data telegram

| User data (up to 8 bytes) |                   |                    |          |            |           |           |           |
|---------------------------|-------------------|--------------------|----------|------------|-----------|-----------|-----------|
| Byte 1                    | 2nd byte          | 3rd byte           | Byte 4   | Byte 5     | Byte 6    | 7th byte  | Byte 8    |
| Command                   | Index<br>Low byte | Index<br>High byte | Subindex | Data 1     | Data 2    | Data 3    | Data 4    |
|                           |                   |                    |          | Low word   |           | High word |           |
|                           |                   |                    |          | Low byte   | High byte | Low byte  | High byte |
|                           |                   |                    |          | Error code |           |           |           |



#### Note!

User data are displayed in the left-justified INTEL format.  
(Calculation examples: 77)

#### Command

The command contains the services for writing and reading parameters and information about the user data length.

Structure of a command:

|                | Bit 7<br>MSB           | Bit 6 | Bit 5 | Bit 4 | Bit 3  | Bit 2 | Bit 1 | Bit 0<br>LSB |                                                                                                                            |
|----------------|------------------------|-------|-------|-------|--------|-------|-------|--------------|----------------------------------------------------------------------------------------------------------------------------|
| Command        | Command Specifier (cs) |       |       | 0     | Length |       | e     | s            | Note                                                                                                                       |
| Write request  | 0                      | 0     | 1     | 0     | x      | x     | 1     | 1            | User data length coding<br>in bits 2 and 3:<br>• 0b00 = 4 bytes<br>• 0b01 = 3 bytes<br>• 0b10 = 2 bytes<br>• 0b11 = 1 byte |
| Write response | 0                      | 1     | 1     | 0     | x      | x     | 0     | 0            |                                                                                                                            |
| Read request   | 0                      | 1     | 0     | 0     | x      | x     | 0     | 0            |                                                                                                                            |
| Read response  | 0                      | 1     | 0     | 0     | x      | x     | 1     | 1            |                                                                                                                            |
| Error response | 1                      | 0     | 0     | 0     | 0      | 0     | 0     | 0            |                                                                                                                            |

The following information is contained/must be entered in the command.

| Command                                                                    | 4-byte data<br>(bytes 5 ... 8) |     | 2-byte data<br>(bytes 5 and 6) |     | 1-byte data<br>(byte 5) |     | Block                |     |
|----------------------------------------------------------------------------|--------------------------------|-----|--------------------------------|-----|-------------------------|-----|----------------------|-----|
|                                                                            | hex                            | dec | hex                            | dec | hex                     | dec | hex                  | dec |
| Write request<br>(Send parameters to drive)                                | 0x23                           | 35  | 0x2B                           | 43  | 0x2F                    | 47  | Writing not possible |     |
| Write response<br>(Controller response to write request (acknowledgement)) | 0x60                           | 96  | 0x60                           | 96  | 0x60                    | 96  |                      |     |
| Read request<br>(Request to read a parameter from the controller)          | 0x40                           | 64  | 0x40                           | 64  | 0x40                    | 64  | 0x40                 | 64  |
| Read response<br>(Response to the read request with an actual value)       | 0x43                           | 67  | 0x4B                           | 75  | 0x4F                    | 79  | 0x41                 | 65  |
| Error response<br>(Controller indicates a communication error)             | 0x80                           | 128 | 0x80                           | 128 | 0x80                    | 128 | 0x80                 | 128 |

### Index low byte / Index high byte

Parameters and Lenze codes are selected with these two bytes according to the following formula:

$$\text{Index} = 24575 - (\text{Lenze code number} + 2000 (\text{parameter set} - 1))$$

| Example                                                               | Calculation                                                                                            | Low/high byteindex entries                      |
|-----------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|-------------------------------------------------|
| Code C0012 (acceleration time) in parameter set 1 is to be addressed. | $24575 - 12 - 0 = 24563 = 0x5FF3$                                                                      | Index low byte = 0xF3<br>Index high byte = 0x5F |
| Code C0012 (acceleration time) in parameter set 2 is to be addressed. | An offset of "2000" is to be added because of parameter set 2:<br>$24575 - 12 - 2000 = 22563 = 0x5823$ | Index low byte = 0x23<br>Index high byte = 0x58 |

### Subindex

Depending on the corresponding Lenze code, further subcodes may exist.

For all array codes, the value of the highest subindex can be read out via subindex 0.

### Data (Data 1 ... 4)

| Parameter value length depending on the data format<br>(Data format: See "table of attributes" in the documentation of the controller) |           |           |           |
|----------------------------------------------------------------------------------------------------------------------------------------|-----------|-----------|-----------|
| Parameter value<br>(length: 1 byte)                                                                                                    | 0x00      | 0x00      | 0x00      |
| Parameter value (length: 2 bytes)<br>Low byte                                                                                          | High byte | 0x00      | 0x00      |
| Parameter value (length: 4 bytes)                                                                                                      |           |           |           |
| Low word                                                                                                                               |           | High word |           |
| Low byte                                                                                                                               | High byte | Low byte  | High byte |



#### Note!

- Lenze parameters are mainly represented as data type FIX32 (32-bit value with sign and four decimal positions). In order to obtain integer values, the desired parameter value must be multiplied by 10000. (See "Attribute table" in the documentation of the controller.)
- The parameters C0135 and C0150 are transferred as bit code and without factor.



#### Documentation of the controllers

Here you can find the descriptions of the codes (see there: "Code table")

## Error messages

| User data (up to 8 bytes) |                   |                    |          |            |        |          |        |
|---------------------------|-------------------|--------------------|----------|------------|--------|----------|--------|
| Byte 1                    | 2nd byte          | 3rd byte           | Byte 4   | Byte 5     | Byte 6 | 7th byte | Byte 8 |
| Command                   | Index<br>Low byte | Index<br>High byte | Subindex | Error code |        |          |        |

- **Byte 1:**  
Code **128** or **0x80** in the **command** byte indicates that an error has occurred.
- **Bytes 2 ... 4:**  
These bytes contain the **index** (bytes 2 and 3) and the **subindex** (byte 4) of the code in which the error has occurred.
- **Bytes 5 ... 8:**  
The data bytes 5 ... 8 contain the **error code**. The error code is described in reversed direction compared to the read direction.

### Example: Representation of the error code 0x06040041 in bytes 5 ... 8

|                                                                                                                |             |             |             |
|----------------------------------------------------------------------------------------------------------------|-------------|-------------|-------------|
|  Read direction of error code |             |             |             |
| <b>0x41</b>                                                                                                    | <b>0x00</b> | <b>0x04</b> | <b>0x06</b> |
| Byte 5                                                                                                         | Byte 6      | 7th byte    | Byte 8      |
| Low word                                                                                                       |             | High word   |             |
| Low byte                                                                                                       | High byte   | Low byte    | High byte   |

## 10.4

## Error codes

| Error code<br>[hex] | Description                                                                                                                                                                          |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x05030000          | Toggle bit not changed                                                                                                                                                               |
| 0x05040000          | SDO protocol expired                                                                                                                                                                 |
| 0x05040001          | Invalid or unknown client/server command specifier                                                                                                                                   |
| 0x05040002          | Invalid block size (only block mode)                                                                                                                                                 |
| 0x05040003          | Invalid processing number (only block mode)                                                                                                                                          |
| 0x05040004          | CRC error (only block mode)                                                                                                                                                          |
| 0x05040005          | Not enough memory                                                                                                                                                                    |
| 0x06010000          | Object access not supported                                                                                                                                                          |
| 0x06010001          | Try to read writable object                                                                                                                                                          |
| 0x06010002          | Try to write readable object                                                                                                                                                         |
| 0x06020000          | Object not listed in object directory                                                                                                                                                |
| 0x06040041          | Object not mapped to PDO                                                                                                                                                             |
| 0x06040042          | Number and length of objects to be transferred longer than PDO.                                                                                                                      |
| 0x06040043          | General parameter incompatibility                                                                                                                                                    |
| 0x06040047          | General internal controller incompatibility                                                                                                                                          |
| 0x06060000          | Access denied because of hardware error                                                                                                                                              |
| 0x06070010          | Inappropriate data type, service parameter length                                                                                                                                    |
| 0x06070012          | Inappropriate data type, service parameter length exceeded                                                                                                                           |
| 0x06070013          | Inappropriate data type, service parameters not long enough                                                                                                                          |
| 0x06090011          | Subindex does not exist                                                                                                                                                              |
| 0x06090030          | Parameter value range exceeded                                                                                                                                                       |
| 0x06090031          | Parameter values too high                                                                                                                                                            |
| 0x06090032          | Parameter values too low                                                                                                                                                             |
| 0x06090036          | Maximum value falls below minimum value                                                                                                                                              |
| 0x08000000          | General error                                                                                                                                                                        |
| 0x08000020          | Data cannot be transferred or stored for the application.                                                                                                                            |
| 0x08000021          | Because of local control, data cannot be transferred or stored for the application.                                                                                                  |
| 0x08000022          | Because of current controller status, data cannot be transferred or stored for the application.                                                                                      |
| 0x08000023          | Dynamic generation of object directory not successful or no object directory available (e.g. object directory generated from file, generation not possible because of a file error). |

## 10.5 Examples of parameter data telegram

### Reading parameters

The heatsink temperature C0061 ( 43 °C) is to be read from the controller with node address 5 via the parameter data channel 1.

#### ► Identifier calculation

| Identifier from SDO1 to controller | Calculation     |
|------------------------------------|-----------------|
| 1536 + node address                | 1536 + 5 = 1541 |

#### ► Command "Read Request" (request to read a parameter from the controller)

| Command      | Value [hex] |
|--------------|-------------|
| Read request | 0x40        |

#### ► Index calculation

| Index               | Calculation                 |
|---------------------|-----------------------------|
| 24575 - code number | 24575 - 61 = 24514 = 0x5FC2 |

#### ► Subindex: 0

#### ► Telegram to controller

| Identifier | User data |                |                 |          |        |        |        |        |
|------------|-----------|----------------|-----------------|----------|--------|--------|--------|--------|
|            | Command   | Index LOW byte | Index HIGH byte | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
| 1541       | 0x40      | 0xC2           | 0x5F            | 0x00     | 0x00   | 0x00   | 0x00   | 0x00   |

#### ► Telegram from controller

| Identifier | User data |                |                 |          |        |        |        |        |
|------------|-----------|----------------|-----------------|----------|--------|--------|--------|--------|
|            | Command   | Index LOW byte | Index HIGH byte | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
| 1413       | 0x43      | 0xC2           | 0x5F            | 0x00     | 0xB0   | 0x8F   | 0x06   | 0x00   |

- Command:  
"Read Response" (response to the read request) = 0x43
- Identifier:  
SDO1 from controller (1408) + node address (5) = 1413
- Index of the read request:  
0x5FC2
- Subindex:  
0
- Data 1 ... 4:  
0x00068FB0 = 430000 → 430000 : 10000 = 43 °C

**Writing parameters**

The acceleration time C0012 (parameter set 1) of the controller with the node address 1 is to be changed to 20 seconds via the SDO1 (parameter data channel 1).

## ► Identifier calculation

| Identifier from SDO1 to controller | Calculation       |
|------------------------------------|-------------------|
| 1536 + node address                | $1536 + 1 = 1537$ |

## ► Command "Write Request" (transmit parameter to drive)

| Command       | Value [hex] |
|---------------|-------------|
| Write request | 0x23        |

## ► Index calculation

| Index               | Calculation                   |
|---------------------|-------------------------------|
| 24575 - code number | $24575 - 12 = 24563 = 0x5FF3$ |

## ► Subindex: 0

## ► Calculation of the acceleration time

| Data 1 ... 4                | Calculation                                           |
|-----------------------------|-------------------------------------------------------|
| Value for acceleration time | $20 \text{ s} \cdot 10000 = 200000$<br>$= 0x00030D40$ |

## ► Telegram to controller

| Identifier | User data |                   |                    |          |        |        |        |        |
|------------|-----------|-------------------|--------------------|----------|--------|--------|--------|--------|
|            | Command   | Index<br>LOW byte | Index<br>HIGH byte | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
| 1537       | 0x23      | 0xF3              | 0x5F               | 0x00     | 0x40   | 0x0D   | 0x03   | 0x00   |

## ► Telegram from controller if executed faultlessly

| Identifier | User data |                   |                    |          |        |        |        |        |
|------------|-----------|-------------------|--------------------|----------|--------|--------|--------|--------|
|            | Command   | Index<br>LOW byte | Index<br>HIGH byte | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
| 1409       | 0x60      | 0xF3              | 0x5F               | 0x00     | 0x00   | 0x00   | 0x00   | 0x00   |

– Command:

"Write Response" (response of the controller (acknowledgement)) = 0x60

– Identifier:

SDO1 from controller (= 1408) + node address (= 1) = 1409

### Read block parameters

The software product code (code C0200) of the Lenze 8200 vector is to be read from parameter set 1. The product code has 14 alphanumeric characters. They are transferred as block parameters. The transfer of block parameters uses the entire data width (2nd to 8th byte).

During user data transfer, the command byte (1. Byte) contains the entry 0x40 or 0x41 to ...

- be able to signalise the end of the block transfer;
- request the next block.

#### ► C0200 - request

| Command | Index<br>Low byte | Index<br>High byte | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
|---------|-------------------|--------------------|----------|--------|--------|--------|--------|
| 0x40    | 0x37              | 0x5F               | 0x00     | 0x00   | 0x00   | 0x00   | 0x00   |

Byte 1: 0x40, "Read request" (requirement to read a parameter from the controller)

Byte 2/3: Index low/high byte: 24575 - 200 - 0 = 24375 (0x5F37)

#### ► Response including the block length (14 characters)

| Command | Index<br>Low byte | Index<br>High byte | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
|---------|-------------------|--------------------|----------|--------|--------|--------|--------|
| 0x41    | 0x37              | 0x5F               | 0x00     | 0x0E   | 0x00   | 0x00   | 0x00   |

Byte 1: 0x41, "read response". The entry 0x41 implies that it is a block diagram.

Byte 2/3: Index low/high byte: 24575 - 200 - 0 = 24375 (0x5F37)

Byte 5: 0x0E = data length of 14 characters (ASCII format)

#### ► First data block - request



#### Note!

The single blocks are toggled\*, i.e. at first it is requested with command 0x60 (0b01100000), then with command 0x70 (0b01110000), after his again with 0x6 etc.

The response is sent accordingly. It is alternating because of a toggle bit. The process is completed by command 0x11 (bit 0 is set, see below).

\*Toggle-Bit = bit 4 (starting with "0")

| Command | Index<br>Low byte | Index<br>High byte | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
|---------|-------------------|--------------------|----------|--------|--------|--------|--------|
| 0x60    | 0x00              | 0x00               | 0x00     | 0x00   | 0x00   | 0x00   | 0x00   |

Byte 1: 0x60, "Write response" (acknowledgement) with access to bytes 2 ... 8.

#### ► Response

| Command | Index<br>Low byte | Index<br>High byte | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
|---------|-------------------|--------------------|----------|--------|--------|--------|--------|
| 0x00    | 0x38              | 0x32               | 0x53     | 0x38   | 0x32   | 0x31   | 0x32   |

Bytes 2 ... 8, ASCII format: 8 2 5 8 2 1 2

## ► Second data block - request

| Command | Index<br>Low byte | Index<br>High byte | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
|---------|-------------------|--------------------|----------|--------|--------|--------|--------|
| 0x70    | 0x00              | 0x00               | 0x00     | 0x00   | 0x00   | 0x00   | 0x00   |

Byte 1: 0x70 (Toggle), "Write response" (acknowledgement) with access to all 4 data bytes

## ► Second data block - response with over-detection

| Command | Index<br>Low byte | Index<br>High byte | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
|---------|-------------------|--------------------|----------|--------|--------|--------|--------|
| 0x11    | 0x56              | 0x5F               | 0x31     | 0x34   | 0x30   | 0x30   | 0x30   |

Byte 1: 0x11, last data block transfer

Bytes 2 ... 8, ASCII format: V\_14000

**Result of data block transfer: 82S8212V\_14000**

**10.6 Special features for parameter setting of the drive controller****10.6.1 8200 vector controller**

The digital and analog input and output signals are freely configurable (see System Manual 8200 vector; code C0410, C0412, C0417 and C0421).

**10.6.2 9300 Servo PLC / Drive PLC / ECS****AIF control byte (C2120)/AIF status byte (C2121)**

The controller and the EMF2178IB communication 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.

**C2120:  
AIF control byte**

| Code  | Subcode | Index          | Possible settings |                 | Data type |
|-------|---------|----------------|-------------------|-----------------|-----------|
|       |         |                | Lenze             | Selection       |           |
| C2120 | -       | 22455 = 0x57B7 | 0                 | See table below | FIX32     |

| Possible settings       |                                                         |
|-------------------------|---------------------------------------------------------|
| Selection               | Description                                             |
| 0                       | No command                                              |
| 1                       | Update codes 23XX and CAN reinitialisation ≡ reset node |
| 2                       | Update codes C23XX                                      |
| 10                      | Reread C2356/1...4                                      |
| 11                      | Reread C2357                                            |
| 12                      | Reread C2375                                            |
| 13                      | Reread C2376 ... C2378                                  |
| 14                      | Reread C2382                                            |
| 16 <sup>1)</sup>        | Update C7999 and C7998 including CAN reinitialisation   |
| 17 <sup>1)</sup>        | Update C7999, mapping table of profile indices          |
| 18 <sup>1)</sup>        | Update C7998, error list of emergency messages          |
| 32 <sup>1)</sup>        | Reset error source, activate emergency messages         |
| 33 ... 62 <sup>1)</sup> | Error source 1 ... 30, activate emergency messages      |

<sup>1)</sup> Value range is only valid for profile applications in a PLC

With the AIF control byte it is possible to read the codes C23XX saved in the 9300 Servo PLC, Drive PLC and ECS into the communication module.

**Execute command**

How to accept the values in the communication module:

- Write the table value of the required command into the AIF control byte.
- Change the status of the MSB of the AIF control byte. The command is executed.

**Example**

By writing the value "2" into the AIF control byte all codes C23XX are re-read when the MSB changes its status.

For some codes it is necessary to carry out a CAN reinitialisation so that new values and the functions derived from them will become effective.

**C2121:****AIF status byte**

| Code  | Subcode | Index             | Possible settings |           |     | Data type |
|-------|---------|-------------------|-------------------|-----------|-----|-----------|
|       |         |                   | Lenze             | Selection |     |           |
| C2121 | -       | 22454 =<br>0x57B6 | 0                 | 0 [1]     | 255 | U8        |

The AIF status byte provides the 9300 Servo PLC, Drive PLC and ECS with information of the communication module. With this information, the 9300 Servo PLC, Drive PLC and ECS can monitor the status of the communication module. Depending on the communication module states, it is possible for you to take corresponding countermeasures.

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

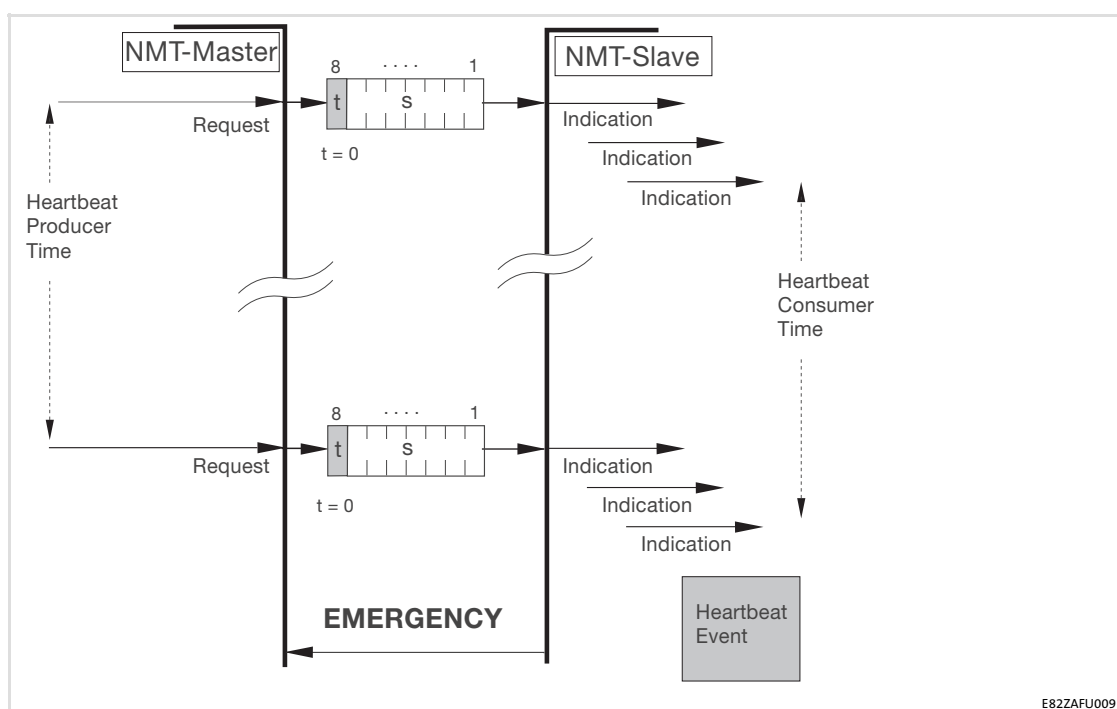
## 11 Monitoring

### 11.1 Heartbeat Protocol



#### Note!

- Only one monitoring function can be active:
  - Heartbeat or node guarding.
- Heartbeat has priority over node guarding:
  - If both functions are configured, the heartbeat settings will be active.



E82ZAFU009

#### CANopen objects for configuration

| CANopen |          |                                    | Lenze |         |                           |
|---------|----------|------------------------------------|-------|---------|---------------------------|
| Index   | Subindex | Designation                        | Code  | Subcode | Description               |
| I-1016  | 1        | Consumer heartbeat time and COB-ID | C1869 |         | Consumer heartbeat COB-ID |
|         |          |                                    | C2369 |         |                           |
|         |          |                                    | C1870 | 1       | Consumer heartbeat time   |
| I-1017  | 0        | Producer heartbeat time            | C2370 |         |                           |
|         |          |                                    | C1870 | 2       | Producer heartbeat time   |

The "heartbeat producer" cyclically sends a heartbeat message to one or several "heartbeat consumers".

### Heartbeat transmission

The heartbeat transmission is activated by entering a time under in the CANopen object I-1017. The monitoring is activated by setting a time and a node ID in the object I-1016/1.

### Telegram structure

NMT identifier plus node ID and a data byte with status information:

| Data value (s) | Status          |
|----------------|-----------------|
| 4              | Stopped         |
| 5              | Operational     |
| 127            | Pre-operational |

### Identifier

**Identifier = Basisadresse (1792) + configurable address (1 ... 127)**

### Heartbeatmonitoring

The consumer monitors whether the message is received within the "heartbeat consumer time". If this is not the case, an emergency telegram with a "heartbeat event" is sent.

The controller response to a "heartbeat event" is set under subcode C1882/5 / C2382/5.

If the "heartbeat event" occurs, the node changes from the "Operational" state to the "Pre-operational" state (default setting). By means of the object I-1029 a different response can be set.

The heartbeat monitoring only starts when the first heartbeat telegram of a node monitored has been successfully received and the "Pre-operational" NMT state has been reached.



### Tip!

Detailed information on the objects I-1016, I-1017 and I-1029 can be found in the chapter entitled "Implemented CANopen objects" (📖 91).

## 11.2 Node Guarding Protocol



### Note!

- ▶ Only one monitoring function can be active:
  - Heartbeat or node guarding.
- ▶ Heartbeat has priority over node guarding:
  - If both functions are configured, the heartbeat settings will be active.

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



### Note!

An NMT master can be 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 EMF2178IB communication module.

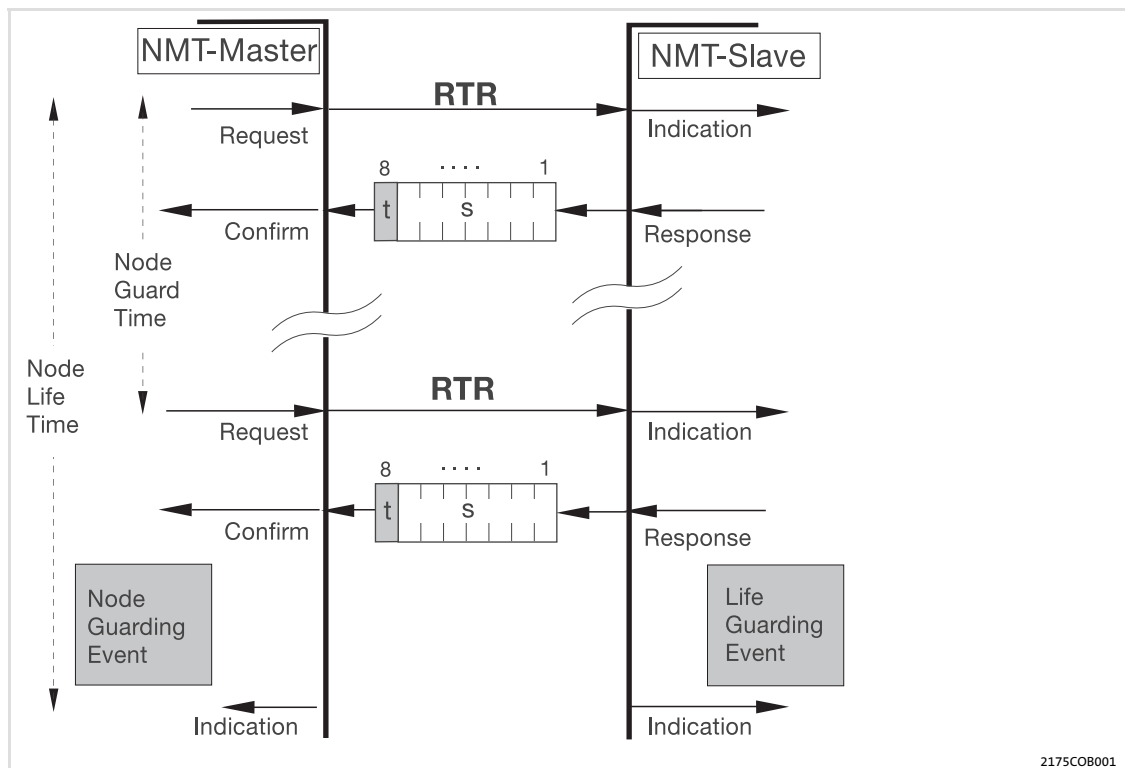


Fig. 11-1 Telegram transfer between NMT master and NMT slave

### CANopen objects for configuration

| CANopen |          |                  | Lenze          |         |                              |
|---------|----------|------------------|----------------|---------|------------------------------|
| Index   | Subindex | Designation      | Code           | Subcode | Description                  |
| I-100C  | 0        | Guard time       | C1827<br>C2327 | -       | Change of "guard time"       |
| I-100D  | 0        | Life time factor | C1828<br>C2328 | -       | Change of "life time factor" |

### RTR telegram

The NMT master cyclically ("node guard time", monitoring time) sends a data telegram called "Remote Transmit Request" (RTR) to the NMT slave.

The RTR prompts the NMT slave to send its current data.

### Response telegram

The NMT slave sends a response telegram with a user data width of 1 byte. The most significant bit of the response telegram is a toggle bit (t).

The data value (s) of the other seven bits indicates the status of the NMT slave:

| Data value (s) | Status          |
|----------------|-----------------|
| 4              | Stopped         |
| 5              | Operational     |
| 127            | Pre-operational |

### Identifier

**Identifier = basic address (1792) + configurable address (1 ... 127)**

### Node life time

The "node life time" is the product of "node guard time" (object I-100C) and "life time factor" (object I-100D).

### Life guarding event

If the NMT slave does not receive an RTR telegram from the NMT master in the "node life time", the "life guarding event" triggers a fault signal in the NMT slave. An emergency telegram with a "life guarding event" is sent.

The controller response to a "life guarding event" is set under subcode C1882/5 / C2382/5.

If the "life guarding event" occurs, the node changes from the "Operational" state to the "Pre-operational" state (default setting). By means of the object I-1029 a different response can be set.

### Node guarding event

The "node guarding event" should occur in the NMT master if despite the request by the NMT master no response from the NMT slave is received in the "node life time" or the toggle bit has not changed in the "node life time".

### 11.3 Emergency telegram

An emergency telegram is sent once to the master if the error status of the communication module changes, i.e.

- ▶ if an internal communication module error occurs.
- ▶ if an internal communication module error is eliminated.

The emergency telegram sent via the CAN bus is structured as follows:

- ▶ Byte 1 + 2: Emergency error code
- ▶ Byte 3: Error register object I-1001
- ▶ Bytes 4 ... 8: Field for manufacturer-specific error messages

Contents:

- Emergency error code 0x1000: Lenze error number
- All other emergency error codes have the value "0".

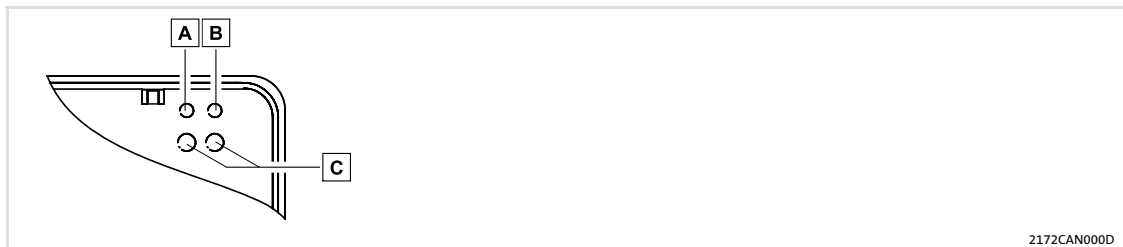
| Emergency error codes<br>[hex] | Cause                                                                       | Error register entry<br>(I-1001) |
|--------------------------------|-----------------------------------------------------------------------------|----------------------------------|
| 0x0000                         | One of several errors eliminated                                            | xx                               |
|                                | Single error eliminated<br>(no more errors)                                 | 00                               |
| 0x1000                         | Basic device in TRIP, message, warning, FAIL, QSP or persistent fault       | 01                               |
| 0x3100                         | Supply voltage of basic device faulty or failed                             | 01                               |
| 0x6280                         | Fault in mapping table with operation with a profile implemented in the PLC | 01                               |
| 0x8100                         | Communication error (warning)                                               | 11                               |
| 0x8130                         | "Life guard error" or "heartbeat error"                                     | 11                               |
| 0x8210                         | PDO length shorter than expected                                            | 11                               |
| 0x8220                         | PDO length longer than expected                                             | 11                               |
| 0x8700                         | Sync telegram monitoring                                                    | 11                               |

The following table shows error causes and remedy measures if there is no communication with the controller.

| Possible cause of error                            | Diagnostics                                                                                     | Remedy                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|----------------------------------------------------|-------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Is the controller switched on?                     | One of the operating status LEDs of the basic device (C) must be on.                            | Supply controller with voltage (see Operating Instructions/Manual for the basic device)                                                                                                                                                                                                                                                                                                                                                                                  |
| Is the communication module supplied with voltage? | The green LED A (connection status with controller) is on (remedy 1) or blinking (remedy 2).    | <ul style="list-style-type: none"> <li>• In case of supply from the controller check the connection. With external supply check the 24 V voltage at terminals 39 and 59. A voltage of 24 V +10 % must be applied.</li> <li>• The communication module has not been initialised with the controller yet.</li> <li>• <b>Remedy 1:</b> Controller not switched on (see fault possibility 1).</li> <li>• <b>Remedy 2:</b> Check the connection to the controller.</li> </ul> |
| Does the controller receive telegrams?             | The green LED B (connection status with bus) must be blinking when communicating with the host. | Check whether the connection corresponds to the instructions given in the chapter "Electrical installation".<br>Check whether host sends telegrams and uses the appropriate interface.<br>Is the existing node address already assigned?                                                                                                                                                                                                                                 |

## 12.2 LED status displays

The LEDs on the front are provided to the communication module for the purpose of fault diagnostics.



2172CAN000D

Fig. 12-1 LEDs of the communication module

| Pos.          | Status display (LED)                                                                                                                                                        |               | Description                                                                                                                                                                                                                                                              |                                                                                                                                |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| A             | Connection status to standard device, two-colour LED (green/red)                                                                                                            |               |                                                                                                                                                                                                                                                                          |                                                                                                                                |
|               | OFF                                                                                                                                                                         |               | <ul style="list-style-type: none"><li>● The communication module is not supplied with voltage.</li><li>● The external voltage supply is switched off.</li></ul>                                                                                                          |                                                                                                                                |
|               | GREEN                                                                                                                                                                       | Blinking      | The communication module is supplied with voltage, but has no connection to the standard device.<br>Cause:<br>The standard device is ... <ul style="list-style-type: none"><li>● switched off;</li><li>● in the initialisation phase;</li><li>● not available.</li></ul> |                                                                                                                                |
|               |                                                                                                                                                                             | Constantly ON | The communication module is supplied with voltage and is connected to the standard device.                                                                                                                                                                               |                                                                                                                                |
|               | RED                                                                                                                                                                         | Constantly ON | CANopen operation impossible.                                                                                                                                                                                                                                            |                                                                                                                                |
|               |                                                                                                                                                                             | Blinking      | Permanently:                                                                                                                                                                                                                                                             | <ul style="list-style-type: none"><li>● Parameters are reset to Lenze setting.</li><li>● CANopen operation possible.</li></ul> |
|               |                                                                                                                                                                             |               | 1 x blinking:                                                                                                                                                                                                                                                            | <ul style="list-style-type: none"><li>● An error occurred while saving a value.</li></ul>                                      |
| 2 x blinking: | <ul style="list-style-type: none"><li>● CANopen operation possible.</li><li>● The node address/ baud rate from C1850/C2350 or C1851/C2351 could not be accepted .</li></ul> |               |                                                                                                                                                                                                                                                                          |                                                                                                                                |

| Pos. | Status display (LED)                                                                    | Description                                                                                                                        |
|------|-----------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| B    | Connection status to fieldbus, two-colour LED (green/red)                               |                                                                                                                                    |
|      | OFF                                                                                     | No connection to the master                                                                                                        |
|      | GREEN  | CANopen status ("Z")                                                                                                               |
|      | RED    | CANopen error ("F")                                                                                                                |
|      | RED constantly on                                                                       | Z: Bus off<br>                                   |
|      | Blinking GREEN every 0.2 seconds                                                        | Z: Pre-operational, F: None<br>                  |
|      | GREEN every 0.2 seconds<br>1 x blinking RED, 1 s OFF                                    | Z: Pre-operational, F: Warning limit reached<br> |
|      | Blinking GREEN every 0.2 seconds<br>2 x blinking RED, 1 s OFF                           | Z: Pre-operational, F: Node guard event<br>      |
|      | GREEN constantly on                                                                     | Z: Operational, F: None<br>                      |
|      | GREEN constantly on<br>1 x blinking RED, 1 s OFF                                        | Z: Operational, F: Warning limit reached<br>     |
|      | GREEN constantly on<br>2 x blinking RED, 1 s OFF                                        | Z: Operational, F: Node guard event<br>          |
|      | GREEN constantly on<br>3 x blinking RED, 1 s OFF                                        | Z: Operational, F: Sync message error<br>        |
|      | Blinking GREEN every second                                                             | Z: Stopped, F: None<br>                        |
| C    | Green and red Drive LED                                                                 | Operating status of standard device<br>(see standard device documentation)                                                         |

## **13 Implemented CANopen objects**

Lenze controllers can be parameterised with Lenze codes and manufacturer-independent "CANopen objects". A completely CANopen-conform communication can only be achieved by using CANopen objects for parameter setting.

All CANopen objects described in these instructions are defined according to the "CiA Draft Standard 301/version 4.02".



### **Note!**

Some of the terms used here derive from the CANopen protocol.

### **13.1 Reference between CANopen object and Lenze code**

CANopen objects and Lenze codes do not have the same functionalities.

Some CANopen objects have a direct influence on the corresponding codes:

- ▶ If a new value is written within an object, the value is also adopted in the corresponding code C18xx or C23xx.
- ▶ When reading an object, the values stored under the corresponding code are displayed.

#### **Example**

The CANopen object I-1017 (producer heartbeat time) is mapped on the codes C1870/2 and C2370/2:

- ▶ Reading the object I-1017:
  - Response: Value under code C1870/C2370, subcode 2.
- ▶ Writing a new value into the object I-1017:
  - The new producer heartbeat time is also entered under C1870/2 and C2370/2.

## 13 Implemented CANopen objects

### Overview

#### 13.2 Overview

| CANopen objects |                                               |                                    | Lenze codes    |         |                                                                                                   |
|-----------------|-----------------------------------------------|------------------------------------|----------------|---------|---------------------------------------------------------------------------------------------------|
| Index           | Subindex                                      | Designation                        | Code           | Subcode | Description                                                                                       |
| I-1000          | 0                                             | Device Type                        | -              | -       | -                                                                                                 |
| I-1001          | 0                                             | Error register                     | -              | -       | -                                                                                                 |
| I-1003          | 0 ... 10<br>(depending on the error messages) | Error history                      | -              | -       | -                                                                                                 |
| I-1005          | 0                                             | COB-ID SYNC message                | C1867<br>C2367 | -       | Change of COB-ID Rx                                                                               |
|                 |                                               |                                    | C1868<br>C2368 | -       | Change of COB-ID Tx                                                                               |
|                 |                                               |                                    | C1856<br>C2356 | 5       | The time entered depends on bit 30 (sync sending).                                                |
|                 |                                               |                                    | C1857<br>C2357 | 6       |                                                                                                   |
| I-1006          | 0                                             | Communication cycle period         | C1856<br>C2356 | 5       | Enter the value corresponding to bit 30 of I-1005 <sub>hex</sub> .                                |
|                 |                                               |                                    | C1857<br>C2357 | 6       |                                                                                                   |
| I-1008          | 0                                             | Manufacturer's device name         | -              | -       | -                                                                                                 |
| I-100A          | 0                                             | Manufacturer's software version    | -              | -       | -                                                                                                 |
| I-100C          | 0                                             | Guard time                         | C1827<br>C2327 | -       | Change of "guard time"                                                                            |
| I-100D          | 0                                             | Life time factor                   | C1828<br>C2328 | -       | Change of "life time factor"                                                                      |
| I-1010          | 0                                             | Store parameters (U8)              | -              | -       | -                                                                                                 |
|                 | 1...3                                         | Store parameters (U32)             |                |         |                                                                                                   |
| I-1011          | 0 ... 7                                       | Restore default parameters         | -              | -       | -                                                                                                 |
| I-1014          | 0                                             | COB-ID emergency object            | C1871<br>C2371 | -       | Change of "COB-ID emergency" (setting bit 31 to deactivate the emergency is not directly depend.) |
| I-1015          | 0                                             | Emergency inhibit time             | C1872<br>C2372 | -       | Change of delay time                                                                              |
| I-1016          | 0, 1                                          | Consumer heartbeat time and COB-ID | C1869<br>C2369 | -       | Change of consumer heartbeat COB-ID                                                               |
|                 |                                               |                                    | C1870<br>C2370 | 1       | Change of consumer heartbeat monitoring time                                                      |
| I-1017          | 0                                             | Producer heartbeat time            | C1870<br>C2370 | 2       | Change of cycle time for the producer heartbeat telegram                                          |
| I-1018          | 0 ... 4                                       | Module device description          | -              | -       | -                                                                                                 |
| I-1029          | 0, 1                                          | Error behaviour                    | -              | -       | -                                                                                                 |
| I-1200          | 0                                             | Server SDO1 parameters             | -              | -       | -                                                                                                 |
|                 | 1                                             | Identifier client →server (rx)     |                |         |                                                                                                   |
|                 | 2                                             | Identifier server →client (tx)     |                |         |                                                                                                   |
| I-1201          | 0                                             | Server SDO2 parameters             | C1865<br>C2365 | 1       | Change of validity of SDO 2                                                                       |
|                 | 1                                             | Identifier client →server (rx)     |                |         |                                                                                                   |
|                 | 2                                             | Identifier server →client (tx)     |                |         |                                                                                                   |

| CANopen objects |          |                                   | Lenze codes |         |                                  |
|-----------------|----------|-----------------------------------|-------------|---------|----------------------------------|
| Index           | Subindex | Designation                       | Code        | Subcode | Description                      |
| I-1400          | 0        | Number of entries                 | -           | -       | -                                |
|                 | 1        | RPDO1 COB-ID                      | C1853       | 1       | Change of COB-ID:                |
|                 |          |                                   | C2353       |         | Set code to CANopen addressing.  |
|                 |          |                                   | C1855       | 1       | Enter the new ID under the code. |
|                 |          |                                   | C2355       |         |                                  |
|                 | 2        | RPDO1 transmission type           | C1865       | 2       | Change of validity of the PDOs   |
|                 |          |                                   | C2365       |         |                                  |
|                 |          |                                   | C1873       | 1       | Change of transmission type      |
|                 | 5        | RPDO1 event timer                 | C2373       |         |                                  |
|                 |          |                                   | C1857       | 1       | Change of monitoring time        |
|                 | C2357    |                                   |             |         |                                  |
|                 |          |                                   |             |         |                                  |
| I-1401          | 0        | Number of entries                 | -           | -       | -                                |
|                 | 1        | RPDO2 COB-ID                      | C1853       | 2       | Change of COB-ID:                |
|                 |          |                                   | C2353       |         | Set code to CANopen addressing.  |
|                 |          |                                   | C1855       | 3       | Enter the new ID under the code. |
|                 |          |                                   | C2355       |         |                                  |
|                 | 2        | RPDO2 transmission type           | C1865       | 3       | Change of validity of the PDOs   |
|                 |          |                                   | C2365       |         |                                  |
|                 |          |                                   | C1873       | 2       | Change of transmission type      |
|                 | 5        | RPDO2 event timer                 | C2373       |         |                                  |
|                 |          |                                   | C1857       | 2       | Change of monitoring time        |
|                 | C2357    |                                   |             |         |                                  |
|                 |          |                                   |             |         |                                  |
| I-1402          | 0        | Number of entries                 | -           | -       | -                                |
|                 | 1        | RPDO3 COB-ID                      | C1853       | 3       | Change of COB-ID:                |
|                 |          |                                   | C2353       |         | Set code to CANopen addressing.  |
|                 |          |                                   | C1855       | 5       | Enter the new ID under the code. |
|                 |          |                                   | C2355       |         |                                  |
|                 | 2        | RPDO3 transmission type           | C1865       | 4       | Change of validity of the PDOs   |
|                 |          |                                   | C2365       |         |                                  |
|                 |          |                                   | C1873       | 3       | Change of transmission type      |
|                 | 5        | RPDO3 event timer                 | C2373       |         |                                  |
|                 |          |                                   | C1857       | 3       | Change of monitoring time        |
|                 | C2357    |                                   |             |         |                                  |
|                 |          |                                   |             |         |                                  |
| I-1600          | 0        | Number of mapped objects in RPDOs | -           | -       | -                                |
|                 | 1        | RPDO mapping 1                    |             |         |                                  |
|                 | 2        | RPDO mapping 2                    |             |         |                                  |
|                 | 3        | RPDO mapping 3                    |             |         |                                  |
|                 | 4        | RPDO mapping 4                    |             |         |                                  |
| I-1601          | 0        | Number of mapped objects in RPDOs | -           | -       | -                                |
|                 | 1        | RPDO mapping 1                    |             |         |                                  |
|                 | 2        | RPDO mapping 2                    |             |         |                                  |
|                 | 3        | RPDO mapping 3                    |             |         |                                  |
|                 | 4        | RPDO mapping 4                    |             |         |                                  |
| I-1602          | 0        | Number of mapped objects in PDOs  | -           | -       | -                                |
|                 | 1        | RPDO mapping 1                    |             |         |                                  |
|                 | 2        | RPDO mapping 2                    |             |         |                                  |
|                 | 3        | RPDO mapping 3                    |             |         |                                  |
|                 | 4        | RPDO mapping 4                    |             |         |                                  |

| CANopen objects |          |                                   | Lenze codes    |         |                                                      |
|-----------------|----------|-----------------------------------|----------------|---------|------------------------------------------------------|
| Index           | Subindex | Designation                       | Code           | Subcode | Description                                          |
| I-1800          | 0        | Number of entries                 | -              | -       | -                                                    |
|                 | 1        | TPDO1 COB-ID                      | C1853<br>C2353 | 1       | Change of COB-ID:<br>Set code to CANopen addressing. |
|                 |          |                                   | C1855<br>C2355 | 2       | Enter the new ID under the code.                     |
|                 |          |                                   | C1865<br>C2365 | 2       | Change of validity of the PDOs                       |
|                 | 2        | TPDO1 transmission type           | C1874<br>C2374 | 1       | Change of transmission type                          |
|                 |          |                                   | C1875<br>C2375 | 1       | Change of mode                                       |
|                 | 3        | Inhibit time                      | -              | -       | -                                                    |
| I-1801          | 5        | TPDO1 event timer                 | C1856<br>C2356 | 2       | Change of cycle time                                 |
|                 | 0        | Number of entries                 | -              | -       | -                                                    |
|                 | 1        | TPDO2 COB-ID                      | C1853<br>C2353 | 2       | Change of COB-ID:<br>Set code to CANopen addressing. |
|                 |          |                                   | C1855<br>C2355 | 4       | Enter the new ID under the code.                     |
|                 |          |                                   | C1865<br>C2365 | 3       | Change of validity of the PDOs                       |
|                 | 2        | TPDO2 transmission type           | C1874<br>C2374 | 2       | Change of transmission type                          |
|                 |          |                                   | C1875<br>C2375 | 2       | Change of mode                                       |
| I-1802          | 3        | Inhibit time                      | -              | -       | -                                                    |
|                 | 5        | TPDO2 event timer                 | C1856<br>C2356 | 3       | Change of cycle time                                 |
|                 | 0        | Number of entries                 | -              | -       | -                                                    |
|                 | 1        | TPDO3 COB-ID                      | C1855<br>C2355 | 6       | Enter the new ID under the code.                     |
|                 |          |                                   | C1865<br>C2365 | 4       | Change of validity of the PDOs                       |
|                 | 2        | TPDO3 transmission type           | C1874<br>C2374 | 3       | Change of transmission type                          |
|                 |          |                                   | C1875<br>C2375 | 3       | Change of mode                                       |
| I-1A00          | 3        | Inhibit time                      | -              | -       | -                                                    |
|                 | 5        | TPDO3 event timer                 | C1856<br>C2356 | 4       | Change of cycle time                                 |
|                 | 0        | Number of mapped objects in TPDOs | -              | -       | -                                                    |
|                 | 1        | TPDO mapping 1                    |                |         |                                                      |
|                 | 2        | TPDO mapping 2                    |                |         |                                                      |
|                 | 3        | TPDO mapping 3                    |                |         |                                                      |
|                 | 4        | TPDO mapping 4                    |                |         |                                                      |
| I-1A01          | 0        | Number of mapped objects in TPDOs | -              | -       | -                                                    |
|                 | 1        | TPDO mapping 1                    |                |         |                                                      |
|                 | 2        | TPDO mapping 2                    |                |         |                                                      |
|                 | 3        | TPDO mapping 3                    |                |         |                                                      |
|                 | 4        | TPDO mapping 4                    |                |         |                                                      |

| CANopen objects |          |                                   | Lenze codes |         |             |
|-----------------|----------|-----------------------------------|-------------|---------|-------------|
| Index           | Subindex | Designation                       | Code        | Subcode | Description |
| I-1A02          | 0        | Number of mapped objects in TPDOs | -           | -       | -           |
|                 | 1        | TPDO mapping 1                    |             |         |             |
|                 | 2        | TPDO mapping 2                    |             |         |             |
|                 | 3        | TPDO mapping 3                    |             |         |             |
|                 | 4        | TPDO mapping 4                    |             |         |             |

## 13 Implemented CANopen objects

### Overview

#### I-1000: Device type

##### 13.2.1 I-1000: Device type

| Index (hex)<br><b>0x1000</b> | Name<br><b>Device Type</b> |             |     |              |                     |
|------------------------------|----------------------------|-------------|-----|--------------|---------------------|
| Subindex                     | Default setting            | Value range |     |              | Access    Data type |
| 0                            | -                          | 0           | ... | $2^{32} - 1$ | ro    U32           |

The object I-1000 indicates the device profile for this device. It is also possible to include additional information here that is defined in the device profile itself. If a specific device profile is not used, the content is "0x0000".

##### Data telegram assignment

| Byte 8                | 7th byte | Byte 6                 | Byte 5 |
|-----------------------|----------|------------------------|--------|
| U32                   |          |                        |        |
| Device profile number |          | Additional information |        |

##### 13.2.2 I-1001: Error register

##### Reading of the error register

| Index (hex)<br><b>0x1001</b> | Name<br><b>Error register</b> |             |     |     |                     |
|------------------------------|-------------------------------|-------------|-----|-----|---------------------|
| Subindex                     | Default setting               | Value range |     |     | Access    Data type |
| 0                            | -                             | 0           | ... | 255 | ro    U8            |

##### Bit assignment in the data byte (U8) for the individual error states

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Error status         |
|-------|-------|-------|-------|-------|-------|-------|-------|----------------------|
| 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | No fault             |
| 0     | 0     | 0     | 0     | 0     | 0     | 0     | 1     | Device error message |
| 0     | 0     | 0     | 1     | 0     | 0     | 0     | 1     | Communication error  |

**13.2.3 I-1003: Error history**

## Error history

| Index (hex)<br><b>0x1003</b> |                           | Name<br><b>Error history</b> |             |                  |                     |
|------------------------------|---------------------------|------------------------------|-------------|------------------|---------------------|
| Subindex                     |                           | Default setting              | Value range |                  | Access    Data type |
| 0                            | Number of recorded errors | -                            | 0           | ... 255          | rw    U8            |
| 1                            | Standard error field      | -                            | 0           | ... $2^{32} - 1$ | ro    U32           |
| ...                          |                           |                              |             |                  |                     |
| 10                           |                           |                              |             |                  |                     |

This object shows that errors have occurred in the communication module and in the basic device:

- ▶ Subindex 0: Number of saved error messages.
- ▶ Subindex 1 ... 10: Error list.

The error messages consist of a 16-bit error code and a 16-bit manufacturer-specific information field.

**Note!**

The values in the "standard error field" under subindex 1 ... 10 will be completely deleted if subindex 0 "number of recorded errors" is overwritten with "0".

## 13.2.4 I-1005: COB-ID SYNC message

| Index (hex)<br>0x1005 | Name<br>COB-ID SYNC message |                    |        |           |
|-----------------------|-----------------------------|--------------------|--------|-----------|
| Subindex              | Default setting             | Value range        | Access | Data type |
| 0                     | 0x80 or 0x80000080          | 0 ... $2^{32} - 1$ | rw     | U32       |

This object ensures that sync telegrams can be created for the communication module and that the identifier value can be written.

**Creating sync telegrams**

Sync telegrams are created when bit 30 (see below) is set to "1".

The time between the syn telegrams can be set using the object I-1006.

**Writing identifiers**

The default setting for receiving PDOs is "0x80" in the 11-bit identifier (also according to the CANopen specification). This means that all communication modules are default set to the same sync telegram.

If sync telegrams are only to be received by certain communication modules, their identifiers can be entered with values up to "0xFF". The identifiers can only be changed when the communication module does not send sync telegrams (bit 30 = 0).

**Data telegram assignment**

| Byte 8    |     |       |              | 7th byte                                                                 |  | Byte 6 |    |                   | Byte 5 |  |   |     |
|-----------|-----|-------|--------------|--------------------------------------------------------------------------|--|--------|----|-------------------|--------|--|---|-----|
| MSB       |     |       |              | U32                                                                      |  |        |    |                   |        |  |   | LSB |
| 31        | 30  | 29    | 28           | ...                                                                      |  |        | 11 | 10                | ...    |  | 0 |     |
| 0/1       | 0/1 | 0     | Bit value: 0 |                                                                          |  |        |    | 11-bit identifier |        |  |   |     |
|           |     |       |              |                                                                          |  |        |    |                   |        |  |   |     |
| Bits      |     | Value |              | Description                                                              |  |        |    |                   |        |  |   |     |
| 0 ... 10  |     | X     |              | 11-bit identifier                                                        |  |        |    |                   |        |  |   |     |
| 11 ... 29 |     | 0     |              | The extended identifier is not supported. Any of these bits must be "0". |  |        |    |                   |        |  |   |     |
| 30        |     | 0     |              | Device does not create sync telegrams.                                   |  |        |    |                   |        |  |   |     |
|           |     | 1     |              | Device creates sync telegrams.                                           |  |        |    |                   |        |  |   |     |
| 31        |     | X     |              | optional                                                                 |  |        |    |                   |        |  |   |     |

**13.2.5 I-1006: Communication cycle period**

| Index (hex)<br><b>0x1006</b> | Name<br><b>Communication cycle period</b> |                    |        |           |
|------------------------------|-------------------------------------------|--------------------|--------|-----------|
| Subindex                     | Default setting                           | Value range        | Access | Data type |
| 0                            | 0                                         | 0 ... $2^{32} - 1$ | rw     | U32       |

This object is used to set a sync telegram cycle time when sync sending is activated (bit 30 in object I-1005).

If sync sending is deactivated and a cycle time is selected, the selected time is used for monitoring the sync telegrams received.

With the default setting of "0", sync telegrams are not created or monitored.

Cycle times can be entered as "1000" or an integer multiple of "1000". The unit of the entered time is  $\mu\text{s}$ . The maximum value is 65535000  $\mu\text{s}$ .

**13.2.6 I-1008: Manufacturer's device name**

| Index (hex)<br><b>0x1008</b> | Name<br><b>Manufacturer's device name</b> |                 |        |                      |
|------------------------------|-------------------------------------------|-----------------|--------|----------------------|
| Subindex                     | Default setting                           | Value range     | Access | Data type            |
| 0                            | -                                         | Module-specific | const  | VS<br>(9 characters) |

Display of the manufacturer's device designation of controller and communication module.

The manufacturer's device designation comprises a total of 9 characters:

- ▶ 1st ... 4th character: The 4th to 7th character is read out of the software ID of the standard device (C0200).
  - e.g. "8212" from the ID of the 8200 vector frequency inverter
- ▶ 5th character: Use of an underscore (" \_ ")
- ▶ 6th ... 9th character: The 4th to 7th character is read out of the software ID of the standard device (C1500).
  - e.g. "AFU0" from the total "82ZAFU0B\_20000" ID of the E82ZAFUC communication module (CANopen)

## 13 Implemented CANopen objects

### Overview

#### I-100A: Manufacturer software version

### 13.2.7 I-100A: Manufacturer software version

| Index (hex)<br><b>0x100A</b> | Name<br><b>Manufacturer's software version</b> |                 |        |                       |
|------------------------------|------------------------------------------------|-----------------|--------|-----------------------|
| Subindex                     | Default setting                                | Value range     | Access | Data type             |
| 0                            | -                                              | Module-specific | const  | VS<br>(11 characters) |

Display of the manufacturer's software version of controller and communication module.

The manufacturer's software version consists of a total of 11 characters:

- ▶ 1st ... 5th character: The 10th to 14th character is read out of the software ID of the standard device (C0200).
  - 10th character: Major software version
  - 11. character: Minor software version
  - 12th ... 14th character: Power class of the standard device (see documentation of the standard device)
- ▶ 6th character: Use of an underscore (" \_ ")
- ▶ 7th ... 11th character: The 10th to 14th character is read out of the software ID of the standard device (C1500).
  - e.g. "20000" from the total "82ZAFU0B\_20000" ID of the E82ZAFUC communication module (CANopen)

### 13.2.8 I-100C: Guard time

| Index (hex)<br><b>0x100C</b> | Name<br><b>Guard time</b> |             |        |           |
|------------------------------|---------------------------|-------------|--------|-----------|
| Subindex                     | Default setting           | Value range | Access | Data type |
| 0                            | 0                         | 0 ... 65535 | rw     | U16       |

The guard time is indicated in milliseconds.

With the default setting of "0", "node guarding" is not supported.

"Node guarding" in the slave is activated if a value > "0" is selected for the "guard time" and the "life time factor". The "guard time" indicates the time within which the RTRs from the master are awaited.

**13.2.9 I-100D: Life time factor**

| Index (hex)<br><b>0x100D</b> | Name<br><b>Life time factor</b> |             |     |     |                     |
|------------------------------|---------------------------------|-------------|-----|-----|---------------------|
| Subindex                     | Default setting                 | Value range |     |     | Access    Data type |
| 0                            | 0                               | 0           | ... | 255 | rw    U8            |

With the default setting of "0", "node guarding" is not supported.

The product of "guard time" and "life time factor" is important for the monitoring process.

**13.2.10 I-1010: Store parameters**

Parameter storage in the EEPROM

| Index (hex)<br><b>0x1010</b> | Name<br><b>Store parameters</b> |             |     |              |                     |
|------------------------------|---------------------------------|-------------|-----|--------------|---------------------|
| Subindex                     | Default setting                 | Value range |     |              | Access    Data type |
| 0    Store parameters        | -                               | 0           | ... | $2^{32} - 1$ | ro    U8            |
| 1                            |                                 |             |     |              | U32                 |
| ...                          |                                 |             |     |              |                     |
| 3                            |                                 |             |     |              |                     |

**Subindices**

| Subindex | Authorisation | Description                                         |                                                                                           |
|----------|---------------|-----------------------------------------------------|-------------------------------------------------------------------------------------------|
|          |               | Write                                               | Read                                                                                      |
| 0        | ro            | While writing, the 0x06010002 error message occurs. | Supported subindex: 3                                                                     |
| 1        |               | While writing, the 0x08000020 error message occurs. | Reading of memory functions of all parameters                                             |
| 2        |               |                                                     | Reading of memory functions of object communication parameters                            |
| 3        |               |                                                     | Reading of memory functions of manufacturer-specific parameters (Index 0x6000 ... 0x9FFF) |

**Bit assignment (reading)**

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

The following functions are controller-dependent. They are indicated by reading the bit states 0 and 1.

| Bits     | Value | Meaning of the bit assignment      |
|----------|-------|------------------------------------|
| 0        | 0     | No automatic saving of parameters  |
|          | 1     | Automatic saving of parameters     |
| 1        | 0     | No saving of parameters on command |
|          | 1     | Saving of parameters on command    |
| 2 ... 31 | 0     |                                    |

## 13.2.11 I-1011: Restore default parameters

Loading of Lenze settings

**Note!**

For this function, the subindices to be used depend on the controller type.

| Index (hex)<br><b>0x1011</b> |                            | Name<br><b>Restore default parameters</b> |                    |        |           |
|------------------------------|----------------------------|-------------------------------------------|--------------------|--------|-----------|
| Subindex                     |                            | Default setting                           | Value range        | Access | Data type |
| 0                            | Restore default parameters | -                                         | 0 ... $2^{32} - 1$ | ro     | U32       |
| 1                            |                            |                                           |                    | rw     | U32       |
| ...                          |                            |                                           |                    |        |           |
| 7                            |                            |                                           |                    |        |           |

**Subindices**

| Subindex | Authorisation | Description                                                                         |                                                                                                                                                                                     |
|----------|---------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|          |               | Write                                                                               | Read                                                                                                                                                                                |
| 0        | ro            | While writing, the 0x06010002 error message occurs.                                 | Max. available subindex (depending on standard device): <ul style="list-style-type: none"> <li>● 93xx : 3</li> <li>● ECSXX: 3</li> <li>● 82vector : 7</li> <li>● 82xx: 5</li> </ul> |
| 1        | rw            |                                                                                     | All parameters can be loaded                                                                                                                                                        |
| 2        |               | This function is not supported. While writing, the 0x08000020 error message occurs. | Only object communication parameters can be loaded                                                                                                                                  |
| 3        |               |                                                                                     | Only manufacturer-specific parameters can be loaded (Index 0x6000 ... 0x9FFF)                                                                                                       |
| 4        |               |                                                                                     | Parameter set 1 can be loaded                                                                                                                                                       |
| 5        |               |                                                                                     | Parameter set 2 can be loaded                                                                                                                                                       |
| 6        |               |                                                                                     | Parameter set 3 can be loaded                                                                                                                                                       |
| 7        |               |                                                                                     | Parameter set 4 can be loaded                                                                                                                                                       |

**Bit assignment (reading)**

| MSB |    | U32  | LSB |     |
|-----|----|------|-----|-----|
| 31  | 30 | .... | 1   | 0   |
| 0   | 0  | 0    | 0   | 0/1 |

**Bit states**

| Bits     | Value | Meaning of the bit assignment |
|----------|-------|-------------------------------|
| 0        | 0     | Parameters cannot be loaded   |
|          | 1     | Parameters can be loaded      |
| 1 ... 31 | 0     |                               |

**Bit assignment (writing)**

The telegram data must include the signature "**load**" to start the parameter download.

| Signature        | MSB  |      |      |  | LSB  |
|------------------|------|------|------|--|------|
| ISO 8859 (ASCII) | d    | a    | o    |  | l    |
| Value (hex)      | 0x64 | 0x61 | 0x6F |  | 0x6C |

## 13.2.12 I-1014: COB-ID emergency object

| Index (hex)<br><b>0x1014</b> | Name<br><b>COB-ID emergency object</b> |                    |        |           |
|------------------------------|----------------------------------------|--------------------|--------|-----------|
| Subindex                     | Default setting                        | Value range        | Access | Data type |
| 0                            | 0x80 + Node ID                         | 0 ... $2^{32} - 1$ | rw     | U32       |

When communication errors occur, are reset or when internal errors occur in the communication module or controller (e.g. TRIP), the CAN bus sends an error message. The telegram is sent once for every fault. This function can be activated and deactivated with bit 31.

## Data telegram assignment

| Byte 8    |    |       | 7th byte  |                                                                          | Byte 6 |  | Byte 5 |                   |     |   |  |
|-----------|----|-------|-----------|--------------------------------------------------------------------------|--------|--|--------|-------------------|-----|---|--|
| MSB       |    |       | U32       |                                                                          |        |  |        |                   | LSB |   |  |
| 31        | 30 | 29    | 28 ... 11 |                                                                          |        |  | 10     |                   |     | 0 |  |
| 0/1       |    | 0     | 0         | Bit value: 0                                                             |        |  |        | 11-bit identifier |     |   |  |
| Bits      |    | Value |           | Description                                                              |        |  |        |                   |     |   |  |
| 0 ... 10  |    | 0/1   |           | 11-bit identifier                                                        |        |  |        |                   |     |   |  |
| 11 ... 29 |    | 0     |           | The extended identifier is not supported. Any of these bits must be "0". |        |  |        |                   |     |   |  |
| 30        |    | 0     |           | Reserved                                                                 |        |  |        |                   |     |   |  |
| 31        |    | 0     |           | Emergency object is valid                                                |        |  |        |                   |     |   |  |
|           |    | 1     |           | Emergency object is invalid                                              |        |  |        |                   |     |   |  |

**Note!**

The COB-ID can only be changed in the "Emergency object invalid" status (bit 31 = 1).

The emergency telegram sent via the CAN bus is structured as follows:

- Byte 1 + 2: Emergency error code
- Byte 3: Error register object I-1001
- Bytes 4 ... 8: Field for manufacturer-specific error messages
  - Contents:
    - Emergency error code 0x1000: Lenze error number
    - All other emergency error codes have the value "0".

**13.2.13 I-1015: Emergency inhibit time**

| Index (hex)<br><b>0x1015</b> | Name<br><b>Emergency inhibit time</b> |             |     |       |                     |
|------------------------------|---------------------------------------|-------------|-----|-------|---------------------|
| Subindex                     | Default setting                       | Value range |     |       | Access    Data type |
| 0                            | 0                                     | 0           | ... | 65535 | rw    U16           |

This object determines the time between two emergency telegrams.

The value entered is multiplied by 0.1, the result is the delay time in milliseconds.

**13.2.14 I-1016: Consumer heartbeat time**

| Index (hex)<br><b>0x1016</b> | Name<br><b>Consumer heartbeat time</b> |             |     |       |                     |
|------------------------------|----------------------------------------|-------------|-----|-------|---------------------|
| Subindex                     | Default setting                        | Value range |     |       | Access    Data type |
| 0    Consumer heartbeat time | 0                                      | 0           | ... | 65535 | ro    U32           |
| 1                            |                                        |             |     |       | rw    U32           |

**Subindices**

| Subindex | Explanation                          |
|----------|--------------------------------------|
| 0        | Highest subindex                     |
| 1        | Node ID and "heartbeat time" of node |

**Bit assignment, subindex 1**

| U32               |           |                     |
|-------------------|-----------|---------------------|
| 31 ... 24         | 23 ... 16 | 15 ... 0            |
| Reserved, value 0 | Node ID   | Heartbeat time [ms] |

The "consumer heartbeat time" is used to monitor a node.

The "consumer heartbeat time" must be longer than the "producer heartbeat time" of the node to be monitored.

If the selected time is exceeded, the communication module will send an emergency telegram including the "heartbeat error" 0x8130.

Under code C1882/subcode 5, you can set the response of the basic device.

## 13 Implemented CANopen objects

### Overview

#### I-1017: Producer heartbeat time

##### 13.2.15 I-1017: Producer heartbeat time

| Index (hex)<br><b>0x1017</b> | Name<br><b>Producer heartbeat time</b> |             |        |           |  |
|------------------------------|----------------------------------------|-------------|--------|-----------|--|
| Subindex                     | Default setting                        | Value range | Access | Data type |  |
| 0                            | 0                                      | 0 ... 65535 | rw     | U32       |  |

The heartbeat telegram will be sent automatically if a value > "0" is entered in I-1017. The cycle time corresponds to the entered value in milliseconds.

"Node guarding" is deactivated when the "heartbeat time" is active.

##### 13.2.16 I-1018: Module device description

Entry of vendor ID

| Index (hex)<br><b>0x1018</b> | Name<br><b>Module device description</b> |                 |        |           |  |
|------------------------------|------------------------------------------|-----------------|--------|-----------|--|
| Subindex                     | Default setting                          | Value range     | Access | Data type |  |
| 0 Module device description  | -                                        | Module-specific | ro     | Identity  |  |
| ...                          |                                          |                 |        |           |  |
| 4                            |                                          |                 |        |           |  |

#### Subindices

| Subindex | Meaning                                                    |
|----------|------------------------------------------------------------|
| 0        | Highest subindex                                           |
| 1        | Vendor ID = ID assigned to Lenze by the organisation "CIA" |
| 2        | Product code                                               |
| 3        | Revision number                                            |
| 4        | Serial number                                              |

##### 13.2.17 I-1029: Error behaviour

| Index (hex)<br><b>0x1029</b> | Name<br><b>Error characteristics</b> |             |        |           |  |
|------------------------------|--------------------------------------|-------------|--------|-----------|--|
| Subindex                     | Default setting                      | Value range | Access | Data type |  |
| 0 No. of error classes       | -                                    | 0 ... 255   | ro     | U8        |  |
| 1 Communication error        | -                                    | 0 ... 2     | rw     | U8        |  |

#### Subindex 1

| Value of subindex 1 | Meaning                            |
|---------------------|------------------------------------|
| 0                   | Status change to "Pre-operational" |
| 1                   | No status change                   |
| 2                   | Status change to "Stopped"         |

Here, the response to communication errors (node guarding, heartbeat) can be set under subindex 1.

**13.2.18 I-1200/I-1201: Server SDO parameters**

Server SDOs can be parameterised with two objects:

- I-1200 for the parameter data channel 1 (SDO1)
- I-1201 for the parameter data channel 2 (SDO2)

With I-1201 the identifier can be written in sending and receiving direction, I-1200 has only got reading access. The server SDO parameters are only valid, if bit 31 = "0" in both transmission directions (subindex 1 and 2).

| Index  | Subindex | Name                               | Data type | Value range                 | Authorisation |
|--------|----------|------------------------------------|-----------|-----------------------------|---------------|
| I-1200 | 0        | Server SDO1 parameters             | U8        | 0 ... 255                   | ro            |
|        | 1        | Identifier<br>Client → server (rx) | U32       | 0 ... (2 <sup>32</sup> - 1) | ro            |
|        | 2        | Identifier<br>Server → client (tx) | U32       | 0 ... (2 <sup>32</sup> - 1) | ro            |
| I-1201 | 0        | Server SDO2 parameters             | U8        | 0 ... 255                   | ro            |
|        | 1        | Identifier<br>Client → server (rx) | U32       | 0 ... (2 <sup>32</sup> - 1) | rw            |
|        | 2        | Identifier<br>Server → client (tx) | U32       | 0 ... (2 <sup>32</sup> - 1) | rw            |

**Subindices**

| Subindex | Description                              |
|----------|------------------------------------------|
| 0        | Max. supported subindex = 3              |
| 1        | Specification of the receive identifier  |
| 2        | Specification of the transmit identifier |

**Data telegram assignment for subindex 1 and 2**

| Byte 8    |    |       |              | 7th byte                                                                 |  | Byte 6 |    |                   | Byte 5 |  |   |     |
|-----------|----|-------|--------------|--------------------------------------------------------------------------|--|--------|----|-------------------|--------|--|---|-----|
| MSB       |    |       |              | U32                                                                      |  |        |    |                   |        |  |   | LSB |
| 31        | 30 | 29    | 28           | ...                                                                      |  |        | 11 | 10                | ...    |  | 0 |     |
| 0/1       | 0  | 0     | Bit value: 0 |                                                                          |  |        |    | 11-bit identifier |        |  |   |     |
| Bits      |    | Value |              | Description                                                              |  |        |    |                   |        |  |   |     |
| 0 ... 10  |    | X     |              | 11-bit identifier                                                        |  |        |    |                   |        |  |   |     |
| 11 ... 29 |    | 0     |              | The extended identifier is not supported. Any of these bits must be "0". |  |        |    |                   |        |  |   |     |
| 30        |    | 0     |              | Reserved                                                                 |  |        |    |                   |        |  |   |     |
| 31        |    | 0     |              | SDO valid                                                                |  |        |    |                   |        |  |   |     |
|           |    | 1     |              | SDO invalid                                                              |  |        |    |                   |        |  |   |     |

**Example:**

Parameter data channel 2 (SDO2) of the controller with node address "4" is to be deactivated. The master must send this command to the communication module via parameter data channel 1 (SDO1).

In sending direction, the basic identifier for SDO2 has the value "1600" (basic identifier for response: "1472").

**Identifier = basic identifier + node address = 1604 (0x0644)**

For deactivating the parameter data channel (= invalid), set bit 31 in I-1201, subindices 1 and 2 to "1". After this, enter the following values under subindices 1 and 2:

I-1201/1:  $0x80000000 + 0x644 = 0x80000644$

I-1201/2:  $0x80000000 + 0x5C4 = 0x800005C4$

► User data assignment for I-1201/1:

| User data                      |                |                 |                                     |             |             |             |             |
|--------------------------------|----------------|-----------------|-------------------------------------|-------------|-------------|-------------|-------------|
| Byte 1                         | 2nd byte       | 3rd byte        | Byte 4                              | Byte 5      | Byte 6      | 7th byte    | Byte 8      |
| Command                        | Index Low byte | Index High byte | Subindex                            | U32         |             |             |             |
| <b>0x23</b><br>(write request) | <b>0x01</b>    | <b>0x12</b>     | <b>0x01</b><br>Client → server (rx) | <b>0x44</b> | <b>0x06</b> | <b>0x00</b> | <b>0x80</b> |

► User data assignment for I-1201/2:

| User data                      |                |                 |                                     |             |             |             |             |
|--------------------------------|----------------|-----------------|-------------------------------------|-------------|-------------|-------------|-------------|
| Byte 1                         | 2nd byte       | 3rd byte        | Byte 4                              | Byte 5      | Byte 6      | 7th byte    | Byte 8      |
| Command                        | Index Low byte | Index High byte | Subindex                            | U32         |             |             |             |
| <b>0x23</b><br>(write request) | <b>0x01</b>    | <b>0x12</b>     | <b>0x02</b><br>Server → client (tx) | <b>0xC4</b> | <b>0x05</b> | <b>0x00</b> | <b>0x80</b> |

**13.2.19 I-1400 ... I-1402: Receive PDO communication parameters**

Receipt of PDO communication parameters

**Note!**

The objects I-1401 and I-1402 are **not** available for 8200 vector and 93XX controllers.

| Index  | Subindex | Name                    | Data type | Value range                 | Authorisation |
|--------|----------|-------------------------|-----------|-----------------------------|---------------|
| I-1400 | 0        | Number of entries       | U8        | 0 ... 255                   | ro            |
|        | 1        | RPDO1 COB-ID            | U32       | 0 ... (2 <sup>32</sup> - 1) | rw            |
|        | 2        | RPDO1 transmission type | U8        | 0 ... 240, 254              | rw            |
|        | 5        | RPDO1 event timer       | U32       | 1 ... 65535                 | rw            |
| I-1401 | 0        | Number of entries       | U8        | 0 ... 255                   | ro            |
|        | 1        | RPDO2 COB-ID            | U32       | 0 ... (2 <sup>32</sup> - 1) | rw            |
|        | 2        | RPDO2 transmission type | U8        | 0 ... 255                   | rw            |
|        | 5        | RPDO2 event timer       | U32       | 1 ... 65535                 | rw            |
| I-1402 | 0        | Number of entries       | U8        | 0 ... 255                   | ro            |
|        | 1        | RPDO3 COB-ID            | U32       | 0 ... (2 <sup>32</sup> - 1) | rw            |
|        | 2        | RPDO3 transmission type | U8        | 0 ... 255                   | rw            |
|        | 5        | RPDO3 event timer       | U32       | 1 ... 65535                 | rw            |

**Subindices**

| Subindex | Description                                                                                                                                                                                            |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0        | Max. supported subindices = 5                                                                                                                                                                          |
| 1        | Identifier setting: <ul style="list-style-type: none"> <li>● RPDO1: 200<sub>hex</sub> + node ID</li> <li>● RPDO2: 300<sub>hex</sub> + node ID</li> <li>● RPDO3: 400<sub>hex</sub> + node ID</li> </ul> |
| 2        | Transmission type setting (see table above)                                                                                                                                                            |
| 5        | Monitoring time setting (see table above)                                                                                                                                                              |

**Description of subindex 1**

Data telegram assignment

| Byte 8    |     |       |              | 7th byte                                                                 |  | Byte 6 |    | Byte 5            |     |     |   |
|-----------|-----|-------|--------------|--------------------------------------------------------------------------|--|--------|----|-------------------|-----|-----|---|
| MSB       |     |       |              | U32                                                                      |  |        |    |                   |     | LSB |   |
| 31        | 30  | 29    | 28           | ...                                                                      |  |        | 11 | 10                | ... |     | 0 |
| 0/1       | 0/1 | 0     | Bit value: 0 |                                                                          |  |        |    | 11-bit identifier |     |     |   |
|           |     |       |              |                                                                          |  |        |    |                   |     |     |   |
| Bits      |     | Value |              | Description                                                              |  |        |    |                   |     |     |   |
| 0 ... 10  |     | X     |              | 11-bit identifier                                                        |  |        |    |                   |     |     |   |
| 11 ... 29 |     | 0     |              | The extended identifier is not supported. Any of these bits must be "0". |  |        |    |                   |     |     |   |
| 30        |     | 0     |              | RTR to RPDO possible (Lenze)                                             |  |        |    |                   |     |     |   |
|           |     | 1     |              | RTR to RPDO not possible (cannot be set)                                 |  |        |    |                   |     |     |   |
| 31        |     | 0     |              | RPDO active                                                              |  |        |    |                   |     |     |   |
|           |     | 1     |              | RPDO not active                                                          |  |        |    |                   |     |     |   |

## Description of subindex 2

| PDO transmission |             |                  | Transmission type | Description                                                         |
|------------------|-------------|------------------|-------------------|---------------------------------------------------------------------|
| cyclic           | synchronous | event-controlled |                   |                                                                     |
| X                | X           |                  | n = 1 ... 240     | By entering value n, this RPDO will be accepted by every n-th sync. |
|                  |             | X                | n = 254           | Manufacturer-specific, see code C1875 / C2375                       |

## Description of subindex 5

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 C1882.

**Note!**

- ▶ The monitoring time starts with the arrival of the first telegram.
- ▶ Subindex 5 = 0: Monitoring deactivated.

**13.2.20 I-1600 ... I-1602: Receive PDO mapping parameters**

With these objects, parameter data can be received as PDOs.

**Note!**

The objects I-1601 and I-1602 are **not** available for 8200 vector and 93XX controllers.

| Index  | Subindex | Name                              | Data type | Value range                 | Authorisation |
|--------|----------|-----------------------------------|-----------|-----------------------------|---------------|
| I-1600 | 0        | Number of mapped objects in RPDOs | U8        | 0 ... 255                   | rw            |
|        | 1        | RPDO mapping 1                    | U32       | 0 ... (2 <sup>32</sup> - 1) |               |
|        | 2        | RPDO mapping 2                    |           |                             |               |
|        | 3        | RPDO mapping 3                    |           |                             |               |
|        | 4        | RPDO mapping 4                    |           |                             |               |
| I-1601 | 0        | Number of mapped objects in RPDOs | U8        | 0 ... 255                   | rw            |
|        | 1        | RPDO mapping 1                    | U32       | 0 ... (2 <sup>32</sup> - 1) |               |
|        | 2        | RPDO mapping 2                    |           |                             |               |
|        | 3        | RPDO mapping 3                    |           |                             |               |
|        | 4        | RPDO mapping 4                    |           |                             |               |
| I-1602 | 0        | Number of mapped objects in RPDOs | U8        | 0 ... 255                   | rw            |
|        | 1        | RPDO mapping 1                    | U32       | 0 ... (2 <sup>32</sup> - 1) |               |
|        | 2        | RPDO mapping 2                    |           |                             |               |
|        | 3        | RPDO mapping 3                    |           |                             |               |
|        | 4        | RPDO mapping 4                    |           |                             |               |

**Subindices**

| Subindex | Description                                                           |
|----------|-----------------------------------------------------------------------|
| 0        | Number of mapped objects in RPDOs                                     |
| 1 ... 4  | This value will be returned in case of a read request for the object. |

For mapping in the module, the master enters indices and data lengths.

**Mapping structure**

| MSB   |     |    |    |          |   |   |                                                | LSB |   |
|-------|-----|----|----|----------|---|---|------------------------------------------------|-----|---|
| 31    | ... | 16 | 15 | ...      | 8 | 7 | ...                                            | 0   | 0 |
| Index |     |    |    | Subindex |   |   | Object length<br>DWord 0 = 0x20<br>Word = 0x10 |     |   |

The EDS file for the EMF2178IB communication module supports the mapping.

**Tip!**

The current EDS file required for configuring the EMF2178IB (CANopen) communication module can be found in the download area on:

**[www.Lenze.com](http://www.Lenze.com)**

## 13.2.21 I-1800 ... I-1802: Transmit PDO communication parameters

Sending of PDO communication parameters

**Note!**

The objects I-1801 and I-1802 are **not** available for 8200 vector and 93XX controllers.

| Index  | Subindex | Name                    | Data type | Value range            | Authorisation |
|--------|----------|-------------------------|-----------|------------------------|---------------|
| I-1800 | 0        | Number of entries       | U8        | 0 ... 255              | ro            |
|        | 1        | TPDO1 COB-ID            | U32       | 0 ... ( $2^{32} - 1$ ) | rw            |
|        | 2        | TPDO1 transmission type | U8        | 0 ... 240, 252, 254    | rw            |
|        | 3        | TPDO1 inhibit time      | U32       | 0 ... 65535            | rw            |
|        | 5        | TPDO1 event timer       | U32       | 0 ... 65535            | rw            |
| I-1801 | 0        | Number of entries       | U8        | 0 ... 255              | ro            |
|        | 1        | TPDO2 COB-ID            | U32       | 0 ... ( $2^{32} - 1$ ) | rw            |
|        | 2        | TPDO2 transmission type | U8        | 0 ... 240, 252, 254    | rw            |
|        | 3        | TPDO2 inhibit time      | U32       | 0 ... 65535            | rw            |
|        | 5        | TPDO2 event timer       | U32       | 0 ... 65535            | rw            |
| I-1802 | 0        | Number of entries       | U8        | 0 ... 255              | ro            |
|        | 1        | TPDO3 COB-ID            | U32       | 0 ... ( $2^{32} - 1$ ) | rw            |
|        | 2        | TPDO3 transmission type | U8        | 0 ... 240, 252, 254    | rw            |
|        | 3        | TPDO3 inhibit time      | U32       | 0 ... 65535            | rw            |
|        | 5        | TPDO3 event timer       | U32       | 0 ... 65535            | rw            |

**Subindices**

| Subindex | Description                                                                                                                                                        |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0        | Max. supported subindices = 5                                                                                                                                      |
| 1        | Identifier setting: <ul style="list-style-type: none"> <li>● TPDO1: 0x180 + node ID</li> <li>● TPDO2: 0x280 + node ID</li> <li>● TPDO3: 0x380 + node ID</li> </ul> |
| 2        | Transmission type setting (see table)                                                                                                                              |
| 3        | Inhibit time setting                                                                                                                                               |
| 5        | Cycle time setting                                                                                                                                                 |

**Description of subindex 1**

Data telegram assignment

| Byte 8    |     |       |              | 7th byte                                                                 |  |  |  | Byte 6 |                   |     |  | Byte 5 |  |   |  |     |  |  |  |
|-----------|-----|-------|--------------|--------------------------------------------------------------------------|--|--|--|--------|-------------------|-----|--|--------|--|---|--|-----|--|--|--|
| MSB       |     |       |              | U32                                                                      |  |  |  |        |                   |     |  |        |  |   |  | LSB |  |  |  |
| 31        | 30  | 29    | 28           | ...                                                                      |  |  |  | 11     | 10                | ... |  |        |  | 0 |  |     |  |  |  |
| 0/1       | 0/1 | 0     | Bit value: 0 |                                                                          |  |  |  |        | 11-bit identifier |     |  |        |  |   |  |     |  |  |  |
|           |     |       |              |                                                                          |  |  |  |        |                   |     |  |        |  |   |  |     |  |  |  |
| Bits      |     | Value |              | Description                                                              |  |  |  |        |                   |     |  |        |  |   |  |     |  |  |  |
| 0 ... 10  |     | X     |              | 11-bit identifier                                                        |  |  |  |        |                   |     |  |        |  |   |  |     |  |  |  |
| 11 ... 29 |     | 0     |              | The extended identifier is not supported. Any of these bits must be "0". |  |  |  |        |                   |     |  |        |  |   |  |     |  |  |  |
| 30        |     | 0     |              | RTR to TPDO possible (Lenze)                                             |  |  |  |        |                   |     |  |        |  |   |  |     |  |  |  |
|           |     | 1     |              | RTR to TPDO not possible (cannot be set)                                 |  |  |  |        |                   |     |  |        |  |   |  |     |  |  |  |
| 31        |     | 0     |              | TPDO active                                                              |  |  |  |        |                   |     |  |        |  |   |  |     |  |  |  |
|           |     | 1     |              | TPDO not active                                                          |  |  |  |        |                   |     |  |        |  |   |  |     |  |  |  |

**Description of subindex 2**

| PDO transmission |             |                  | Transmission type | Description                                                         |
|------------------|-------------|------------------|-------------------|---------------------------------------------------------------------|
| cyclic           | synchronous | event-controlled |                   |                                                                     |
| X                | X           |                  | n = 1 ... 240     | By entering value n, this TPDO will be accepted by every n-th sync. |
|                  | X           |                  | n = 252           | By sync, TPDO will be filled with new data, but only sent on RTR.   |
|                  |             | X                | n = 254           | Manufacturer-specific, see code C1875 / C2375                       |

**Description of subindex 3**

Minimum time between two process data telegrams.

**Note!**

Subindex 3 = 0: Inhibit time deactivated.

**Description of subindex 5**

With event-controlled/cyclic operation, you can set the cycle time for sending PDOs via the CAN bus.

**Note!**

Subindex 5 = 0: Cyclic PDO sending deactivated.

## 13.2.22 I-1A00 ... I-1A02: Transmit PDO mapping parameters

With these objects, parameter data can be sent as PDOs.

**Note!**

The objects I-1A01 and I-1A02 are **not** available for 8200 vector and 93XX controllers.

| Index  | Subindex | Name                              | Data type | Value range                 | Authorisation |
|--------|----------|-----------------------------------|-----------|-----------------------------|---------------|
| I-1A00 | 0        | Number of mapped objects in TPDOs | U8        | 0 ... 255                   | rw            |
|        | 1        | TPDO mapping 1                    | U32       | 0 ... (2 <sup>32</sup> - 1) |               |
|        | 2        | TPDO mapping 2                    |           |                             |               |
|        | 3        | TPDO mapping 3                    |           |                             |               |
|        | 4        | TPDO mapping 4                    |           |                             |               |
| I-1A01 | 0        | Number of mapped objects in TPDOs | U8        | 0 ... 255                   | rw            |
|        | 1        | TPDO mapping 1                    | U32       | 0 ... (2 <sup>32</sup> - 1) |               |
|        | 2        | TPDO mapping 2                    |           |                             |               |
|        | 3        | TPDO mapping 3                    |           |                             |               |
|        | 4        | TPDO mapping 4                    |           |                             |               |
| I-1A02 | 0        | Number of mapped objects in TPDOs | U8        | 0 ... 255                   | rw            |
|        | 1        | TPDO mapping 1                    | U32       | 0 ... (2 <sup>32</sup> - 1) |               |
|        | 2        | TPDO mapping 2                    |           |                             |               |
|        | 3        | TPDO mapping 3                    |           |                             |               |
|        | 4        | TPDO mapping 4                    |           |                             |               |

Detailed information on mapping: (📖 65)

## 14 Codes

### 14.1 Overview



#### Note!

- ▶ Some CANopen objects have a direct influence on the corresponding codes. If a new value is written within an index, the value is also adopted in the corresponding code C18xx or C23xx.
- ▶ With the 9300 Servo PLC, Drive PLC and ECS, a change of code C23xx will only become effective after one of the following actions:
  - Fresh power-on of communication module.
  - Sending the NMT command "Reset Node" / "Reset Communication".
  - Entering the value "1" under code C2120 (AIF control byte).
- ▶ With all other controllers, changes will become effective immediately.
- ▶ Since the codes C18xx are in the module, the corresponding values are directly known. The codes C23xx are in the PLC and therefore must be read by the module first.

| Code  | Subcode | Index<br>(dec = hex) | Designation                                  | Info |
|-------|---------|----------------------|----------------------------------------------|------|
| C0001 | -       | 24574 =<br>0x5FFE    | Operating mode for 8200 vector               | 144  |
| C0009 | -       | 24566 =<br>0x5FF6    | CAN node address                             | 144  |
| C0046 | -       | 24529 =<br>0x5FD1    | Display of frequency setpoint                | 144  |
| C0125 | -       | 24450 =<br>0x5F82    | Baud rate                                    | 145  |
| C0126 | -       | 24449 =<br>0x5F81    | Behaviour with communication error           | 145  |
| C0135 | -       | 24440 =<br>0x5F78    | Controller control word                      | 145  |
| C0150 | -       | 24425 =<br>0x5F69    | Controller status word                       | 145  |
| C0161 | -       | 24414 =<br>0x5F5E    | Fault memory 8200 vector: Active fault       | 146  |
| C0162 | -       | 24413 =<br>0x5F5D    | Fault memory 8200 vector: Last fault         | 146  |
| C0163 | -       | 24412 =<br>0x5F5C    | Fault memory 8200 vector: Last but one fault | 146  |
| C0164 | -       | 24411 =<br>0x5F5B    | Fault memory 8200 vector: Last but two fault | 146  |
| C0168 | -       | 24407 =<br>0x5F57    | Fault memory 9300 / ECSXX                    | 146  |
| C1810 | -       | 22765 =<br>0x58ED    | Software identification code as string       | 120  |
| C1811 | -       | 22764 =<br>0x58EC    | Software creation date as string             | 120  |
| C1812 | 1 ... 4 | 22763 =<br>0x58EB    | Software identification code in subcodes     | 120  |

| Code  | Subcode  | Index<br>(dec = hex) | Designation                                              | Info                                                                                      |
|-------|----------|----------------------|----------------------------------------------------------|-------------------------------------------------------------------------------------------|
| C1813 | 1 ... 4  | 22762 =<br>0x58EA    | Software creation date in subcodes                       |  120   |
| C1822 | 1 ... 12 | 22753 =<br>0x58E1    | AIF input words                                          |  121   |
| C1823 | 1 ... 12 | 22752 =<br>0x58E0    | AIF output words                                         |  121   |
| C1824 | 1 ... 11 | 22751 =<br>0x58DF    | AIF input double words                                   |  121   |
| C1825 | 1 ... 11 | 22750 =<br>0x58DE    | AIF output double words                                  |  121   |
| C1827 | -        | 22748 =<br>0x58DC    | Guard time                                               |  122   |
| C1828 | -        | 22747 =<br>0x58DB    | Life time factor                                         |  122   |
| C1850 | -        | 22725 =<br>0x58C5    | CAN node address                                         |  123   |
| C1851 | -        | 22724 =<br>0x58C4    | Baud rate                                                |  124   |
| C1852 | -        | 22723 =<br>0x58C3    | Master / slave operation                                 |  125   |
| C1853 | 1...3    | 22722 =<br>0x58C2    | Addressing CAN-INx/CAN-OUTx                              |  126   |
| C1854 | 1 ... 6  | 22721 =<br>0x58C1    | Individual addressing CAN-INx / CAN-OUTx                 |  129  |
| C1855 | 1 ... 6  | 22720 =<br>0x58C0    | Display of resulting identifiers CAN-IN / CAN-OUT        |  129 |
| C1856 | 1 ... 5  | 22719 =<br>0x58BF    | Boot up and cycle times                                  |  130 |
| C1857 | 1 ... 6  | 22718 =<br>0x58BE    | Monitoring times                                         |  131 |
| C1859 | -        | 22716 =<br>0x58BC    | Display of the DIP switch position during initialisation |  132 |
| C1860 | -        | 22715 =<br>0x58BB    | Display of the current DIP switch position               |  133 |
| C1861 | -        | 22714 =<br>0x58BA    | Display of the active CAN node address                   |  133 |
| C1862 | -        | 22713 =<br>0x58B9    | Display of the active baud rate                          |  133 |
| C1864 | -        | 22711 =<br>0x58B7    | PDO transmission with status change to "Operational"     |  134 |
| C1865 | 1 ... 4  | 22710 =<br>0x58B6    | Validity of SDO2 and PDOs                                |  134 |
| C1867 | -        | 22708 =<br>0x58B4    | Sync Rx identifier                                       |  135 |
| C1868 | -        | 22707 =<br>0x58B3    | Sync Tx identifier                                       |  135 |
| C1869 | -        | 22706 =<br>0x58B2    | Consumer heartbeat ID                                    |  136 |
| C1870 | 1 ... 2  | 22705 =<br>0x58B1    | Heartbeat time (consumer, producer)                      |  136 |
| C1871 | -        | 22704 =<br>0x58B0    | Emergency ID                                             |  137 |
| C1872 | -        | 22703 =<br>0x58AF    | Emergency inhibit time                                   |  137 |

| Code  | Subcode | Index<br>(dec = hex) | Designation                                              | Info |
|-------|---------|----------------------|----------------------------------------------------------|------|
| C1873 | 1...3   | 22702 =<br>0x58AE    | Sync rate CAN-IN1 ... CAN IN3                            | 138  |
| C1874 | 1...3   | 22701 =<br>0x58AD    | Sync rate CAN-OUT1 ... CAN OUT3                          | 139  |
| C1875 | 1...3   | 22700 =<br>0x58AC    | Tx mode CAN-OUT1 ... CAN OUT3                            | 140  |
| C1876 | 1 ... 4 | 22699 =<br>0x58AB    | Masks CAN-OUT1                                           | 142  |
| C1877 | 1 ... 4 | 22698 =<br>0x58AA    | Masks CAN-OUT2                                           | 143  |
| C1878 | 1 ... 4 | 22697 =<br>0x58A9    | Masks CAN-OUT3                                           | 143  |
| C1882 | 1 ... 6 | 22693 =<br>0x58A5    | Monitoring responses                                     | 143  |
| C2120 | -       | 22455 =<br>0x57B7    | AIF control byte                                         | 147  |
| C2121 | -       | 22454 =<br>0x57B6    | AIF status byte                                          | 148  |
| C2327 | -       | 22248 =<br>0x56E8    | Guard time                                               | 122  |
| C2328 | -       | 22247 =<br>0x56E7    | Life time factor                                         | 122  |
| C2350 | -       | 22225 =<br>0x56D1    | CAN node address                                         | 123  |
| C2351 | -       | 22224 =<br>0x56D0    | Baud rate                                                | 124  |
| C2352 | -       | 22223 =<br>0x56CF    | Master / slave operation                                 | 125  |
| C2353 | 1...3   | 22222 =<br>0x56CE    | Addressing CAN-INx/CAN-OUTx                              | 126  |
| C2354 | 1 ... 6 | 22221 =<br>0x56CD    | Individual addressing CAN-INx / CAN-OUTx                 | 129  |
| C2355 | 1 ... 6 | 22220 =<br>0x56CC    | Display of resulting identifiers CAN-IN / CAN-OUT        | 129  |
| C2356 | 1 ... 5 | 22219 =<br>0x56CB    | Boot up and cycle times                                  | 130  |
| C2357 | 1 ... 6 | 22218 =<br>0x56CA    | Monitoring time                                          | 131  |
| C2359 | -       | 22216 =<br>0x56C8    | Display of the DIP switch position during initialisation | 132  |
| C2364 | -       | 22211 =<br>0x56C3    | PDO transmission with status change to "Operational"     | 134  |
| C2365 | 1 ... 4 | 22210 =<br>0x56C2    | Validity of SDO2 and PDOs                                | 134  |
| C2367 | -       | 22208 =<br>0x56C0    | Sync Rx identifier                                       | 135  |
| C2368 | -       | 22207 =<br>0x56BF    | Sync Tx identifier                                       | 135  |
| C2369 | -       | 22206 =<br>0x56BE    | Consumer heartbeat ID                                    | 136  |
| C2370 | 1 ... 2 | 22205 =<br>0x56BD    | Heartbeat time (consumer, producer)                      | 136  |
| C2371 | -       | 22204 =<br>0x56BC    | Emergency ID                                             | 137  |

| Code  | Subcode | Index<br>(dec = hex) | Designation                     | Info                                                                                    |
|-------|---------|----------------------|---------------------------------|-----------------------------------------------------------------------------------------|
| C2372 | -       | 22203 =<br>0x56BB    | Emergency inhibit time          |  137 |
| C2373 | 1...3   | 22202 =<br>0x56BA    | Sync rate CAN-IN1 ... CAN IN3   |  138 |
| C2374 | 1...3   | 22201 =<br>0x56B9    | Sync rate CAN-OUT1 ... CAN OUT3 |  139 |
| C2375 | 1...3   | 22200 =<br>0x56B8    | Tx mode CAN-OUT1 ... CAN OUT3   |  140 |
| C2376 | 1 ... 4 | 22199 =<br>0x56B7    | Masks CAN-OUT1                  |  142 |
| C2377 | 1 ... 4 | 22198 =<br>0x56B6    | Masks CAN-OUT2                  |  143 |
| C2378 | 1 ... 4 | 22197 =<br>0x56B5    | Masks CAN-OUT3                  |  143 |
| C2382 | 1 ... 6 | 22193 =<br>0x56B1    | Monitoring response             |  143 |

## 14.2 Communication-relevant Lenze codes

The configuration of the entire system where the individual controller often only represents one of many components in general is defined or altered during the commissioning phase or during service.

For configuring the Lenze controllers, parameters are provided which are stored as so-called "codes" in a numerically ascending order in the memory of the controller.

All codes can be accessed via the CAN bus.

The following codes are available for communication via the CAN bus:

- ▶ Codes C1810 ... C1825 and C1860 ... C1862:
  - For all devices.
  - These codes are stored in the EMF2178IB communication module.
- ▶ Codes C1827 ... C1882
  - For 8200 vector and 93XX controllers.
  - These codes are stored in the EMF2178IB communication module.
- ▶ Codes C2327 ... C2382:
  - For 9300 Servo PLC, Drive PLC and ECS.
  - These codes are stored in the controller.

Communication with the drive requires that the system identifies the drive as node. The identification takes place in the initialisation phase of the communication module.

### How to read the tables

| Column    | Meaning                                                 |                                                             |
|-----------|---------------------------------------------------------|-------------------------------------------------------------|
| Code      | (Lenze) code                                            |                                                             |
| Subcode   | Subcode                                                 |                                                             |
| Index     | Information on the addressing of the code               |                                                             |
| Lenze     | Lenze setting of code                                   |                                                             |
|           | <input type="checkbox"/> Disp                           | Display code<br>Configuration of this code is not possible. |
| Selection | Minimum value                                           | [minimum increment/unit] maximum value                      |
|           | For a display code, the displayed values are specified. |                                                             |
| Data type | FIX32                                                   | 32-bit value with sign; decimal with 4 decimal positions    |
|           | S8                                                      | 8-bit value, with sign                                      |
|           | S16                                                     | 16-bit value, with sign                                     |
|           | S32                                                     | 32-bit value, with sign                                     |
|           | U8                                                      | 8-bit value, without sign                                   |
|           | U16                                                     | 16-bit value, without sign                                  |
|           | U32                                                     | 32-bit value, without sign                                  |
|           | VS                                                      | Visible string, character string with defined length        |

**C1810:  
Software ID**

| Code  | Subcode | Index             | Possible settings             |           | Data type |
|-------|---------|-------------------|-------------------------------|-----------|-----------|
|       |         |                   | Lenze                         | Selection |           |
| C1810 | -       | 22765 =<br>0x58ED | <input type="checkbox"/> Disp |           | VS        |

Software identification code as string

Only important in the event of service.

**C1811:  
Software creation date**

| Code  | Subcode | Index             | Possible settings             |           | Data type |
|-------|---------|-------------------|-------------------------------|-----------|-----------|
|       |         |                   | Lenze                         | Selection |           |
| C1811 | -       | 22764 =<br>0x58EC | <input type="checkbox"/> Disp |           | VS        |

Software creation date as string

Mainly important in the event of service.

**C1812:  
Software ID**

| Code  | Subcode           | Index             | Possible settings             |           | Data type |
|-------|-------------------|-------------------|-------------------------------|-----------|-----------|
|       |                   |                   | Lenze                         | Selection |           |
| C1812 | /1:<br>...<br>/4: | 22763 =<br>0x58EB | <input type="checkbox"/> Disp |           | U32       |

Software identification code in subcodes

**C1813:  
Software creation date**

| Code  | Subcode           | Index             | Possible settings             |           | Data type |
|-------|-------------------|-------------------|-------------------------------|-----------|-----------|
|       |                   |                   | Lenze                         | Selection |           |
| C1813 | /1:<br>...<br>/4: | 22762 =<br>0x58EA | <input type="checkbox"/> Disp |           | U32       |

Software creation date in subcodes

**C1822:  
AIF-IN all words**

| Code  | Subcode            | Index             | Possible settings             |           | Data type |
|-------|--------------------|-------------------|-------------------------------|-----------|-----------|
|       |                    |                   | Lenze                         | Selection |           |
| C1822 | /1:<br>...<br>/12: | 22753 =<br>0x58E1 | <input type="checkbox"/> Disp |           | U16       |

All AIF-IN words as 16-bit values

**C1823:  
AIF-OUT all words**

| Code  | Subcode            | Index             | Possible settings             |           | Data type |
|-------|--------------------|-------------------|-------------------------------|-----------|-----------|
|       |                    |                   | Lenze                         | Selection |           |
| C1823 | /1:<br>...<br>/12: | 22752 =<br>0x58E0 | <input type="checkbox"/> Disp |           | U16       |

All AIF-OUT words as 16-bit values

**C1824:  
AIF-IN all double words**

| Code  | Subcode            | Index             | Possible settings             |           | Data type |
|-------|--------------------|-------------------|-------------------------------|-----------|-----------|
|       |                    |                   | Lenze                         | Selection |           |
| C1824 | /1:<br>...<br>/11: | 22751 =<br>0x58DF | <input type="checkbox"/> Disp |           | U32       |

All AIF-IN double words as 32-bit values, with an offset of 16 bits each.

( 66)

**C1825:  
AIF-OUT all double words**

| Code  | Subcode            | Index             | Possible settings             |           | Data type |
|-------|--------------------|-------------------|-------------------------------|-----------|-----------|
|       |                    |                   | Lenze                         | Selection |           |
| C1825 | /1:<br>...<br>/11: | 22750 =<br>0x58DE | <input type="checkbox"/> Disp |           | U32       |

All AIF-OUT double words as 32-bit values, with an offset of 16 bits each.

( 66)

**C1827/C2327:  
Guard Time**

| Code  | Subcode | Index                          | Possible settings |           |       | Data type |
|-------|---------|--------------------------------|-------------------|-----------|-------|-----------|
|       |         |                                | Lenze             | Selection |       |           |
| C1827 | -       | 22748 <sub>d</sub> =<br>0x58DC | 0                 | 0 [1 ms]  | 65535 | FIX32     |
| C2327 | -       | 22248 <sub>d</sub> =<br>0x56E8 |                   |           |       |           |

It is possible to select a different "node life time" for each NMT slave.

The "node life time" is the product of "guard time" (object I-100C) and the "life time factor" (object I-100D). The NMT master must know the two values.

**Relation to CANopen**

The object I-100C (guard time) has a direct influence on code C1827/C2327. A newly written value in the object is also transferred to code C1827/C2327.

**C1828/C2328:  
Life time factor**

| Code  | Subcode | Index             | Possible settings |           |     | Data type |
|-------|---------|-------------------|-------------------|-----------|-----|-----------|
|       |         |                   | Lenze             | Selection |     |           |
| C1828 | -       | 22747 =<br>0x58DB | 0                 | 0 [1]     | 255 | FIX32     |
| C2328 | -       | 22247 =<br>0x56E7 |                   |           |     |           |

It is possible to select a different "node life time" for each NMT slave.

The "node life time" is the product of "guard time" (object I-100C) and the "life time factor" (object I-100D). The NMT master must know the two values.

**Relation to CANopen**

The object I-100D (life time factor) has a direct influence on code C1828/C2328. A newly written value in the object is also transferred to code C1828/C2328.

**C1850/C2350:  
CAN node address**

| Code  | Subcode | Index             | Possible settings |           | Data type |
|-------|---------|-------------------|-------------------|-----------|-----------|
|       |         |                   | Lenze             | Selection |           |
| C1850 | -       | 22725 =<br>0x58C5 | 1                 | 1 [1]     | 99 FIX32  |
| C2350 | -       | 22225 =<br>0x56D1 |                   |           |           |

This code serves to set the CAN node address of the communication module.

**Note!**

- ▶ If several CAN nodes are interconnected, their node addresses must differ from each other.
- ▶ Node addresses > 99 can only be set via DIP switch (33).
- ▶ All DIP switches 1 ... 7 = OFF (Lenze setting):
  - When the device is switched on, the settings under code C1850/C2350 (node address) and C1851/C2351 (baud rate) are active.
  - Code C1850/C2350 is an image of code C0009 of the basic device. Setting C1850/C2350 has a direct effect on C0009.

**C1851/C2351:****Baud rate**

| Code  | Subcode | Index             | Possible settings |           |                                           | Data type |
|-------|---------|-------------------|-------------------|-----------|-------------------------------------------|-----------|
|       |         |                   | Lenze             | Selection |                                           |           |
| C1851 | -       | 22724 =<br>0x58C4 | 0                 | 0 ... 4   | 0: 500 kbps<br>1: 250 kbps<br>2: 125 kbps | FIX32     |
| C2351 | -       | 22224 =<br>0x56D0 |                   |           | 3: 50 kbps<br>4: 1000 kbit/s              |           |

This code serves to set the baud rate of the communication module.

- ▶ The baud rate must be the same for all CAN nodes.
- ▶ The baud rates 10 kbps and 20 kbps can only be selected via DIP switch(📖 33).
- ▶ Setting the code (e.g. with GDC via CAN) has a direct effect on the standard device code C0125.

**Note!**

- ▶ The baud rate must be the same for all CAN nodes.
- ▶ The baud rates 10 kbps and 20 kbps can only be selected via DIP switch(📖 33).
- ▶ All DIP switches 1 ... 7 = OFF (Lenze setting):
  - When the device is switched on, the settings under code C1850/C2350 (node address) and C1851/C2351 (baud rate) are active.
  - Code C1851/C2351 is an image of code C0125 of the basic drive. Setting C1851/C2351 has a direct effect on C0125.

**C1852/C2352:  
Master/slave operation**

| Code  | Subcode | Index             | Possible settings |                                           | Data type |
|-------|---------|-------------------|-------------------|-------------------------------------------|-----------|
|       |         |                   | Lenze             | Selection                                 |           |
| C1852 | -       | 22723 =<br>0x58C3 | 0                 | 0: Slave operation<br>1: Master operation | FIX32     |
| C2352 | -       | 22223 =<br>0x56CF |                   |                                           |           |

After switch-on the communication module has the state "Pre-operational". In this state only an exchange of parameter data (SDOs) is possible.

In slave operation, the communication module stays in this state until it is put by the NMT command "Start remote node" into the state "Operational".

In the state "Operational" also process data (PDOs) are exchanged besides parameter data (SDOs).

In master operation, the NMT command "Start remote node" is transmitted after an adjustable boot-up time (C1856/1, C2356/1), which puts all CAN nodes into the state "Operational".

### C1853/C2353: Addressing CAN-IN/OUT

| Code  | Subcode                                      | Index          | Possible settings |                 | Data type |
|-------|----------------------------------------------|----------------|-------------------|-----------------|-----------|
|       |                                              |                | Lenze             | Selection       |           |
| C1853 | /1: CAN-IN1/CAN-OUT1<br>/2: CAN-IN2/CAN-OUT2 | 22722 = 0x58C2 | 0                 | see table below | FIX32     |
| C2353 | /3: CAN-IN3/CAN-OUT3                         | 22222 = 0x56CE |                   |                 |           |

#### Possible settings

| Selection | Description                                      |
|-----------|--------------------------------------------------|
| 0         | Addressing to CANopen (default identifier)       |
| 1         | Addressing to C1854/C2354 (selective addressing) |
| 2         | Addressing to Lenze system bus (CAN)             |
| 3         | Addressing to CANopen objects I-140X/I-180X      |

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

### Relation to CANopen

The addressing of the corresponding PDP pair or subcode in code C1853/C2353 is switched to the objects I-140X/I-180X (see above: Selection "3") if a new value is entered into the objects I-1400 ... I-1402 or I-1800 ... I-1802.

### Identifier for addressing to CANopen (default identifier)

This is the Lenze setting of the EMF2178IB communication module. The calculation consists of the basic identifier and the node address. The basic identifier corresponds to the preset value according to DS301 V4.02.

| PDO      | Identifier                                                       |                      |
|----------|------------------------------------------------------------------|----------------------|
|          | C1853/C2353 = 0: Addressing following CANopen (basic identifier) |                      |
| CAN-IN1  | 512 + node address                                               | 0x200 + node address |
| CAN-IN2  | 768 + node address                                               | 0x300 + node address |
| CAN-IN3  | 1024 + node address                                              | 0x400 + node address |
| CAN-OUT1 | 384 + node address                                               | 0x180 + node address |
| CAN-OUT2 | 640 + node address                                               | 0x280 + node address |
| CAN-OUT3 | 896 + node address                                               | 0x380 + node address |

**Identifier for selective addressing via C1854/C2354**

In case of this addressing, the identifier is the sum of a fixed basic identifier 384 (0x180) + the value of the corresponding subcode of C1854/C2354. Here, the node address has no influence anymore.

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

| PDO      | Identifier for addressing via C1854/C2354 |                                         |       |
|----------|-------------------------------------------|-----------------------------------------|-------|
|          | C1853/C2353 = 1                           | C1853/C2353 = 1<br>(with Lenze setting) |       |
| CAN-IN1  | 384 + C1854/1 or 384 + C2354/1            | 384 + 129                               | 0x201 |
| CAN-IN2  | 384 + C1854/3 or 384 + C2354/3            | 384 + 257                               | 0x281 |
| CAN-IN3  | 384 + C1854/5 or 384 + C2354/5            | 384 + 385                               | 0x301 |
| CAN-OUT1 | 384 + C1854/2 or 384 + C1854/2            | 384 + 1                                 | 0x181 |
| CAN-OUT2 | 384 + C1854/4 or 384 + C1854/4            | 384 + 258                               | 0x282 |
| CAN-OUT3 | 384 + C1854/6 or 384 + C1854/6            | 384 + 386                               | 0x302 |

**Identifier for addressing to Lenze system bus**

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

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

| PDO      | Identifier for addressing via Lenze system bus (CAN) |                            |
|----------|------------------------------------------------------|----------------------------|
|          | C1853/C2353 = 2                                      |                            |
| CAN-IN1  | 512 + node address                                   | 0x200 + node address (hex) |
| CAN-IN2  | 640 + node address                                   | 0x280 + node address (hex) |
| CAN-IN3  | 768 + node address                                   | 0x300 + node address (hex) |
| CAN-OUT1 | 384 + node address                                   | 0x180 + node address (hex) |
| CAN-OUT2 | 641 + node address                                   | 0x281 + node address (hex) |
| CAN-OUT3 | 769 + node address                                   | 0x301 + node address (hex) |

**Identifier for addressing to CANopen projects I-140X/I-180X**

If the subcode has the value "3", this indicates that the identifiers have been changed via the objects I-140X/I-180X. Now, the identifier is developed directly from the objects.

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

| PDO      | Identifier for addressing to CANopen projects I-140X/I-180X |
|----------|-------------------------------------------------------------|
|          | C1853/C2353 = 3                                             |
| CAN-IN1  | Object I-1400, subindex 1                                   |
| CAN-IN2  | Object I-1401, subindex 1                                   |
| CAN-IN3  | Object I-1402, subindex 1                                   |
| CAN-OUT1 | Object I-1800, subindex 1                                   |
| CAN-OUT2 | Object I-1801, subindex 1                                   |
| CAN-OUT3 | Object I-1802, subindex 1                                   |

### C1854/C2354: Individual addressing CAN-IN/OUT

| Code  | Subcode       | Index             | Possible settings |            |  | Data type |
|-------|---------------|-------------------|-------------------|------------|--|-----------|
|       |               |                   | Lenze             | Selection  |  |           |
| C1854 | /1: CAN-IN1   | 22721 =<br>0x58C1 | /1: 129           | 0 [1] 1663 |  | FIX32     |
|       | /2: CAN-OUT1  |                   | /2: 1             |            |  |           |
|       | /3*: CAN-IN2  |                   | /3: 257*          |            |  |           |
| C2354 | /4*: CAN-OUT2 | 22221 =<br>0x56CD | /4: 258*          |            |  |           |
|       | /5*: CAN-IN3  |                   | /5: 385*          |            |  |           |
|       | /6*: CAN-OUT3 |                   | /6: 386*          |            |  |           |

\*) not effective when using 8200 vector or 93XXcontrollers

With this code it is possible to set the addresses of the input and output PDOs individually via six subcodes.

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

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

| Code  | Subcode       | Index             | Possible settings |           | Data type |
|-------|---------------|-------------------|-------------------|-----------|-----------|
|       |               |                   | Lenze             | Selection |           |
| C1855 | /1: CAN-IN1   | 22720 =<br>0x58C0 | Disp              |           | FIX32     |
|       | /2: CAN-OUT1  |                   |                   |           |           |
|       | /3*: CAN-IN2  |                   |                   |           |           |
| C2355 | /4*: CAN-OUT2 | 22220 =<br>0x56CC |                   |           |           |
|       | /5*: CAN-IN3  |                   |                   |           |           |
|       | /6*: CAN-OUT3 |                   |                   |           |           |

\*) not effective when using 8200 vector or 93XXcontrollers

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



#### Note!

In case of 9300 Servo PLC, Drive PLC and ECS, the value will be transferred to the standard device when the communication module is initialised.

**C1856/C2356:****Boot-up- and cycle times**

| Code  | Subcode                 | Index             | Possible settings |           |        | Data type |       |
|-------|-------------------------|-------------------|-------------------|-----------|--------|-----------|-------|
|       |                         |                   | Lenze             | Selection |        |           |       |
| C1856 | /1: Boot-up time        | 22719 =<br>0x58BF | 0 ms              | 0         | [1 ms] | 65535     | FIX32 |
|       | /2: Cycle time CAN-OUT1 |                   |                   |           |        |           |       |
| C2356 | /3: Cycle time CAN-OUT2 | 22219 =<br>0x56CB |                   |           |        |           |       |
|       | /4: Cycle time CAN-OUT3 |                   |                   |           |        |           |       |
|       | /5: Sync Tx cycle time  |                   |                   |           |        |           |       |

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 code C1875/C2375, "Tx mode" for CAN-OUT1 ... 3).

**Note!**

The value "0" deactivates the cyclic sending of the PDO.

"Sync Tx cycle time" (C1856/5, C2356/5) describes the interval time (in ms) that is required for sending a sync telegram to the CAN bus.

**Relation to CANopen**

- ▶ The object I-1006 (communication cycle period) has a direct influence on subcode C1856/5 or C2356/5.
  - As the data processing speed of the EMF2178IB communication module amounts to 1000 µs, the entry via the object I-1006 is rounded to an integer multiple of 1000 µs and stored under C1856/5 or C2356/5 (depending on the standard device).
  - When the object I-1006 is read, the content of the corresponding subcode (in µs) is returned as a response.
  - When a value is entered under subcode C1856/5 or C2356/5, bit 30 of the object I-1005 (COB-ID SYNC message) is set automatically and the communication module sends a sync telegram.
- ▶ The objects I-1800/5 ... I-1802/5 (TPDO Event Timer) have a direct influence on the subcodes C1856/2 ... 4 or C2356/2 ... 4. Newly entered values in the objects I-1800/5 ... I-1802/5 are also transferred to the subcodes C1856/2 ... 4 or C2356/2 ... 4.

**C1857/C2357:  
Monitoring time**

| Code  | Subcode                                                                               | Index             | Possible settings     |           |        | Data type |       |
|-------|---------------------------------------------------------------------------------------|-------------------|-----------------------|-----------|--------|-----------|-------|
|       |                                                                                       |                   | Lenze                 | Selection |        |           |       |
| C1857 | /1: CAN-IN1<br>/2: CAN-IN2<br>/3: CAN-IN3                                             | 22718 =<br>0x58BE | /1 ... /5:<br>3000 ms | 0         | [1 ms] | 65535     | FIX32 |
| C2357 | /4: BUS-OFF monitoring time<br>/5: AIF monitoring time<br>/6: Sync Rx monitoring time | 22218 =<br>0x56CA | /6: 0 ms              |           |        |           |       |

**Note!**

- ▶ The monitoring time starts with the arrival of the first telegram.
- ▶ C1857/x = 0: Monitoring deactivated.
- ▶ C2357/x = 0: Monitoring deactivated.

**CAN-IN1 ... 3 (C1857/1 ... 3, C2357/1 ... 3)**

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 C1882/C2382.

**BUS-OFF (C1857/4, C2357/4)**

Even if the node changes to the state BUS-OFF, it is possible to set a time (in ms) in which a reaction can take place. A monitoring response can be set under code C1882/C2382.

**AIF monitoring time (C1857/5, C2357/5)**

Time during boot-up for detecting a basic device. If no basic device is detected, an emergency telegram (📖 87) will be sent on the CAN bus after this time.

**Sync monitoring time (C1857/6, C2357/6)**

Monitoring time for a received, cyclic sync telegram.

**Relation to CANopen**

- ▶ The object I-1006 (Communication Cycle Period) has a direct influence on the subcodes C1857/6 or C2357/6. A new value in the code deletes bit 30 of the object I-1005 (COB-ID SYNC message).
- ▶ The objects I-1400/5 ... I-1402/5 (TPDO Event Timer) have a direct influence on the subcodes C1857/1 ... 3 or C2357/1 ... 3. Newly entered values in the objects I-1400/5 ... I-1402/5 are also transferred to the subcodes C1857/1 ... 3 or C2357/1 ... 3.

**C1859/C2359:****Display of DIP switch position**

| Code  | Subcode | Index                          | Possible settings |           | Data type |
|-------|---------|--------------------------------|-------------------|-----------|-----------|
|       |         |                                | Lenze             | Selection |           |
| C1859 | -       | 22716 =<br>0x58BC              | Disp              |           | U16       |
| C2359 | -       | 22216 <sub>d</sub> =<br>0x56C8 |                   |           | U32       |

The DIP switch position is indicated as a hexadecimal number with the initialisation of the communication module.

**Note!**

In case of 9300 Servo PLC, Drive PLC and ECS, the value will be transferred to the standard device when the communication module is initialised.

**DIP switch valencies for hexadecimal calculation**

| Switch | -         | 1   | 2   | 3   | 4  | 5  | 6  | 7 | 8 | 9 | 10 |
|--------|-----------|-----|-----|-----|----|----|----|---|---|---|----|
| Value  | -         | 512 | 256 | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1  |
| Bit    | 15 ... 10 | 9   | 8   | 7   | 6  | 5  | 4  | 3 | 2 | 1 | 0  |

**Example for DIP switch position**

- ▶ Switches 3, 5, 6, 7 = ON (address 23)
  - ▶ Switches 8, 9, 10 = OFF (baud rate 500 kbit/s)
- The sum of the corresponding valencies "0xB8" (184) is displayed when code C1859/C2359 is read.

### C1860: Display of the current DIP switch position

| Code  | Subcode | Index          | Possible settings             |           | Data type |
|-------|---------|----------------|-------------------------------|-----------|-----------|
|       |         |                | Lenze                         | Selection |           |
| C1860 | -       | 22715 = 0x58BB | <input type="checkbox"/> Disp |           | U16       |

By displaying the current DIP switch position it is possible to find out if the switch position for node address and baud rate has changed since the last initialisation.

#### DIP switch valencies

| Switch | -         | 1   | 2   | 3   | 4  | 5  | 6  | 7 | 8 | 9 | 10 |
|--------|-----------|-----|-----|-----|----|----|----|---|---|---|----|
| Value  | -         | 512 | 256 | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1  |
| Bit    | 15 ... 10 | 9   | 8   | 7   | 6  | 5  | 4  | 3 | 2 | 1 | 0  |

#### Example for DIP switch position

- Switches 3, 5, 6, 7 = ON (address 23)
  - Switches 8, 9, 10 = OFF (baud rate 500 kbit/s)
- The sum of the corresponding valencies "0xB8" (184) is displayed when code C1859/C2359 is read.

### C1861: Display of the active CAN node address

| Code  | Subcode | Index          | Possible settings             |           | Data type |
|-------|---------|----------------|-------------------------------|-----------|-----------|
|       |         |                | Lenze                         | Selection |           |
| C1861 | -       | 22714 = 0x58BA | <input type="checkbox"/> Disp |           | FIX32     |

Display of the CAN node address active on the CAN bus.

### C1862: Display of the active baud rate

| Code  | Subcode | Index          | Possible settings             |                                                                                                       | Data type |
|-------|---------|----------------|-------------------------------|-------------------------------------------------------------------------------------------------------|-----------|
|       |         |                | Lenze                         | Selection                                                                                             |           |
| C1862 | -       | 22713 = 0x58B9 | <input type="checkbox"/> Disp | 0: 500 kbps<br>1: 250 kbps<br>2: 125 kbps<br>3: 50 kbps<br>4: 1000 kbps<br>5: 20 kbps<br>6: 10 kbit/s | FIX32     |

Display of the baud rate active on the CAN bus.

**C1864/C2364:****PDP transmission with status change to "Operational"**

| Code  | Subcode | Index          | Possible settings |                     | Data type |
|-------|---------|----------------|-------------------|---------------------|-----------|
|       |         |                | Lenze             | Selection           |           |
| C1864 | -       | 22711 = 0x58B7 | 0                 | 0: Do not send PDOs | FIX32     |
| C2364 | -       | 22211 = 0x56C3 |                   | 1: Send PDOs        |           |

When the CAN status changes from "Pre-Operational" to "Operational", all PDOs are sent once.

**C1865/C2365:****Validity of SDO2 and PDOs**

| Code  | Subcode                                                             | Index          | Possible settings                |           | Data type |
|-------|---------------------------------------------------------------------|----------------|----------------------------------|-----------|-----------|
|       |                                                                     |                | Lenze                            | Selection |           |
| C1865 | /1: SDO2<br>/2: CAN-IN/OUT1<br>/3*: CAN-IN/OUT2<br>/4*: CAN-IN/OUT3 | 22710 = 0x58B6 | /1: 0<br>/2: 1<br>/3: 1<br>/4: 1 | 0 [1]     | 3 FIX32   |
| C2365 | /1: SDO2<br>/2: CAN-IN/OUT1<br>/3*: CAN-IN/OUT2<br>/4*: CAN-IN/OUT3 | 22210 = 0x56C2 | /1: 0<br>/2: 1<br>/3: 1<br>/4: 1 |           |           |

\*) not effective when using 8200 vector or 93XX controllers

**Validity of service data object 2 (SDO2) and of process data objects**

| Possible selections C1865/C2365 |                           |
|---------------------------------|---------------------------|
| Value                           | Description               |
| 0                               | SDO/PDO invalid           |
| 1                               | SDO/PDO valid             |
| 2                               | Receiving direction valid |
| 3                               | Sending direction valid   |

**C1867/C2367:  
Sync Rx identifier**

| Code  | Subcode | Index             | Possible settings |           |          | Data type |
|-------|---------|-------------------|-------------------|-----------|----------|-----------|
|       |         |                   | Lenze             | Selection |          |           |
| C1867 | -       | 22708 =<br>0x58B4 | 128               | 0         | [1] 2047 | FIX32     |
| C2367 | -       | 22208 =<br>0x56C0 |                   |           |          |           |

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

**Relation to CANopen**

The object I-1005 (COB-ID SYNC message) has a direct influence on this code. Newly written values in the object are also transferred to code C1867/C2367.

**C1868/C2368:  
Sync Tx identifier**

| Code  | Subcode | Index             | Possible settings |           |          | Data type |
|-------|---------|-------------------|-------------------|-----------|----------|-----------|
|       |         |                   | Lenze             | Selection |          |           |
| C1868 | -       | 22707 =<br>0x58B3 | 128               | 0         | [1] 2047 | FIX32     |
| C2368 | -       | 22207 =<br>0x56BF |                   |           |          |           |

This code contains the identifier with which the sync telegram is sent.

**Relation to CANopen**

The object I-1005 (COB-ID SYNC message) has a direct influence on this code. Newly written values in the object are also transferred to code C1868/C2368.

### C1869/C2369: Consumer heartbeat ID

| Code  | Subcode | Index             | Possible settings |           |     | Data type |
|-------|---------|-------------------|-------------------|-----------|-----|-----------|
|       |         |                   | Lenze             | Selection |     |           |
| C1869 | -       | 22706 =<br>0x58B2 | 0                 | 0 [1]     | 255 | FIX32     |
| C2369 | -       | 22206 =<br>0x56BE |                   |           |     |           |

This code contains the identifier with which the heartbeat telegram for monitoring is received.

#### Relation to CANopen

The object I-1016, subindex 1 (consumer heartbeat) has a direct influence on this code. A value entered into subindex 1 of I-1016 is also transferred to code C1869/C2369.

### C1870/C2370: Heartbeat time (consumer, producer)

| Code  | Subcode      | Index             | Possible settings |           |       | Data type |
|-------|--------------|-------------------|-------------------|-----------|-------|-----------|
|       |              |                   | Lenze             | Selection |       |           |
| C1870 | /1: Consumer | 22705 =<br>0x58B1 | 0                 | 0 [1 ms]  | 65535 | FIX32     |
|       | /2: Producer |                   | 0                 |           |       |           |
| C2370 | /1: Consumer | 22205 =<br>0x56BD | 0                 |           |       |           |
|       | /2: Producer |                   | 0                 |           |       |           |

#### Consumer heartbeat time

Time intervals at which the heartbeat telegram to be monitored is awaited on the CAN bus.

#### Producer heartbeat time

Time intervals at which the module sends a heartbeat telegram including its status on the CAN bus.

#### Relation to CANopen

- The object I-1016, subindex 1 (consumer heartbeat time) has a direct influence on this code. A value written into subindex 1 of I-1016 is also transferred to code C1870/1 C23701 .
- The object I-1017 (producer heartbeat time) has a direct influence on this code. A value written into the object is also transferred to code C1870/2 / C2370/2.

**C1871/C2371:  
Emergency ID**

| Code  | Subcode | Index             | Possible settings |           |      | Data type |
|-------|---------|-------------------|-------------------|-----------|------|-----------|
|       |         |                   | Lenze             | Selection |      |           |
| C1871 | -       | 22704 =<br>0x58B0 | 128               | 0 [1]     | 2047 | FIX32     |
| C2371 | -       | 22204 =<br>0x56BC |                   |           |      |           |

This code contains the identifier with which the emergency telegram is sent.

**Relation to CANopen**

The object I-1014 (COB-ID Emergency Object) has a direct influence on this code. A value written into the object is also transferred to code C1871/C2371.

**C1872/C2372:  
Emergency inhibit time**

| Code  | Subcode | Index             | Possible settings |           |       | Data type |
|-------|---------|-------------------|-------------------|-----------|-------|-----------|
|       |         |                   | Lenze             | Selection |       |           |
| C1872 | -       | 22703 =<br>0x58AF | 0                 | 0 [1 ms]  | 65535 | FIX32     |
| C2372 | -       | 22203 =<br>0x56BB |                   |           |       |           |

This code contains the delay time for sending an emergency telegram.

**Relation to CANopen**

The object I-1015 (Inhibit Time EMCY) has a direct influence on this code. A value written into the object is also transferred to code C1872/C2372.

**C1873/C2373:**  
**Sync rate CAN-IN1 ... 3**

| Code  | Subcode      | Index          | Possible settings |           | Data type |
|-------|--------------|----------------|-------------------|-----------|-----------|
|       |              |                | Lenze             | Selection |           |
| C1873 | /1: CAN-IN1  | 22702 = 0x58AE | 1                 | 0 [1]     | 240       |
| C2373 | /2*: CAN-IN2 | 22202 = 0x56BA |                   |           |           |
|       | /3*: CAN-IN3 |                |                   |           |           |

\*) not effective when using 8200 vector or 93XXcontrollers

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

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 code C1867/C2367 of the communication module.

**Example:**

- Selection n = 23:

Acceptance of input PDO (to CAN-IN1 ... 3) into the controller after arrival of the 23rd sync telegram.

**Relation to CANopen**

The objects I-1400 ... I-1402 (Receive PDO Communication Parameters), subindex 2 each (transmission type), are mapped on the corresponding subcodes of C1873/C2373 in the value range 1 ... 240.

Exception: Object I-140X, subindex 2 = "254" is mapped manufacturer-specifically as value "0" on the corresponding subcodes of C1873/C2373.

| Object I-140X, subindex        | Mapping to code / subcode    |
|--------------------------------|------------------------------|
| I-140X, subindex 2 = 1 ... 240 | C1873, subcode X = 1 ... 240 |
| I-140X, subindex 2 = 254       | C1873, subcode X = 0         |

### C1874/C2374: Sync rate CAN-OUT1...3

| Code  | Subcode                       | Index             | Possible settings |           |     | Data type |
|-------|-------------------------------|-------------------|-------------------|-----------|-----|-----------|
|       |                               |                   | Lenze             | Selection |     |           |
| C1874 | /1: CAN-OUT1<br>/2*: CAN-OUT2 | 22701 =<br>0x58AD | 1                 | 1 [1]     | 240 | FIX32     |
| C2374 | /3*: CAN-OUT3                 | 22201 =<br>0x56B9 |                   |           |     |           |

\*) not effective when using 8200 vector or 93XXcontrollers

The process output 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.

#### Relation to CANopen

The objects I-1800 ... I-1802 (Transmit PDO Communication Parameters), subindex 2 each (transmission type), are transferred to the corresponding subcodes of C1874/C2374 in the value range 1 ... 240.

**C1875/C2375:  
Tx mode CAN-OUT1...3**

| Code  | Subcode                       | Index             | Possible settings |           | Data type |
|-------|-------------------------------|-------------------|-------------------|-----------|-----------|
|       |                               |                   | Lenze             | Selection |           |
| C1875 | /1: CAN-OUT1<br>/2*: CAN-OUT2 | 22700 =<br>0x58AC | /1: 0<br>/2: 1    | 0 [1]     | 3 FIX32   |
| C2375 | /3*: CAN-OUT3                 | 22200 =<br>0x56B8 | /3: 1             |           |           |

\*) not effective when using 8200 vector or 93XXcontrollers

This code contains a selection which indicates at which event the TPDOs CAN-OUT1 ... 3 are to be sent. It is possible to make an individual selection for each TPDO by dividing the code in subcodes.

- ▶ Value = 0:
  - The TPDOs are sent when a sync telegram has been received.
  - Under code C1874/C2374 you can set that the TPDOs are only to be sent after the n-th sync telegram (n = 1 ... 240).
- ▶ Value = 1:
  - TPDOs are not sent.
  - With this selection the transmission of TPDOs is deactivated.  
This is useful for 8200 vector and 9300 drives which are only able to exchange four words as a maximum of control and status information via the AIF. For this purpose, the use of one TPDO is sufficient, TPDOs 2 and 3 are deactivated (Lenze setting). This avoids an unnecessary load of the CAN bus.
- ▶ Value = 2:
  - The corresponding TPDO is either transmitted event-controlled or cyclically.
  - The TPDO is transmitted cyclically if a cycle time is defined for the TPDO under code C1856/C2356. If the cycle time is zero, the TPDO is sent in case of an event, i.e. bit change within the TPDO.
- ▶ Value = 3:
  - The TPDO is event-controlled and cyclically transmitted. This means that the object is transmitted with the cycle time defined under code C1856/C2356.
  - In addition, the object will be transmitted if one or several bits are changed within the TPDO.


**Note!**

If the transmission is event-controlled (also with cyclic superimpositions) some bits can be hidden by masking the object using codes C1876 ... C1878 or with the 9300 Servo PLC, Drive PLC and ECS C2376 ... C2378. This means that the TPDO will not be sent when a bit is being changed.

- ▶ C1875/C2375, subcodes 1 ... 3 = 0  
The value of code C1874/C2374, subcode 1, 2 or 3 is displayed in the objects I-1800 ... I-1802 (subindex 2 each).
- ▶ C1875/C2375, subcodes 1 ... 3 = 1  
The value "252" is displayed in the objects I-1800 ... I-1802hex.
- ▶ C1875/C2375, subcodes 1 ... 3 = 2 or 3  
The value "254" is displayed in the objects I-1800 ... I-1802hex.

When setting the objects I-1800 ... I-1802, the same relation prevails:

| Code C1875/C2375, subcode 1, 2 or 3 | Object I-180X, subindex 2                                  |
|-------------------------------------|------------------------------------------------------------|
| 0                                   | Code C1874/C2374,<br>subcode 1, 2 or 3 (value = 1 ... 240) |
| 1                                   | 252                                                        |
| 2 or 3                              | 254                                                        |

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

This mask is used to skip one or several bits of the output PDO CAN-OUT1.

| Code  | Subcode                            | Index             | Possible settings |           |       | Data type |
|-------|------------------------------------|-------------------|-------------------|-----------|-------|-----------|
|       |                                    |                   | Lenze             | Selection |       |           |
| C1876 | /1: CAN-OUT1.W1<br>/2: CAN-OUT1.W2 | 22699 =<br>0x58AB | 65535             | 0 [1]     | 65535 | FIX32     |
| C2376 | /3: CAN-OUT1.W3<br>/4: CAN-OUT1.W4 | 22199 =<br>0x56B7 |                   |           |       |           |

The event-controlled transmission of the CAN-OUT object can be e.g. dependent on one bit only. (See also code C1875/C2375,  140.)

**Example:**

The mask in word 3 of the process data object CAN-OUT1 is set by code C1876/3 with the value "0x20".

**► 1st cycle**

Result after 1st cycle: The PDO is sent.

|               | CAN-OUT1.W3 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---------------|-------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| <b>MASK</b>   | 0           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| <b>Data</b>   | 1           | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| <b>Result</b> | 0           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

**► 2nd cycle**

Result after 2nd cycle:

- New data have been written into CAN-OUT1.
- The PDO is sent due to the bit change.

|               | CAN-OUT1.W3 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---------------|-------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| <b>MASK</b>   | 0           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| <b>Data</b>   | 1           | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| <b>Result</b> | 0           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

### C1877/C2377: Masks CAN-OUT2

This mask is used to skip one or several bits of the output PDO CAN-OUT2.

| Code  | Subcode                            | Index             | Possible settings |           |     | Data type |       |
|-------|------------------------------------|-------------------|-------------------|-----------|-----|-----------|-------|
|       |                                    |                   | Lenze             | Selection |     |           |       |
| C1877 | /1: CAN-OUT2.W1<br>/2: CAN-OUT2.W2 | 22698 =<br>0x58AA | 65535             | 0         | [1] | 65535     | FIX32 |
| C2377 | /3: CAN-OUT2.W3<br>/4: CAN-OUT2.W4 | 22198 =<br>0x56B6 |                   |           |     |           |       |

For more information, please see the description of code C1876/C2376 (142).

### C1878/C2378: Masks CAN-OUT3

This mask is used to skip one or several bits of the output PDO CAN-OUT3.

| Code  | Subcode                            | Index             | Possible settings |           |     | Data type |       |
|-------|------------------------------------|-------------------|-------------------|-----------|-----|-----------|-------|
|       |                                    |                   | Lenze             | Selection |     |           |       |
| C1878 | /1: CAN-OUT3.W1<br>/2: CAN-OUT3.W2 | 22697 =<br>0x58A9 | 65535             | 0         | [1] | 65535     | FIX32 |
| C2378 | /3: CAN-OUT3.W3<br>/4: CAN-OUT3.W4 | 22197 =<br>0x56B5 |                   |           |     |           |       |

For more information, please see the description of code C1876/C2376 (142).

### C1882/C2382: Monitoring response

| Code  | Subcode                                                                                        | Index             | Possible settings            |                                                                       |     | Data type |       |
|-------|------------------------------------------------------------------------------------------------|-------------------|------------------------------|-----------------------------------------------------------------------|-----|-----------|-------|
|       |                                                                                                |                   | Lenze                        | Selection                                                             |     |           |       |
| C1882 | /1: response CAN-IN1<br>/2: response CAN-IN2<br>/3: response CAN-IN3<br>/4: response BUS-OFF   | 22693 =<br>0x58A5 | /1:0<br>/2:0<br>/3:0<br>/4:0 | 0                                                                     | [1] | 2         | FIX32 |
| C2382 | /5: response<br>life guarding event /<br>heartbeat event<br>/6: response Rx sync<br>monitoring | 22193 =<br>0x56B1 | /5:0<br>/6: 0                | 0: no response<br>1: Controller inhibit (CINH)<br>2: Quick stop (QSP) |     |           |       |

Setting the response in the controller if the monitoring times have been exceeded (see code C1857/C2357, 131.).

### 14.3 Important controller codes

This section briefly describes the most important codes of the standard device for CAN communication.



#### Documentation for the basic drive

Here you can find detailed information on the codes of the standard device.

#### C0001:

##### Operating mode for 8200 vector

| Code  | Subcode | Index             | Possible settings |                                   | Data type |
|-------|---------|-------------------|-------------------|-----------------------------------|-----------|
|       |         |                   | Lenze             | Selection                         |           |
| C0001 | -       | 24574 =<br>0x5FFE | 0                 | see System Manual for 8200 vector | FIX32     |

The operating mode defines which source may currently write to which parameters.

The operating unit and CAN are always authorised for parameter setting.



#### Note!

Please observe that the operating mode C0001 is contained in all parameter sets. Therefore C0001 must be set equally in all parameter sets.

#### C0009:

##### CAN node address

| Code  | Subcode | Index             | Possible settings |           | Data type |
|-------|---------|-------------------|-------------------|-----------|-----------|
|       |         |                   | Lenze             | Selection |           |
| C0009 | -       | 24566 =<br>0x5FF6 | 1                 | 1 [1]     | 99 FIX32  |

Node address for unambiguous addressing of the device in a CAN network.

#### C0046:

##### Display of frequency setpoint

| Code  | Subcode | Index             | Possible settings             |                                          | Data type |
|-------|---------|-------------------|-------------------------------|------------------------------------------|-----------|
|       |         |                   | Lenze                         | Selection                                |           |
| C0046 | -       | 24529 =<br>0x5FD1 | <input type="checkbox"/> Disp | see documentation of the standard device | FIX32     |

**C0125:  
Baud rate**

| Code  | Subcode | Index             | Possible settings |                                             | Data type |
|-------|---------|-------------------|-------------------|---------------------------------------------|-----------|
|       |         |                   | Lenze             | Selection                                   |           |
| C0125 | -       | 24450 =<br>0x5F82 | 0                 | see documentation of the<br>standard device | FIX32     |

**C0126:  
Behaviour with communication error (extract)**

| Code  | Subcode | Index             | Possible settings |                                             | Data type |
|-------|---------|-------------------|-------------------|---------------------------------------------|-----------|
|       |         |                   | Lenze             | Selection                                   |           |
| C0126 | -       | 24449 =<br>0x5F81 |                   | see documentation of the<br>standard device | FIX32     |

Monitoring of the internal communication between communication module and standard device (controller).

**C0135:  
Controller control word**

| Code  | Subcode | Index             | Possible settings |                                           | Data type |
|-------|---------|-------------------|-------------------|-------------------------------------------|-----------|
|       |         |                   | Lenze             | Selection                                 |           |
| C0135 | -       | 24440 =<br>0x5F78 |                   | see documentation for the<br>basic device | I16       |

**C0150:  
Controller status word**

| Code  | Subcode | Index             | Possible settings |                                             | Data type |
|-------|---------|-------------------|-------------------|---------------------------------------------|-----------|
|       |         |                   | Lenze             | Selection                                   |           |
| C0150 | -       | 24425 =<br>0x5F69 |                   | see documentation of the<br>standard device | I16       |

**C0161 ... C0164:**  
**Fault memory 8200 vector**

| Code  | Subcode | Index          | Possible settings             |                    | Data type |
|-------|---------|----------------|-------------------------------|--------------------|-----------|
|       |         |                | Lenze                         | Selection          |           |
| C0161 | -       | 24414 = 0x5F5E | <input type="checkbox"/> Disp | active fault       | FIX32     |
| C0162 |         | 24413 = 0x5F5D |                               | last fault         |           |
| C0163 |         | 24412 = 0x5F5C |                               | last but one fault |           |
| C0164 |         | 24411 = 0x5F5B |                               | last but two fault |           |


**System Manual for 8200 vector**

Here you can find detailed information on the fault messages.

**C0168:**  
**Fault memory 9300 / ECSXX**

| Code  | Subcode | Index          | Possible settings             |                                          | Data type |
|-------|---------|----------------|-------------------------------|------------------------------------------|-----------|
|       |         |                | Lenze                         | Selection                                |           |
| C0168 | 0 ... 8 | 24407 = 0x5F57 | <input type="checkbox"/> Disp | see documentation of the standard device | FIX32     |


**9300 system manual / ECSXX operating instructions**

Here you can find detailed information on the fault messages.

**C2120:  
AIF control byte**

| Code  | Subcode | Index          | Possible settings |                 | Data type |
|-------|---------|----------------|-------------------|-----------------|-----------|
|       |         |                | Lenze             | Selection       |           |
| C2120 | -       | 22455 = 0x57B7 | 0                 | See table below | FIX32     |

| Possible settings       |                                                         |
|-------------------------|---------------------------------------------------------|
| Selection               | Description                                             |
| 0                       | No command                                              |
| 1                       | Update codes 23XX and CAN reinitialisation ≡ reset node |
| 2                       | Update codes C23XX                                      |
| 10                      | Reread C2356/1...4                                      |
| 11                      | Reread C2357                                            |
| 12                      | Reread C2375                                            |
| 13                      | Reread C2376 ... C2378                                  |
| 14                      | Reread C2382                                            |
| 16 <sup>1)</sup>        | Update C7999 and C7998 including CAN reinitialisation   |
| 17 <sup>1)</sup>        | Update C7999, mapping table of profile indices          |
| 18 <sup>1)</sup>        | Update C7998, error list of emergency messages          |
| 32 <sup>1)</sup>        | Reset error source, activate emergency messages         |
| 33 ... 62 <sup>1)</sup> | Error source 1 ... 30, activate emergency messages      |

<sup>1)</sup> Value range is only valid for profile applications in a PLC

With the AIF control byte it is possible to read the codes C23XX saved in the 9300 Servo PLC, Drive PLC and ECS into the communication module.

**Execute command**

How to accept the values in the communication module:

- Write the table value of the required command into the AIF control byte.
- Change the status of the MSB of the AIF control byte. The command is executed.

**Example**

By writing the value "2" into the AIF control byte all codes C23XX are re-read when the MSB changes its status.

For some codes it is necessary to carry out a CAN reinitialisation so that new values and the functions derived from them will become effective.

**C2121:****AIF status byte**

| Code  | Subcode | Index             | Possible settings |           |     | Data type |
|-------|---------|-------------------|-------------------|-----------|-----|-----------|
|       |         |                   | Lenze             | Selection |     |           |
| C2121 | -       | 22454 =<br>0x57B6 | 0                 | 0 [1]     | 255 | U8        |

The AIF status byte provides the 9300 Servo PLC, Drive PLC and ECS with information of the communication module. With this information, the 9300 Servo PLC, Drive PLC and ECS can monitor the status of the communication module. Depending on the communication module states, it is possible for you to take corresponding countermeasures.

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

## 15 Index

### Zahlen

#### 8200 vector

- control word, 53
- status word, 55

#### 8200 vector fault memory, 146

#### 8200 vector operating mode (C0001), 144

#### 8200 vector special features, 81

#### 9300

- control word, 57
- status word, 60

#### 9300 fault memory, 146

#### 9300 Servo PLC special features, 81

### A

#### Address setting, 34

#### Addressing CAN-IN/OUT, 126

- individual, 129

#### AIF control byte (C2120), 81, 147

#### AIF status byte (C2121), 82, 148

#### AIF-IN, function block, 58

#### AIF-IN all double words, 121

#### AIF-IN all words, 121

#### AIF-OUT, function block, 60

#### AIF-OUT all double words, 121

#### AIF-OUT all words, 121

#### Application as directed, 13

### B

#### Basic identifier, 43

#### Basic insulation, 24

#### Baud rate, 17, 124

- setting, 35
- system bus (CAN). *Siehe* baud rate

#### Baud rate (C0125), 145

#### Behaviour with communication error (C0126), 145

#### Boot-up time, 130

#### Bus cable length, 27

### C

#### C0001: Operating mode for 8200 vector, 144

#### C0009: CAN node address, 144

#### C0046: Display of frequency setpoint, 144

#### C0125: Baud rate, 145

#### C0126: Behaviour with communication error, 145

#### C0135: Controller control word, 145

#### C0150: Controller status word, 145

#### C0161 ... C0164: Fault memory 8200 vector, 146

#### C0168: Fault memory 9300 / ECSXX, 146

#### C1810: Software ID, 120

#### C1811: Software creation date, 120

#### C1812: Software ID, 120

#### C1813: Software creation date, 120

#### C1822: AIF-IN all words, 121

#### C1823: AIF-OUT all words, 121

#### C1824: AIF-IN all double words, 121

#### C1825: AIF-OUT all double words, 121

#### C1827/C2327: Guard Time, 122

#### C1828/C2328: Life time factor, 122

#### C1850/C2350: CAN node address, 123

#### C1851/C2351: Baud rate, 124

#### C1852/C2352: Master/slave operation, 125

#### C1853/C2353: Addressing CAN-IN/OUT, 126

#### C1854/C2354: Individual addressing CAN-IN/OUT, 129

#### C1855/C2355: Display of resulting identifiers CAN-IN/OUT, 129

#### C1856/C2356: Boot-Up and cycle times, 130

#### C1857/C2357: Monitoring time, 131

#### C1859/C2359: Display DIP switch position, 132

#### C1860: Display of the current DIP switch position, 133

#### C1861: Display of the active CAN node address, 133

#### C1862: Display of the active baud rate, 133

#### C1864/C2364: PDO transmission with status change to "Operational", 134

#### C1865/C2365: Validity of SDO2 and PDOs, 134

#### C1867/C2367: Sync Rx identifier, 135

#### C1868/C2368: Sync Tx identifier, 135

#### C1869/C2369: Consumer heartbeat ID, 136

#### C1870/C2370: Heartbeat time (consumer, producer), 136

**C1871/C2371:** Emergency ID, 137  
**C1872/C2372:** Emergency inhibit time, 137  
**C1873/C2373:** Sync rate CAN-IN1 ... 3, 138  
**C1874/C2374:** Sync rate CAN-OUT1...3, 139  
**C1875/C2375:** Tx mode CAN-OUT1...3, 140  
**C1876/C2376:** Masks CAN-OUT1, 142  
**C1877/C2377:** Masks CAN-OUT2, 143  
**C1878/C2378:** Masks CAN-OUT3, 143  
**C1882/C2382:** Monitoring response, 143  
**C2120:** AIF control byte, 81 , 147  
**C2121:** AIF status byte, 82 , 148  
**Cable specification,** 26  
**CAN communication phases,** 45  
**CAN data telegram,** 42  
**CAN network, statuses,** 45  
**CAN network management (NMT),** 45 , 46  
**CAN node address,** 123  
**CAN node address (C0009),** 144  
**CAN state transitions,** 47  
**CANopen indices for mapping,** 67  
**CANopen objects,** 91  
**CE-typical drive system,** 23  
**Checking the use of repeaters,** 29  
**COB-ID,** 43  
**COB-ID emergency object (I-1014),** 104  
**COB-ID SYNC message (I-1005),** 98  
**Code numbers / index, conversion,** 70  
**Codes,** 115  
**Command,** 73  
**Commissioning,** 32  
- enable drive, 37  
- Initial switch-on, 36  
**Communication cycle period (I-1006),** 99  
**Communication media,** 17  
**Communication Object Identifier,** 43

**Communication phases,** 45  
**Communication profile,** 17  
**Communication time,** 19  
**Communication-relevant Lenze codes,** 119  
**Connections,** 16  
**Consumer heartbeat ID,** 136  
**Consumer heartbeat time (I-1016),** 105  
**Control word**  
- 8200 vector, 53  
- 9300, 57  
**Control word of controller,** 145  
**Controller, Process data signals,** 52  
**Controller codes,** 144  
**Controller control word,** 145  
**Controller status word,** 145  
**Cycle times,** 130  
**Cyclic process data objects (PDO),** 51

## D

**Data telegram,** 42  
**Data transfer,** 42  
**Definition of notes used,** 10  
**Definitions,** 9  
**Device protection,** 12 , 21  
**Device type (I-1000),** 96  
**Diagnostics,** 88  
**DIP switch position (current), display,** 133  
**DIP switch position (during initialisation), display,** 132  
**Display**  
- DIP switch position (current), 133  
- DIP switch position (during initialisation), 132  
**Display of frequency setpoint (C0046),** 144  
**Display of resulting identifiers CAN-IN/OUT,** 129  
**Display of the active baud rate,** 133  
**Display of the active CAN node address,** 133  
**Drive PLC special features,** 81

**E**

ECS fault memory, 146  
 ECS special features, 81  
 Electrical installation, 23  
 Electrical isolation, 24  
 Emergency error codes, 87  
 Emergency ID, 137  
 Emergency inhibit time, 137  
 Emergency inhibit time (I-1015), 105  
 Emergency telegram, 87  
 Enable drive, 37  
 Error behaviour (I-1029), 106  
 Error codes, 76  
   - Emergency, 87  
 Error history (I-1003), 97  
 Error messages, 75  
 Error register (I-1001), 96  
 Examples  
   - Read block parameters, 79  
   - reading parameters, 77  
   - writing parameters, 78  
 External voltage supply, 30

**F**

Fault memory 8200 vector, 146  
 Fault memory 9300 / ECSXX, 146  
 Frequency setpoint display (C0046), 144

**G**

Guard Time, 122  
 Guard time (I-100C), 100

**H**

Hardware version, type code, 14  
 Heartbeat Protocol, 83  
 Heartbeat time (consumer, producer), 136

**I**

I-1000: Device type, 96  
 I-1001: Error register, 96  
 I-1003: Error history, 97  
 I-1005: COB-ID SYNC message, 98  
 I-1006: Communication cycle period, 99

I-1008: Manufacturer's device name, 99  
 I-100A: Manufacturer software version, 100  
 I-100C: Guard time, 100  
 I-100D: Life time factor, 101  
 I-1010: Store parameters, 101  
 I-1011: Restore default parameters, 102  
 I-1014: COB-ID emergency object, 104  
 I-1015: Emergency inhibit time, 105  
 I-1016: Consumer heartbeat time, 105  
 I-1017: Producer heartbeat time, 106  
 I-1018: Module device description, 106  
 I-1029: Error behaviour, 106  
 I-1200: Server SDO parameters, 107  
 I-1201: Server SDO parameters, 107  
 I-1400: Receive PDO communication parameters, 109  
 I-1401: Receive PDO communication parameters, 109  
 I-1402: Receive PDO communication parameters, 109  
 I-1600: Receive PDO mapping parameters, 111  
 I-1601: Receive PDO mapping parameters, 111  
 I-1602: Receive PDO mapping parameters, 111  
 I-1800: Transmit PDO communication parameters, 112  
 I-1801: Transmit PDO communication parameters, 112  
 I-1802: Transmit PDO communication parameters, 112  
 I-1A00: Transmit PDO mapping parameters, 114  
 I-1A01: Transmit PDO mapping parameters, 114  
 I-1A02: Transmit PDO mapping parameters, 114  
 Identification, 14  
 Identifier (COB-ID), 43  
 Identifiers, Display of the resulting identifier, 129  
 Indexing of Lenze codes, 70  
 Individual addressing CAN-IN/OUT, 129  
 Initial switch-on, 36  
 Installation, 21  
   - electrical, 23  
   - mechanical, 22  
   - system bus (CAN), 25  
 Installing EDS files, 32  
 Interfaces, 16  
 Internal voltage supply, 30

## L

LED status displays, 89  
LEDs, 36 , 89  
Lenze codes, 115  
Life time factor, 122  
Life time factor (I-100D), 101

## M

Manufacturer software version (I-100A), 100  
Manufacturer's device name (I-1008), 99  
Mapping in CANopen objects, 65  
- AIF data image in codes, 66  
- AIF interface assignment, 68  
- AIF modes, 68  
- CANopen indices for mapping, 67  
- Device-internal mechanisms, 67  
Masks CAN-OUT1, 142  
Masks CAN-OUT2, 143  
Masks CAN-OUT3, 143  
Master/slave operation, 125  
Mechanical installation, 22  
Module device description (I-1018), 106  
Monitoring, 83  
Monitoring response, 143  
Monitoring time, 131

## N

Nameplate, 14  
Network management (NMT), 45 , 46  
Network topology, 17  
Node address, 43 , 123  
Node address setting, 34  
Node addresses, 17  
Node Guarding Protocol, 85  
Node ID, 43  
Notes, definition, 10

## O

Operating mode for 8200 vector (C0001), 144  
Order designation, 17

## P

Parameter, C0142 (restart protection), 38  
Parameter channel, 25  
Parameter data telegram, 73  
Parameter data transfer, 69  
parameter for 8200 vector, frequency setpoint (C0046), 50  
Parameter sets  
- 8200 vector, 71  
- 93XX, 72  
- Lenze, 71  
PDO transmission with "Operational", 134  
Plug connector with double screw connection, 24  
Process data channel, 25  
- configuring, 50  
Process Data Objects (PDO), 44  
Process data objects (PDO)  
- available, 48  
- cyclic, 51  
Process data signals, 52  
- 8200 vector, 52  
- 9300 Servo PLC, 61  
- 93XX, 56  
- Drive PLC, 61  
- ECSxA, 61  
Process data telegram  
- RPDO, 49  
- TPDO, 49  
Process data transfer, 48  
Processing time, 19  
- in the controller, 19  
Producer heartbeat time (I-1017), 106  
Product description, 13  
- application as directed, 13  
Product features, 15  
Protection against restart, 38  
Protection of persons, 12  
Protective insulation, 18

**R**

Receive PDO communication parameters (I-1400 ... I-1402), 109

Receive PDO mapping parameters (I-1600 ... I-1602), 111

Replacing EMF2172IB (CAN), 39

Replacing the EMF2172IB communication module (CAN), 39

Residual hazards, 12

Restart protection, 38

Restore default parameters (I-1011), 102

**S**

Safety instructions, 11

- application as directed, 13
- definition, 10
- device- and application-specific, 12
- layout, 10

Segment cable length, 28

Server SDO parameters (I-1200/I-1201), 107

Service Data Objects (SDO), 44

Setpoint source, selection, 50

Setting baud rate, 33

Setting node address, 33

Software creation date, 120

Software ID, 120

Software version, type code, 14

Specification of the transmission cable, 26

State transitions, 47

Status displays, 89

Status word, 9300, 60

status word, 8200 vector, 55

Status word of controller, 145

Statuses, CAN network, 45

Store parameters (I-1010), 101

Structure of the parameter data telegram, 73

Switch on, initial, 36

Sync rate CAN-IN1...3, 138

Sync rate CAN-OUT1...3, 139

Sync Rx identifier, 135

Sync telegram, 51

Sync Tx identifier, 135

Synchronisation of cyclic process data, 51

System bus (CAN), baud rate, 27

system bus (CAN), wiring, 25

**T**

Technical data, 17

Telegram time, 19

Terminal data, 31

Total cable length, 27

Transmission cable

- checking the use of repeaters, 29
- segment cable length, 28
- specification, 26
- total cable length, 27

Transmit PDO communication parameters (I-1800 ... I-1802), 112

Transmit PDO mapping parameters (I-1A00 ... I-1A02), 114

Tx mode CAN-OUT1...3, 140

Type code, 14

- finding, 14

Type of insulation, 18

**U**

User data, 44, 52, 73, 75

- process data telegram from drive, 59
- process data telegram to the drive, 56

**V**

Validity of SDO2 and PDOs, 134

Validity of the documentation, 6

Voltage supply, 17, 30

- internal, 30

Voltage supply: external, 30

**W**

Wiring according to EMC, 23

Wiring with a host (master), 24



© 06/2013



Lenze Drives GmbH  
Postfach 10 13 52  
D-31763 Hameln  
Germany



+49 (0)51 54 / 82-0



+49 (0)51 54 / 82-28 00



Lenze@Lenze.de



www.Lenze.com

#### Service

Lenze Service GmbH  
Breslauer Straße 3  
D-32699 Extertal  
Germany



00 80 00 / 24 4 68 77 (24 h helpline)



+49 (0)51 54 / 82-11 12



Service@Lenze.de

EDSMF2178IB ■ 13437291 ■ EN ■ 3.0 ■ TD17

10 9 8 7 6 5 4 3 2 1

# 2175 (CANopen) fieldbus module

## Contents

## 7 2175 (CANopen) fieldbus module

### 7.1 Contents

|        |                                                                            |        |
|--------|----------------------------------------------------------------------------|--------|
| 7.2    | General information .....                                                  | 7.2-1  |
| 7.3    | Technical data .....                                                       | 7.3-1  |
| 7.3.1  | General data and application conditions .....                              | 7.3-1  |
| 7.3.2  | Rated data .....                                                           | 7.3-1  |
| 7.3.3  | Communication times .....                                                  | 7.3-2  |
| 7.3.4  | Dimensions .....                                                           | 7.3-4  |
| 7.4    | Installation .....                                                         | 7.4-1  |
| 7.4.1  | Components of the communication module .....                               | 7.4-1  |
| 7.4.2  | Mechanical installation .....                                              | 7.4-2  |
| 7.4.3  | Electrical installation .....                                              | 7.4-3  |
| 7.4.4  | Bus cable length .....                                                     | 7.4-7  |
| 7.5    | Commissioning .....                                                        | 7.5-1  |
| 7.5.1  | Before switching on .....                                                  | 7.5-3  |
| 7.5.2  | Enable drive via 2175 fieldbus module .....                                | 7.5-4  |
| 7.6    | Data transfer .....                                                        | 7.6-1  |
| 7.6.1  | Structure of a CAN data telegram .....                                     | 7.6-2  |
| 7.6.2  | Process data channel .....                                                 | 7.6-8  |
| 7.6.3  | Process data transfer .....                                                | 7.6-9  |
| 7.6.4  | Process-data assignments for 82XX .....                                    | 7.6-11 |
| 7.6.5  | Process data assignment for 8200 vector .....                              | 7.6-13 |
| 7.6.6  | Process-data assignment for 93XX .....                                     | 7.6-17 |
| 7.6.7  | Process data assignment for 9300 Servo PLC and Drive PLC .....             | 7.6-22 |
| 7.6.8  | Parameter data channel .....                                               | 7.6-26 |
| 7.6.9  | Examples for parameter data telegrams .....                                | 7.6-31 |
| 7.6.10 | Notes to be observed when setting the parameters for the controllers ..... | 7.6-34 |
| 7.7    | CANopen objects and Lenze codes .....                                      | 7.7-1  |
| 7.7.1  | Communication relevant Lenze codes .....                                   | 7.7-1  |
| 7.7.2  | Description of communication relevant Lenze codes .....                    | 7.7-2  |
| 7.7.3  | Implemented CANopen objects .....                                          | 7.7-20 |
| 7.8    | Troubleshooting .....                                                      | 7.8-1  |
| 7.9    | Appendix .....                                                             | 7.9-1  |
| 7.9.1  | Code table .....                                                           | 7.9-1  |
| 7.10   | Index .....                                                                | 7.10-1 |



# 2175 (CANopen) fieldbus module

## General information

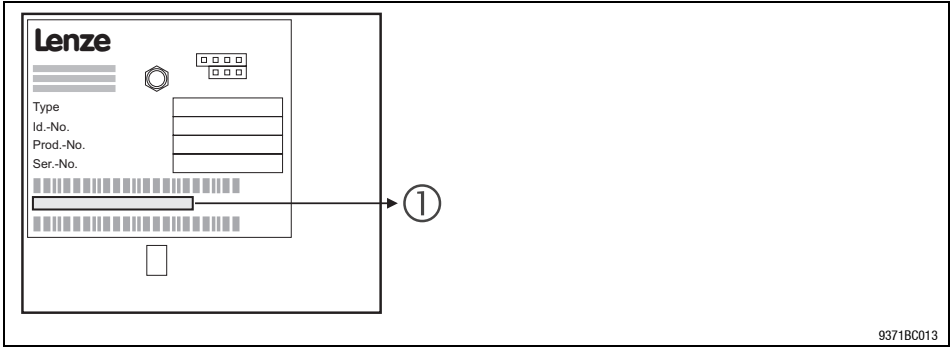
### 7.2 General information

**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:

|                 |         |     |     |      |                                    |
|-----------------|---------|-----|-----|------|------------------------------------|
| 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)                      |
| 82EVxxxxBxxxXX  |         | Vx  | 1x  |      | (8200 vector)                      |
| 82CVxxxxBxxxXX  |         | 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)       |

|                  |  |
|------------------|--|
| Type             |  |
| Design           |  |
| Hardware version |  |
| Software version |  |
| Variant          |  |
| Explanation      |  |

## 2175 (CANopen) fieldbus module

### General information

#### 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 ( $R = 120\ \Omega$ )
- Easy connection because of pluggable screw terminals



2175 (CANopen) fieldbus module

Technical data  
General data and application conditions

7.3 Technical data

7.3.1 General data and application conditions

| Field                                           | Values                                                                      |
|-------------------------------------------------|-----------------------------------------------------------------------------|
| Order name                                      | EMF2175IB                                                                   |
| Communication medium                            | DIN ISO 11898                                                               |
| Network 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   |

# 2175 (CANopen) fieldbus module

Technical data  
Communication times

Processing times  
821X/822X/8200 vector

|              |                                                                                           |
|--------------|-------------------------------------------------------------------------------------------|
| Parameters   | 30 ... 50 ms                                                                              |
| Process data | 3 ... 5 ms<br>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)<br>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  $t_{Tmax}$ .

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

$t_T$  = telegram time [ms]  
 $L_D$  = telegram length [byte]  
 $d_U$  = baud rate [kBit/s]

7.3.4 Dimensions

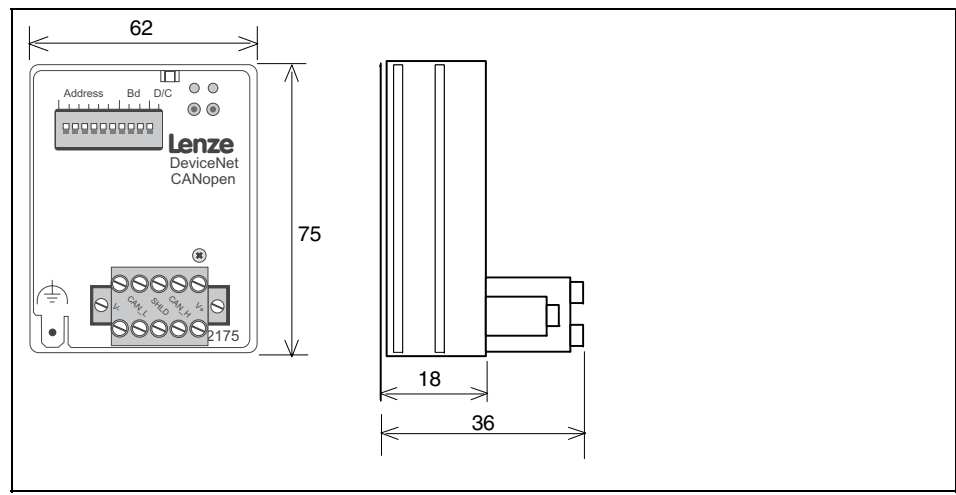


Fig. 7.3-1 Dimensions of the 2175 fieldbus module (all dimensions in mm)

# 2175 (CANopen) fieldbus module

Installation  
Components of the communication module

## 7.4 Installation

### 7.4.1 Components of the communication module

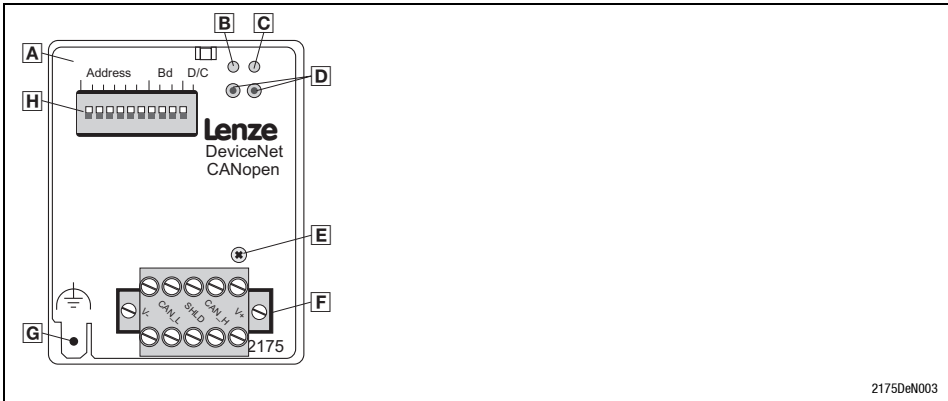


Fig. 7.4-1 2175 communication module

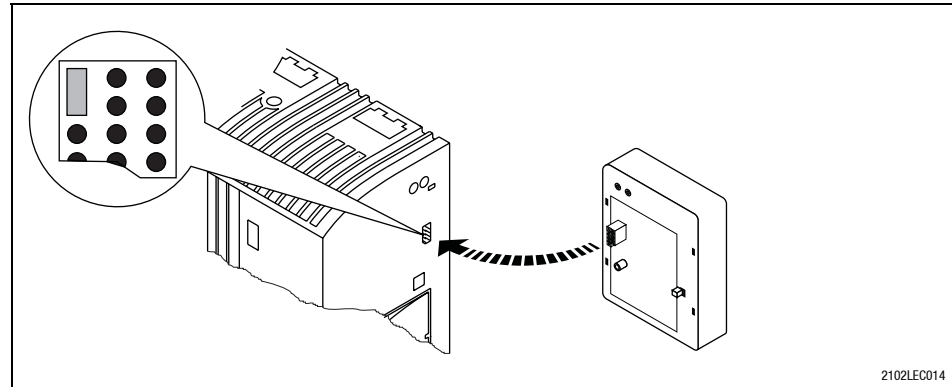
| Pos. | Designation                                            | Meaning                                                                                                                                     | Notes                                                                                                                                                        |                                                                                                                                                    |
|------|--------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| 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                                                                                                                                         | <ul style="list-style-type: none"><li>• No communication with the communication module</li><li>• Communication module is not supplied with voltage</li></ul> |                                                                                                                                                    |
|      |                                                        | 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



2102LEC014

- 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](#) .

# 2175 (CANopen) fieldbus module

Installation  
Electrical installation

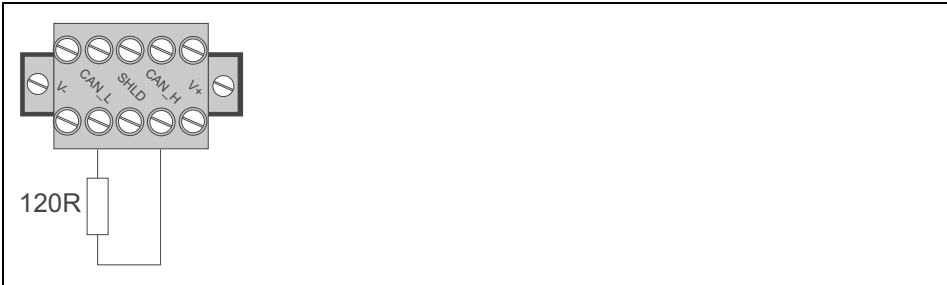
### 7.4.3 Electrical installation



**Note!**  
Only for 820X and 821X:  
If necessary use an additional PE shield cable which avoids EMC-related communication interference in especially noisy environments.

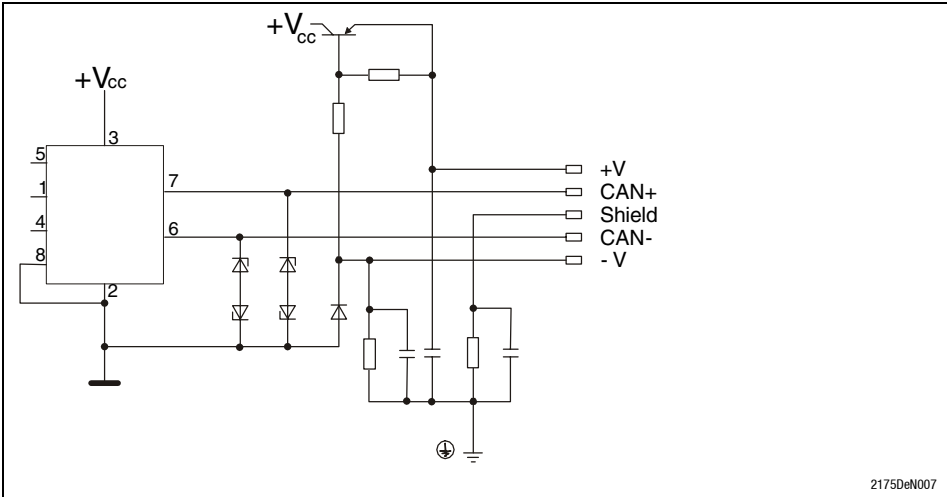
Assignment of the plug connector

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



2175DeN007

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-  $\pm$  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<br>only external voltage supply possible | Internal voltage supply |
|--------------------------------------------------------|-------------------------|
|                                                        |                         |

# 2175 (CANopen) fieldbus module

Installation

Electrical installation

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

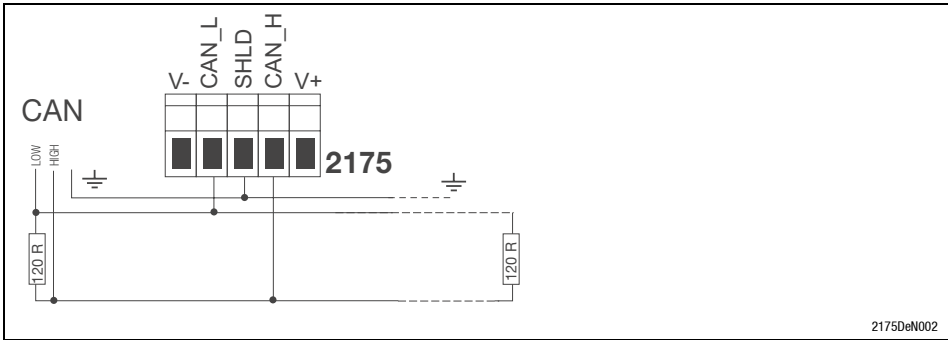


Fig. 7.4-2 Connection to the plug connector

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 <sup>2</sup><br>(twisted in pairs with shield) | CYPIMF 2 x 2 x 0.5 mm <sup>2</sup><br>(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 sent 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.

## 2175 (CANopen) fieldbus module

7

### Installation

7.4

### Bus cable length

7.4.4

#### 7.4.4 Bus cable length

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

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.

| Station | Cable cross-section  |                     |                      |                     |
|---------|----------------------|---------------------|----------------------|---------------------|
|         | 0.25 mm <sup>2</sup> | 0.5 mm <sup>2</sup> | 0.75 mm <sup>2</sup> | 1.0 mm <sup>2</sup> |
| 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. Repeater 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 <sup>2</sup> (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 <sup>2</sup> |
| • 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 <sup>2</sup> . |                     |
| 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) <u>minus</u> 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 address 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

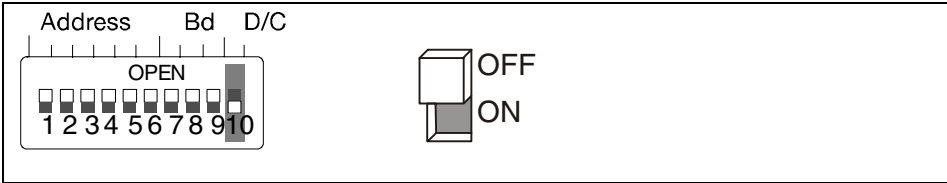


Fig. 7.5-1 Communication profile setting

|                       |     |
|-----------------------|-----|
| Communication profile | S10 |
| DeviceNet             | OFF |
| CANopen               | ON  |

## Device address setting


**Note!**

Please observe that the addresses are not the same when using several controllers.

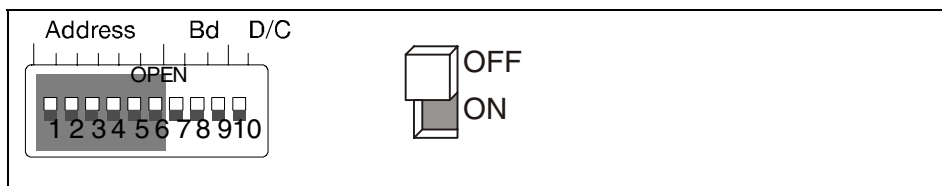


Fig. 7.5-2 Setting of the controller address

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

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.

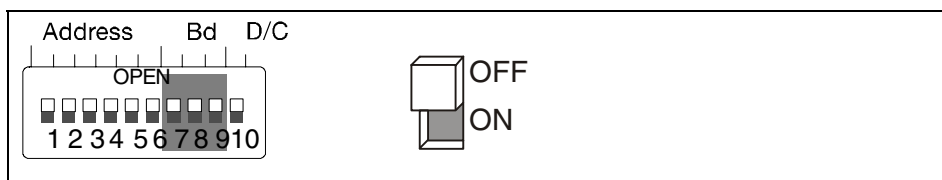


Fig. 7.5-3 Baud rate setting

| 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 |

# 2175 (CANopen) fieldbus module

Commissioning  
Before switching on

## 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.

|                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>82XX / 8200 vector</b> | <p>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.</p> <p>Examples for Write (L-C0001=3):</p> <ul style="list-style-type: none"> <li>– Index = 5FFE<sub>hex</sub> (results from: 5FFF<sub>hex</sub> – (L-C0001)<sub>hex</sub>;)             <ul style="list-style-type: none"> <li>– Subindex: 0</li> <li>– Value: 30000<sub>dec</sub> (results from: L-C0001 = 3 x 10000)</li> </ul> </li> </ul> <p>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.</p> <ul style="list-style-type: none"> <li>– 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).</li> </ul> <p>The controller now accepts parameter and process data.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>93XX</b>               | <p>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.</p> <p>Examples for Write (L-C0005=1013):</p> <ul style="list-style-type: none"> <li>– Index = 5FFA<sub>hex</sub> (results from: 5FFF<sub>hex</sub> – (L-C0005)<sub>hex</sub>;)             <ul style="list-style-type: none"> <li>– Subindex: 0</li> <li>– Value: 10130000<sub>dec</sub> (results from: L-C0005 = 1013 x 10000)</li> </ul> </li> </ul> <p>2. Set the parameter L-C0142 to 0. Observe the chapter “Protection against unexpected start”.</p> <p>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.</p> <ul style="list-style-type: none"> <li>– 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).</li> </ul> <p><b>Note</b><br/>With the signal configuration L-C0005=xx13, terminal A1 is switched as voltage output. Connect the following terminals:</p> <ul style="list-style-type: none"> <li>– X5.A1 with X5.28 (controller enable).</li> <li>– X5.A1 with X5.E1 (CW/QSP)</li> </ul> <p>The controller now accepts parameter and process data.</p> |

## 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         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|---------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Parameter data</b><br>(SDO,<br>Service-Data-Objects) | These are, for instance, <ul style="list-style-type: none"><li>• Operationg parameters</li><li>• Diagnostics information</li><li>• Motor data</li></ul> In general, the parameter transfer is not as time-critical as the tranfer of process data. | <b>Parameter data channel</b> | <ul style="list-style-type: none"><li>• Enables the access to all Lenze codes and the CANopen index.</li><li>• Parameter changes are usually stored automatically in the controller (observe L-C0003).</li></ul>                                                                                                                                                                                                                                                    |
| <b>Process data</b><br>(PDO,<br>Process-Data-Objects)   | These are, for instance, <ul style="list-style-type: none"><li>• Setpoints</li><li>• Actual values</li></ul> Exchange between host and controller required as fast as possible. Small amounts of data which can be transferred cyclically.         | <b>Process data channel</b>   | <ul style="list-style-type: none"><li>• You can control the controller using the process data.</li><li>• The host has direct access to the process data. Data are for instance directly assigned to the I/O area.</li><li>• Process data are<ul style="list-style-type: none"><li>– not stored in the controller.</li><li>– transferred between host and controller in order to provide a continuous exchange of current input and output data.</li></ul></li></ul> |

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

7.6.1 Structure of a CAN data telegram

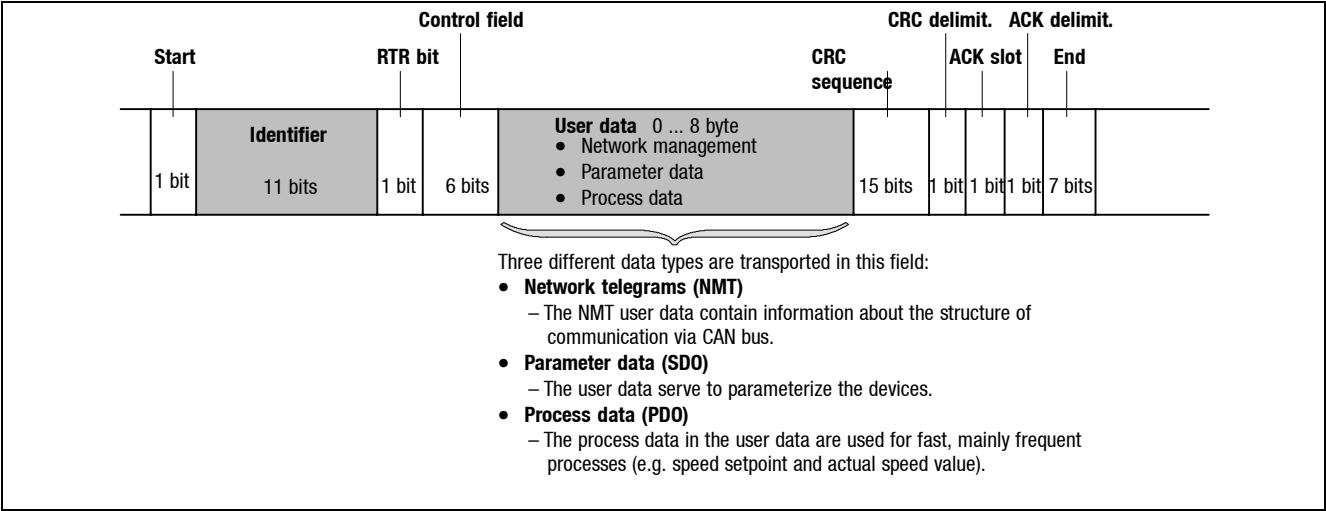


Fig. 7.6-1 Basic structure of the CAN data telegram

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](http://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.

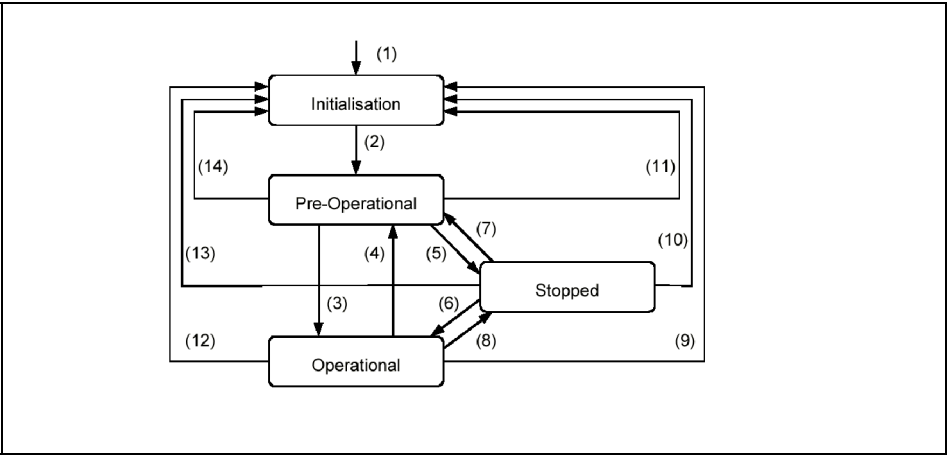
The identifier allocation is defined in the CANopen protocol. According to Lenze settings, the basic identifier is preset with the following values (see CiA DS301, Pre-Defined Connection Set):

|                          | 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<br>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,<br>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,<br>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   | <b>Controller address: xx</b><br>The following applies to the assignment of the bytes marked with "xx" in the table below: <ul style="list-style-type: none"><li>• <b>xx = 00<sub>hex</sub></b><br/>With this assignment, all controllers connected are addressed by the telegram. All controller can change their status at the same time.</li><li>• <b>xx = Controller address</b><br/>If a certain address is indicated, the status will only be changed for the controller addressed.</li></ul> |



| State transition | Command (hex) | Network status after change | Effect on process and parameter data after state transition                                                                                                                                                                                                                                                                                                      |
|------------------|---------------|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (1)              | -             | Initialisation              | At power-on the initialisation <u>automatically</u> started. The drive does not take part in the data transfer.                                                                                                                                                                                                                                                  |
| (2)              | -             | Pre-Operational             | After initialisation <ul style="list-style-type: none"><li>• the participant automatically passes over to the Pre-Operational state.</li><li>• the master decides how the controller/s are to participate in the communication.</li></ul> 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 parameters in the field bus module with the stored values (corresponds to "Reset-Node")                                                                                                                                                                                                                                                    |
| (10)             |               |                             |                                                                                                                                                                                                                                                                                                                                                                  |
| (11)             |               |                             |                                                                                                                                                                                                                                                                                                                                                                  |
| (12)             |               |                             |                                                                                                                                                                                                                                                                                                                                                                  |
| (13)             | 82 xx         | Initialisation              | Initialisation of communication-relevant parameters (CIA DS 301) in the field bus module with the stored values (corresponds to "Reset Communication")                                                                                                                                                                                                           |
| (14)             |               |                             |                                                                                                                                                                                                                                                                                                                                                                  |

# 2175 (CANopen) fieldbus module

Data transfer  
Structure of a CAN data telegram

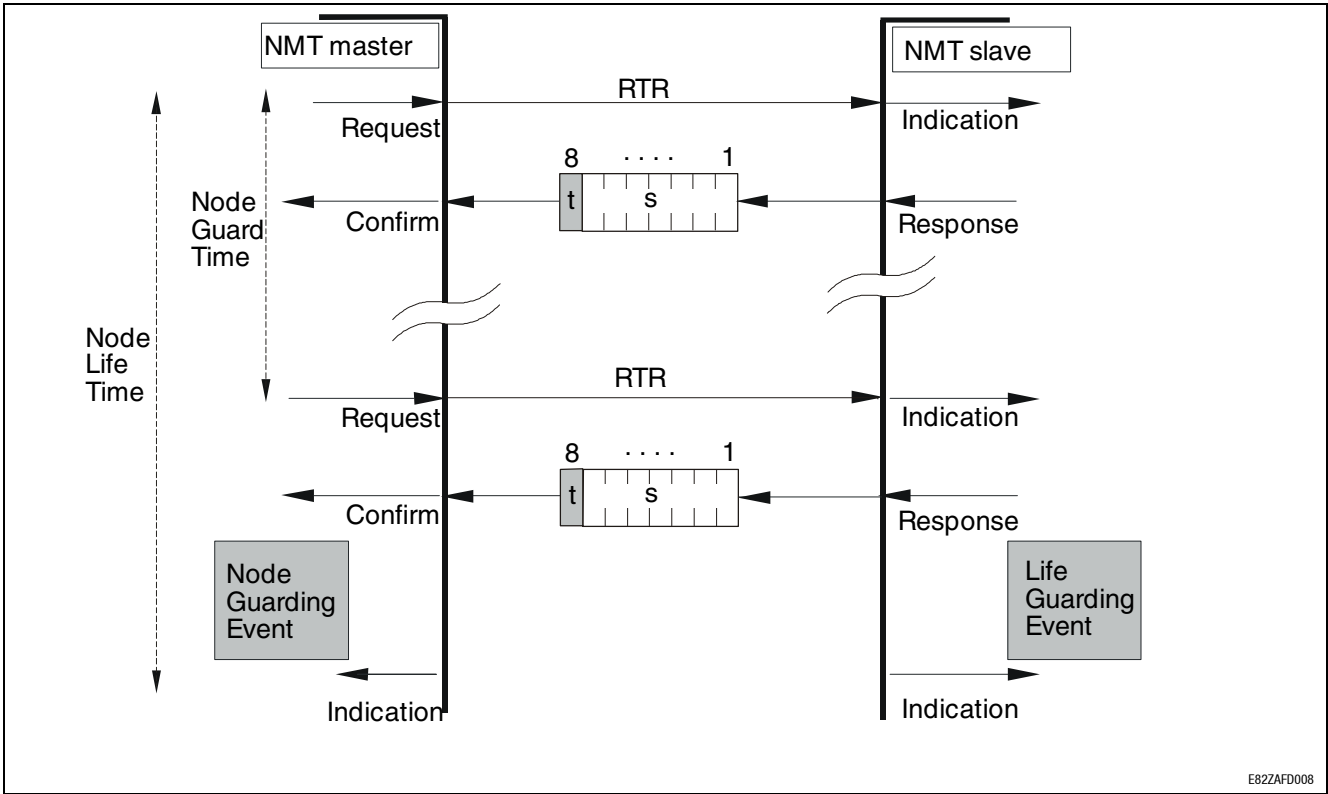
Node Guarding Protocol

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 ( $1792_{dec} + \text{slave address}$ , (7.6-3)).

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

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

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.

# 2175 (CANopen) fieldbus module

## Data transfer Structure of a CAN data telegram

### 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<sub>hex</sub>). 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<sub>hex</sub>). 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.

Data transfer  
Process data transfer

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.

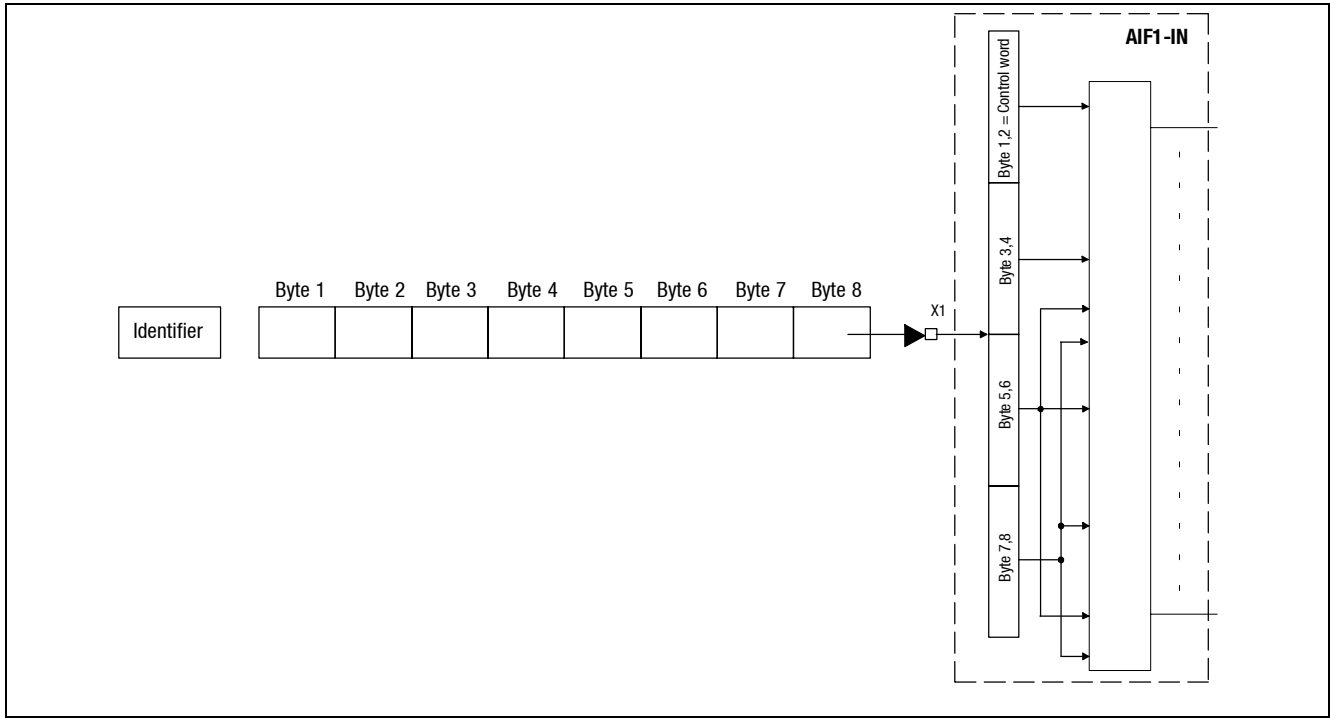


Fig. 7.6-2 Example: Device series 93XX

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.

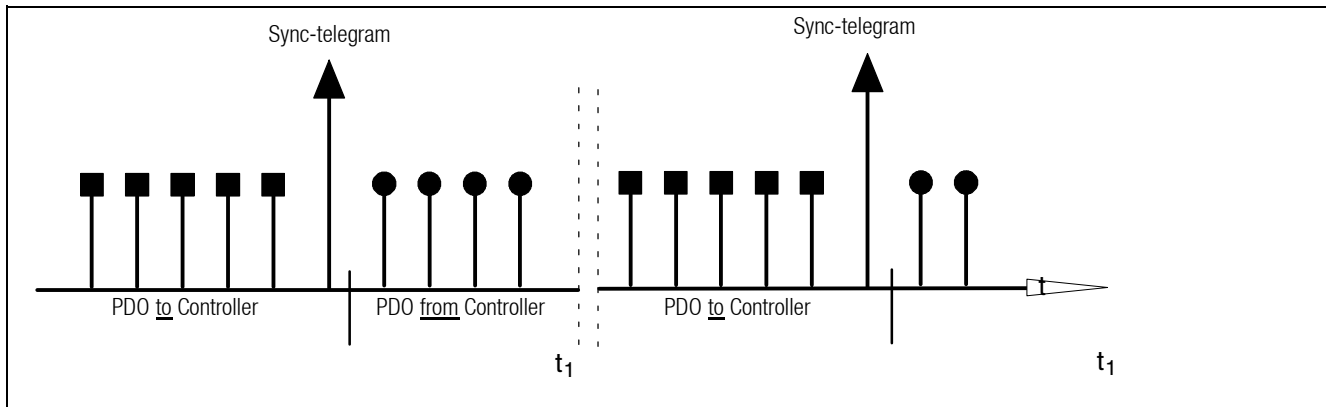


Fig. 7.6-3 Synchronisation of cyclic PDOs (represented by a bus participant)

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!

# 2175 (CANopen) fieldbus module

7

## Data transfer

7.6

## Process-data assignments for 82XX

7.6.4

### 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<br>L-C0135<br>Low byte | Control word<br>L-C0135<br>High byte | Setpoint<br>L-C0046<br>Low byte | Setpoint<br>L-C0046<br>High byte | xx     | xx     | xx     | xx     |

|                                                           |        |                                                                                                                                                    |
|-----------------------------------------------------------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| More information<br>about the process<br>data<br>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))<br>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)

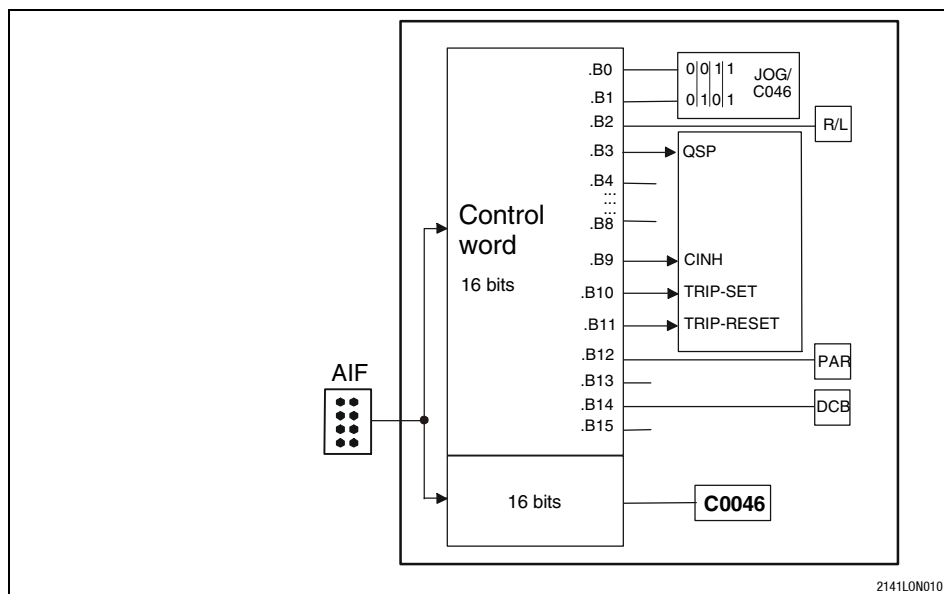


Fig. 7.6-4 Access to the control word and the frequency setpoint in 82XX (fixed assignment, see (□ 7.6-15))

Process data telegram from 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<br>L-C0150<br>Low byte | Status word<br>L-C0150<br>High byte                                                                                                                | Actual value<br>L-C0050<br>Low byte | Actual value<br>L-C0050<br>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)).<br>The description of the bits can be obtained from the Code Table. |                                     |                                      |        |        |        |        |
|                                                   | Byte 3                             | The actual frequency value with signed normalization (L-C0050)                                                                                     |                                     |                                      |        |        |        |        |
|                                                   | Byte 4                             | 24000 = 480 Hz is provided here.                                                                                                                   |                                     |                                      |        |        |        |        |
|                                                   | Byte 5                             | No evaluation of these data, any content is possible                                                                                               |                                     |                                      |        |        |        |        |
|                                                   | Byte 6                             |                                                                                                                                                    |                                     |                                      |        |        |        |        |
|                                                   | Byte 7                             |                                                                                                                                                    |                                     |                                      |        |        |        |        |
|                                                   | Byte 8                             |                                                                                                                                                    |                                     |                                      |        |        |        |        |

Status word: see (7.6-16)

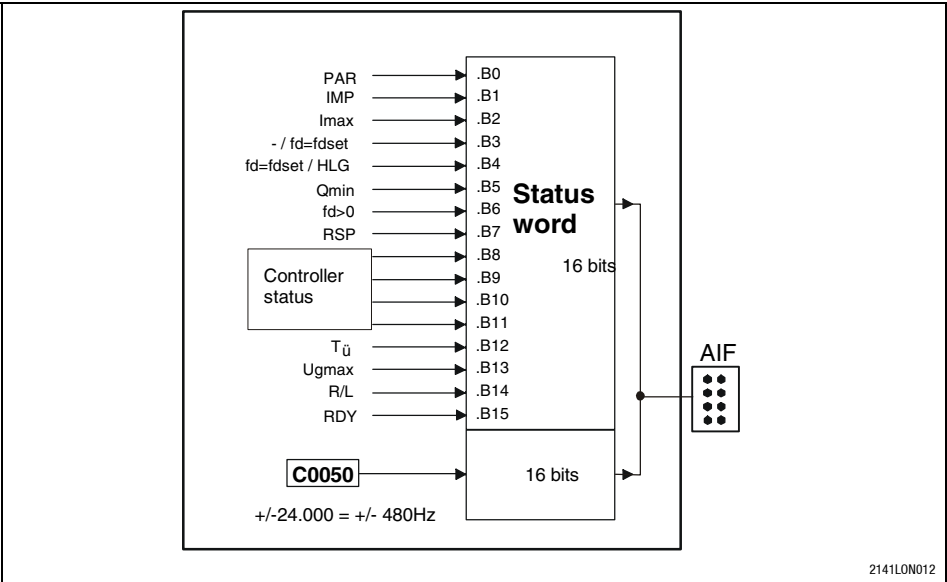


Fig. 7.6-5 Read access to the status word and the actual frequency value in 82XX (fixed assignment, see (7.6-16))

# 2175 (CANopen) fieldbus module

Data transfer

Process data assignment for 8200 vector

### 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  $\pm 24000 \equiv \pm 480 \text{ Hz}$ .

Process data telegram to drive

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

Control word: see ( 7.6-15)

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

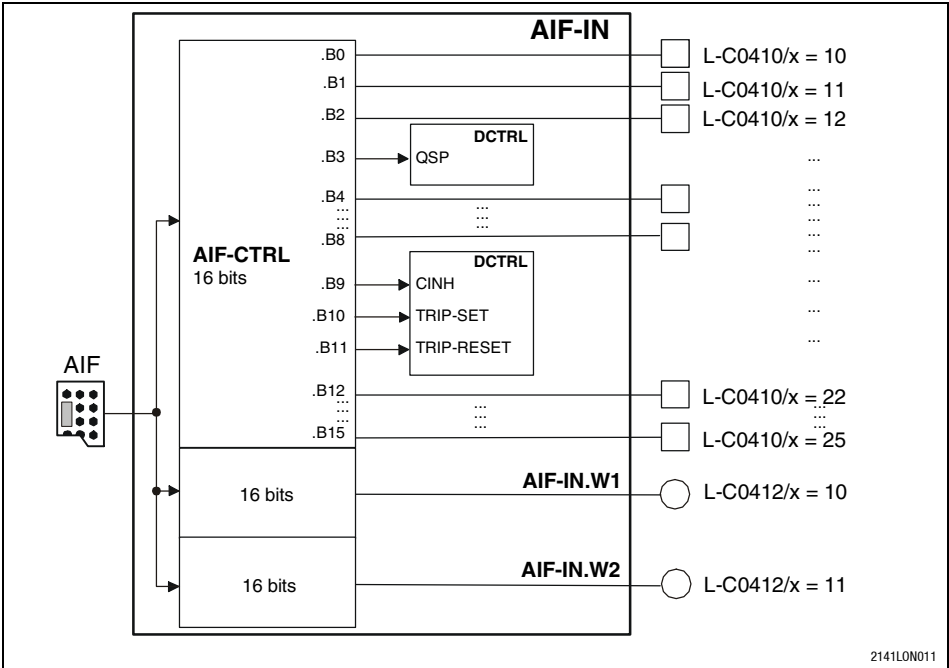


Fig. 7.6-6 Function block AIF-IN in 8200 vector (freely programmable assignment, factory setting see ( 7.6-15))

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<br>Low byte | Status word<br>High byte | AIF-OUT.W1<br>Low byte | AIF-OUT.W1<br>High byte | AIF-OUT.W2<br>Low byte | AIF-OUT.W2<br>High byte | xx     | xx     |

Status word: see ( 7.6-16)

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

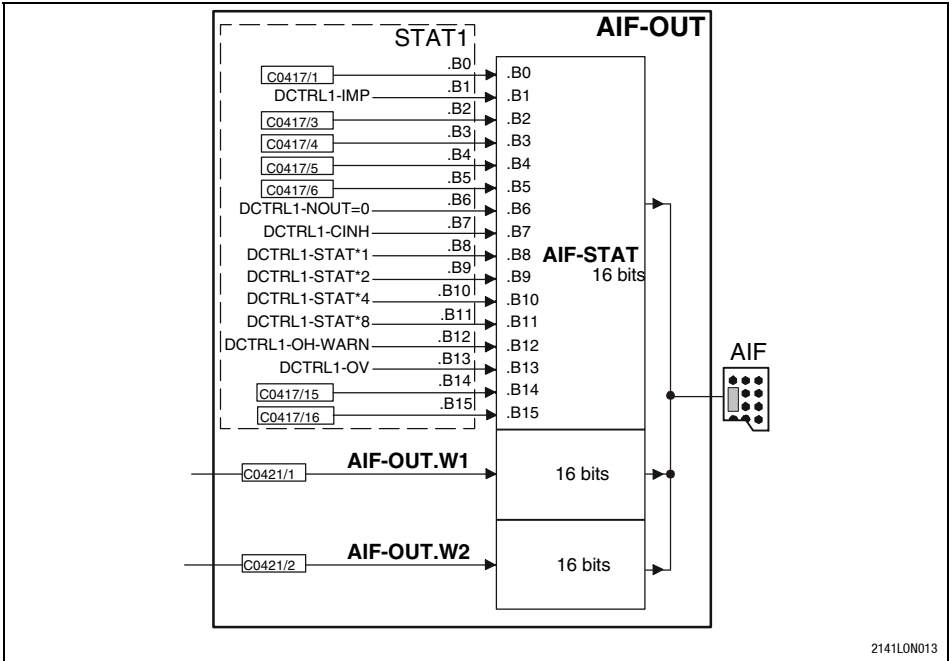


Fig. 7.6-7 Function block AIF-OUT in 8200 vector (freely programmable assignment, factory setting see ( 7.6-16))

# 2175 (CANopen) fieldbus module

7

## Data transfer

7.6

## Process data assignment for 8200 vector

7.6.5

Control word for 82XX and 8200 vector

| Bit  | 820X                                                                                                     | 821x, 822x                                                                                                     | 8200 vector                                                                                                             |                                           |
|------|----------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|
|      |                                                                                                          |                                                                                                                | Default setting:<br>C0001=3 if C0007 < 52                                                                               | Default setting:<br>C0001=3 if C0007 > 51 |
| 0, 1 | 00 = C0046 active<br>01 = JOG1 in C0037 active<br>10 = JOG2 in C0038 active<br>11 = JOG3 in C0039 active | 00 = C0046 active<br>01 = JOG1 in C0037 active<br>10 = JOG2 in C0038 active<br>11 = JOG3 in C0039 active       | 00 = C0046 active<br>01 = NSET1-JOG1 (C0037) active<br>10 = NSET1-JOG2 (C0038) active<br>11 = NSET1-JOG3 (C0039) active | freely configurable through user          |
| 2    | CW/CCW (CW rotation/CCW rotation)<br>0 = CW rotation<br>1 = CCW rotation                                 | CW/CCW (CW rotation/CCW rotation)<br>0 = CW rotation<br>1 = CCW rotation                                       | DCTRL1-CW/CCW<br>0 = not active<br>1 = active                                                                           |                                           |
| 3    | QSP (quick stop)<br>0 = QSP not active<br>1 = QSP active                                                 | QSP (quick stop)<br>0 = QSP not active<br>1 = QSP active                                                       | AIF-CTRL-QSP<br>0 = not active<br>1 = active                                                                            |                                           |
| 4    | Reserved                                                                                                 | RFG stop (stop of the ramp function generator)<br>0 = RFG stop not active<br>1 = RFG stop active               | NSET1-RFG1-STOP<br>0 = not active<br>1 = active                                                                         | freely configurable through user          |
| 5    | Reserved                                                                                                 | RFG zero (deceleration along the T <sub>if</sub> ramp C0013)<br>0 = RFG zero not active<br>1 = RFG zero active | NSET1-RFG1-0<br>0 = not active<br>1 = active                                                                            |                                           |
| 6    | Reserved                                                                                                 | UP function for motor potentiometer<br>0 = UP not active<br>1 = UP active                                      | MPOT1-UP<br>0 = not active<br>1 = active                                                                                |                                           |
| 7    | Reserved                                                                                                 | DOWN function for motor potentiometer<br>0 = DOWN not active<br>1 = DOWN active                                | MPOT1-DOWN<br>0 = not active<br>1 = active                                                                              |                                           |
| 8    | Reserved                                                                                                 | Reserved                                                                                                       | freely configurable through user                                                                                        |                                           |
| 9    | Ctrl. inhibit (controller inhibit)<br>0 = controller not inhibited<br>1 = controller inhibited           | Ctrl. inhibit (controller inhibit)<br>0 = controller not inhibited<br>1 = controller inhibited                 | AIF-CTRL-CINH<br>0 = not active<br>1 = active                                                                           |                                           |
| 10   | Reserved                                                                                                 | Reserved                                                                                                       | AIF-CTRL-TRIP-SET<br>0 = not active<br>1 = active                                                                       |                                           |
| 11   | Reserved                                                                                                 | TRIP reset<br>0 -> 1 = Edge from 0 to 1                                                                        | AIF-CTRL-TRIP-RESET<br>0 -> 1 = Edge from 0 to 1                                                                        |                                           |
| 12   | PAR1 (Parameter set changeover)<br>0 -> 1 = Parameter set<br>1 -> 0 = Parameter set                      | PAR1 (Parameter set changeover)<br>0 -> 1 = Parameter set<br>1 -> 0 = Parameter set                            | DCTRL1-PAR2/4<br>0 = not active<br>1 = active                                                                           | freely configurable through user          |
| 13   | Reserved                                                                                                 | Reserved                                                                                                       | DCTRL1-PAR3/4<br>0 = not active<br>1 = active                                                                           |                                           |
| 14   | DC brake (DC injection brake)<br>0 = DC brake not active<br>1 = DC brake active                          | DC brake (DC injection brake)<br>0 = DC brake not active<br>1 = DC brake active                                | MCTRL1-DCB<br>0 = not active<br>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<br>0 = Parameter set 1 or 3 active<br>1 = Parameter set 2 or 4 active        | Actual parameter set<br>0 = Parameter set 1 or 3 active<br>1 = Parameter set 2 or 4 active                                                                                                                                            | DCTRL1-PAR-B0                                                                                                                                                                                                                         |
| 1      | IMP (pulse inhibit)<br>0 = Pulses for power stage enabled<br>1 = Pulses for power stage inhibited | IMP (pulse inhibit)<br>0 = Pulses for power stage enabled<br>1 = Pulses for power stage inhibited                                                                                                                                     | DCTRL1-IMP                                                                                                                                                                                                                            |
| 2      | $I_{\max}$ (current limit reached)<br>0 = Current limit not reached<br>1 = current limit reached  | $I_{\max}$ (current limit reached)<br>0 = Current limit not reached<br>1 = current limit reached                                                                                                                                      | MCTRL1-IMAX                                                                                                                                                                                                                           |
| 3      | Not assigned                                                                                      | $f_d = f_{dset}$<br>0 = $f_d \neq f_{dset}$<br>1 = $f_d = f_{dset}$                                                                                                                                                                   | MCTRL1-RFG1=NOUT                                                                                                                                                                                                                      |
| 4      | $f_d = f_{dset}$<br>0 = $f_d \neq f_{dset}$<br>1 = $f_d = f_{dset}$                               | RFG on = RFG off<br>0 = RFG on $\neq$ RFG off<br>1 = RFG on = RFG out                                                                                                                                                                 | NSET1-RFG1-I=0                                                                                                                                                                                                                        |
| 5      | $Q_{\min}$ ( $f_d \leq f_{dQ\min}$ )<br>0 = $Q_{\min}$ not active<br>1 = $Q_{\min}$ active        | $Q_{\min}$ ( $f_d \leq f_{dQ\min}$ )<br>0 = $Q_{\min}$ not active<br>1 = $Q_{\min}$ active                                                                                                                                            | PCTRL1-QMIN                                                                                                                                                                                                                           |
| 6      | $f_d + 0$ (act. frequency = 0)<br>0 = $f_d \neq 0$<br>1 = $f_d + 0$                               | $f_d + 0$ (act. frequency = 0)<br>0 = $f_d \neq 0$<br>1 = $f_d + 0$                                                                                                                                                                   | DCTRL1-NOUT=0                                                                                                                                                                                                                         |
| 7      | Ctrl. inhibit (controller inhibit)<br>0 = controller not inhibited<br>1 = controller inhibited    | Ctrl. inhibit (controller inhibit)<br>0 = controller not inhibited<br>1 = controller inhibited                                                                                                                                        | DCTRL1-CINH                                                                                                                                                                                                                           |
| 8...11 | Controller status<br>0 = Controller initialisation<br>8 = Error active                            | Controller status<br>0 = Controller initialisation<br>2 = Switch-on inhibit<br>3 = Operation inhibited<br>4 = Flying-restart circuit active<br>5 = DC brake active<br>6 = Operation enabled<br>7 = Message active<br>8 = Error active | Controller status<br>0 = Controller initialisation<br>2 = Switch-on inhibit<br>3 = Operation inhibited<br>4 = Flying-restart circuit active<br>5 = DC brake active<br>6 = Operation enabled<br>7 = Message active<br>8 = Error active |
| 12     | Overtemperature warning<br>0 = No warning<br>1 = warning                                          | Overtemperature warning<br>0 = No warning<br>1 = warning                                                                                                                                                                              | DCTRL1-OH-WARN                                                                                                                                                                                                                        |
| 13     | $V_{G\max}$ (DC-bus overvoltage)<br>0 = No overvoltage<br>1 = overvoltage                         | $V_{G\max}$ (DC-bus overvoltage)<br>0 = No overvoltage<br>1 = overvoltage                                                                                                                                                             | DCTRL1-OV                                                                                                                                                                                                                             |
| 14     | Direction of rotation<br>0 = CW rotation<br>1 = CCW rotation                                      | Direction of rotation<br>0 = CW rotation<br>1 = CCW rotation                                                                                                                                                                          | DCTRL1-CCW                                                                                                                                                                                                                            |
| 15     | Ready for operation<br>0 = not ready for operation<br>1 = ready for operation                     | Ready for operation<br>0 = not ready for operation<br>1 = ready for operation                                                                                                                                                         | DCTRL1-RDY                                                                                                                                                                                                                            |

2175 (CANopen) fieldbus module

Data transfer  
Process-data assignment for 93XX

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<br>Low byte | Control word<br>High byte | AIF-IN.W1<br>Low byte | AIF-IN.W1<br>High byte | AIF-IN.W2<br>Low byte | AIF-IN.W2<br>High byte | AIF-IN.W3<br>Low byte | AIF-IN.W3<br>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.

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

## Control word for 93XX

| 9300  | 9300 Servo          |                     |                     |                     | 9300<br>Positioning<br>controller | 9300<br>Cam profiler | 9300 Vector                                |                     |                     |
|-------|---------------------|---------------------|---------------------|---------------------|-----------------------------------|----------------------|--------------------------------------------|---------------------|---------------------|
| C0005 | 1xx3                | 4xx3                | 5xx3                | 6xx3,7xx3           | 2xxx3                             | xxx3                 | 1xxx, 2xxx,<br>3xxx, 5xxx,<br>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    | DCTRL-PAR-LOAD      | DCTRL-PAR-LOAD      | DCTRL-PAR-LOAD      | DCTRL-PAR-LOAD      | POS-PARAM-RD                      | not assigned         | DCTRL-PAR-LOAD                             | DCTRL-PAR-LOAD      | DCTRL-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.

# 2175 (CANopen) fieldbus module

7

## Data transfer

7.6

## Process-data assignment for 93XX

7.6.6

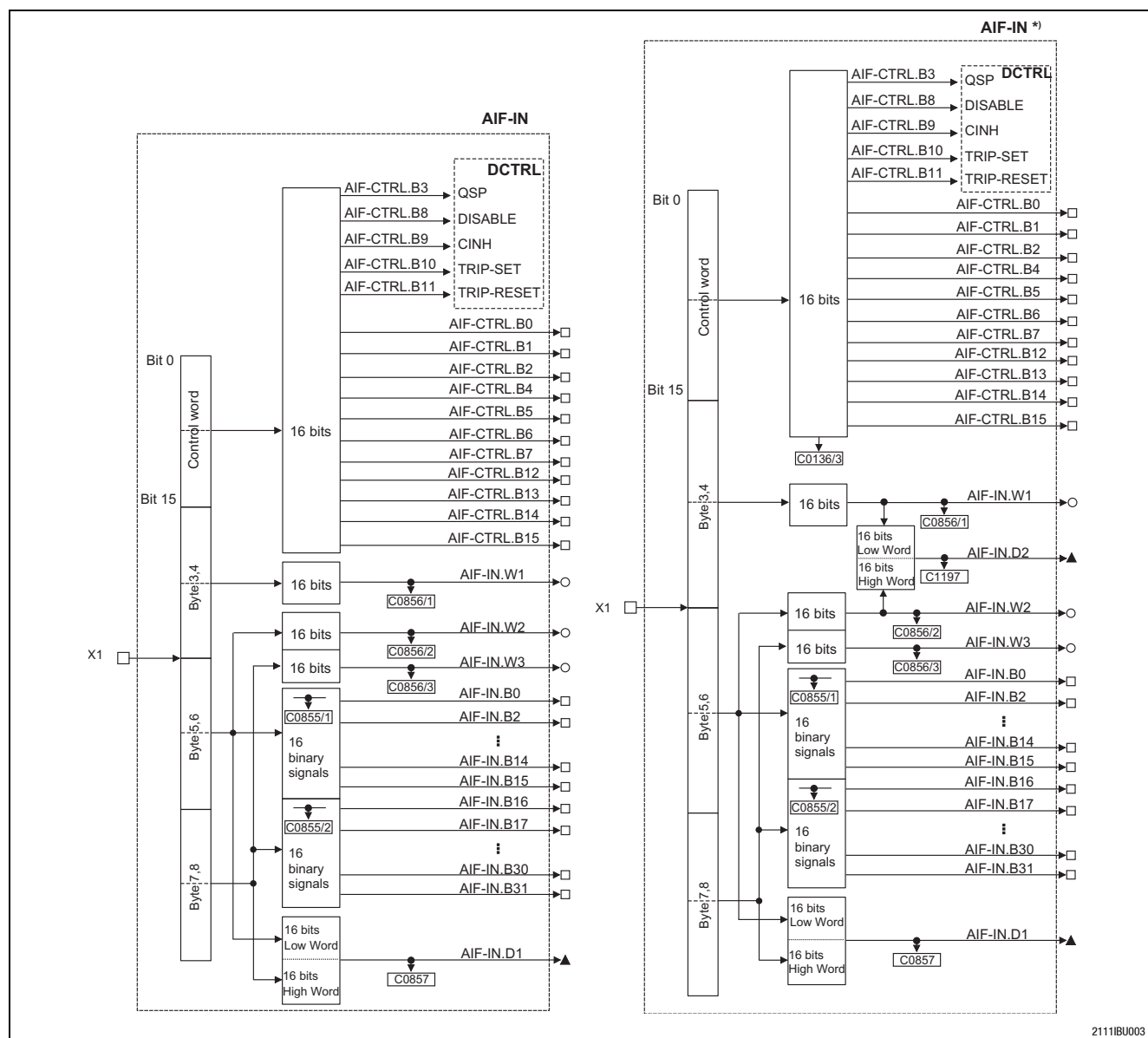


Fig. 7.6-8

Function blocks AIF-IN and AIF-IN<sup>\*)</sup>

AIF-IN<sup>\*)</sup> 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 from drive**

| Byte 1                   | Byte 2                  | Byte 3                  | Byte 4                 | Byte 5                  | Byte 6                 | Byte 7                  | Byte 8                 |
|--------------------------|-------------------------|-------------------------|------------------------|-------------------------|------------------------|-------------------------|------------------------|
| Status word<br>High byte | Status word<br>Low byte | AIF-OUT.W1<br>High byte | AIF-OUT.W1<br>Low byte | AIF-OUT.W2<br>High byte | AIF-OUT.W2<br>Low byte | AIF-OUT.W3<br>High byte | AIF-OUT.W3<br>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<br>(L-C0005) |                              | AIF-OUT.W1                                     | AIF-OUT.W2                                     | AIF-OUT.W3                                                | AIF-OUT.D1   |
|-----------------------------------|------------------------------|------------------------------------------------|------------------------------------------------|-----------------------------------------------------------|--------------|
| Speed control                     | 1003/1013/11<br>13           | MCTRL-NACT<br>Actual speed<br>100 % = 16383    | MCTRL-MSET2<br>Torque display<br>100 % = 16383 | MCTRL-NSET2<br>Speed controller<br>input<br>100 % = 16383 | not assigned |
| Torque control                    | 4003/4013/41<br>13           | MCTRL-MSET2<br>Torque display<br>100 % = 16383 | MCTRL-NACT<br>Act. speed in %<br>100 % = 16383 | MCTRL-NSET2<br>Speed controller<br>input<br>100 % = 16383 |              |
| DF master                         | 5003/5013/51<br>13           | MCTRL-NACT<br>Actual speed<br>100 % = 16383    | MCTRL-MSET2<br>Torque display<br>100 % = 16383 | MCTRL-NSET2<br>Speed controller<br>input<br>100 % = 16383 |              |
| DF-slave bus                      | 6003/6013/61<br>13           | MCTRL-NACT<br>Actual speed<br>100 % = 16383    | MCTRL-PHI-ACT<br>Actual phase                  | MCTRL-MSET2<br>Torque setpoint<br>in %<br>100 % = 16383   |              |
| DF-slave<br>cascade               | 7003/7013/71<br>13           | MCTRL-NACT<br>Actual speed<br>100 % = 16383    | MCTRL-PHI-ACT<br>Actual phase                  | MCTRL-MSET2<br>Torque setpoint<br>in %<br>100 % = 16383   |              |
| Cam profiler                      | 1xx3                         | MCTRL-NACT<br>Actual speed<br>100 % = 16383    | not assigned                                   | not assigned                                              |              |
| Positioning                       | 2xx3                         | MCTRL-NACT<br>Actual speed<br>100 % = 16383    | not assigned                                   | not assigned                                              |              |
| vector control                    | 1xx3/4xx3/5xx<br>3/<br>10xx3 | MCTRL-NACT<br>Actual speed<br>100 % = 16383    | MCTRL-IACT                                     | MCTRL-NSET2<br>Speed controller<br>input<br>100 % = 16383 |              |
| vector control                    | 6xx3/7xx3/8xx<br>3/<br>9xx3  | MCTRL-NACT<br>Actual speed<br>100 % = 16383    | MCTRL-PHI-ANA                                  | MCTRL-MSET2<br>Torque setpoint<br>in %<br>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.

# 2175 (CANopen) fieldbus module

7

## Data transfer

7.6

## Process-data assignment for 93XX

7.6.6

### 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:<br>0 = Unit initialisation<br>1 = Switch-on inhibit<br>3 = Operation inhibited<br>4 = Flying-restart circuit active<br>5 = DC-injection brake active<br>6 = Operation enabled<br>7 = Message active<br>8 = Fault active<br>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              |

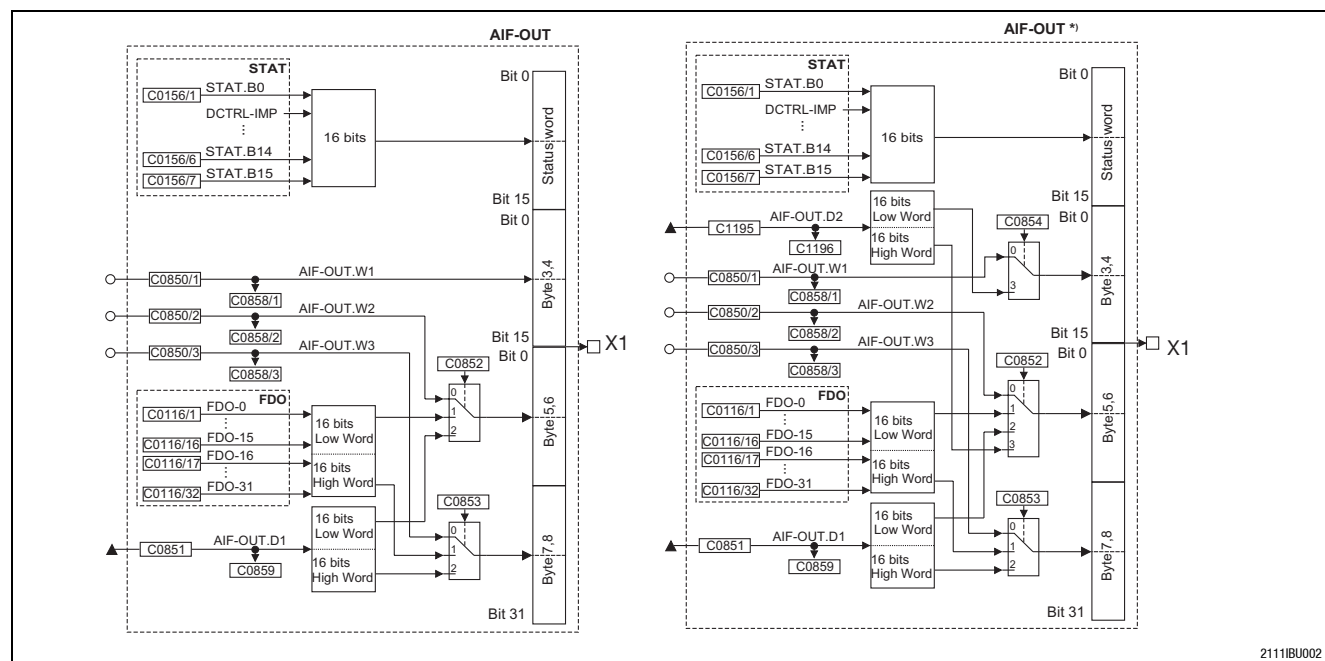


Fig. 7.6-9

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_nlnW1_a                          | AIF word 1        |
| AIF1_nlnW2_a                          | AIF word 2        |
| AIF1_nlnW3_a                          | AIF word 3        |
| AIF2_nlnW1_a                          | AIF word 4        |
| AIF2_nlnW2_a                          | AIF word 5        |
| AIF2_nlnW3_a                          | AIF word 6        |
| AIF2_nlnW4_a                          | AIF word 7        |
| AIF3_nlnW1_a                          | AIF word 8        |
| AIF3_nlnW2_a                          | AIF word 9        |
| AIF3_nlnW3_a                          | AIF word 10       |
| AIF3_nlnW4_a                          | AIF word 11       |
| AIF1_dlnD1_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.

# 2175 (CANopen) fieldbus module

7

## Data transfer

7.6

## Process data assignment for 9300 Servo PLC and Drive PLC

7.6.7

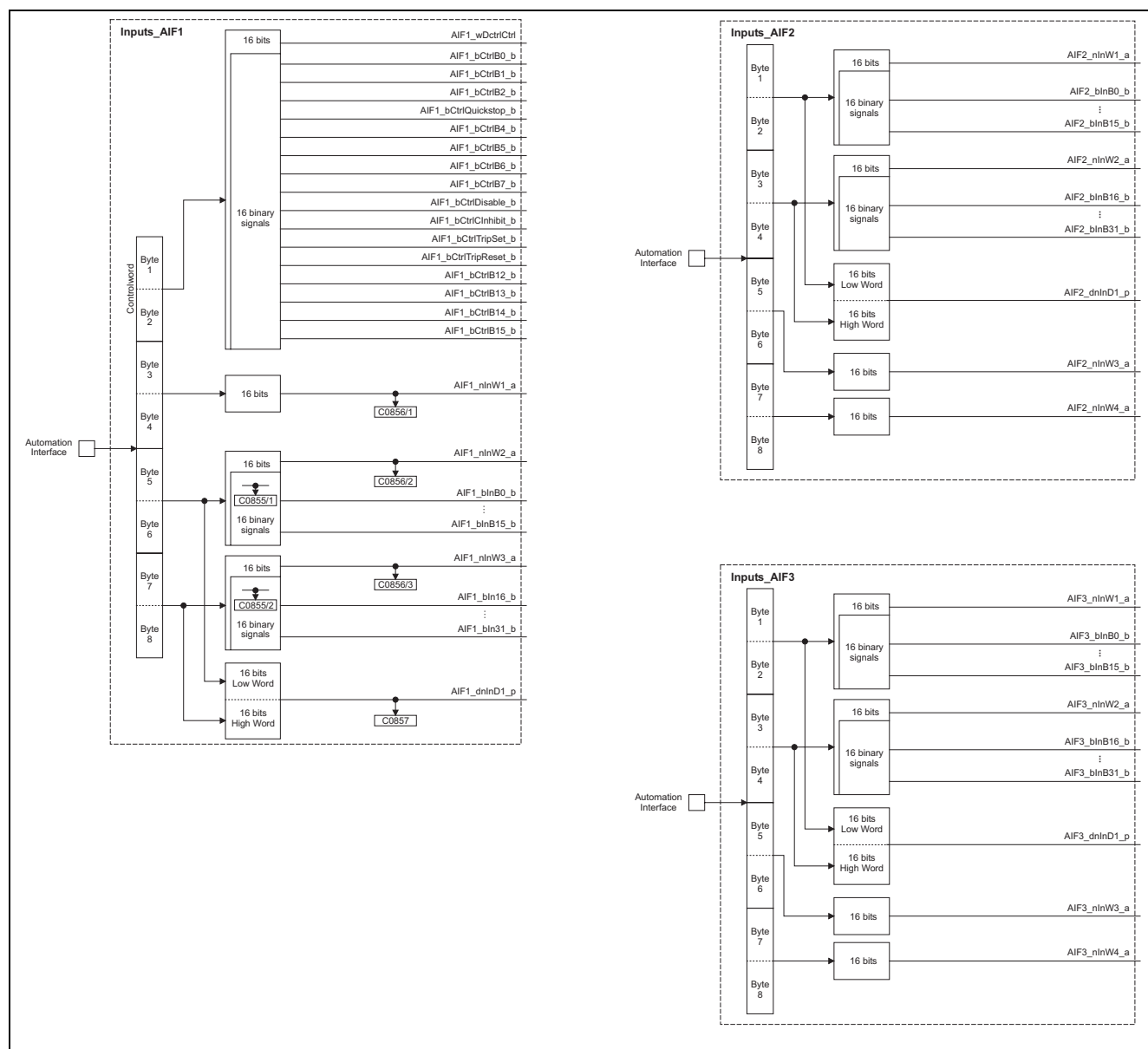


Fig. 7.6-10 Function blocks AIF-IN1, AIF-IN2 and AIF-IN3

# 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.

# 2175 (CANopen) fieldbus module

## Data transfer

### Process data assignment for 9300 Servo PLC and Drive PLC

7

7.6

7.6.7

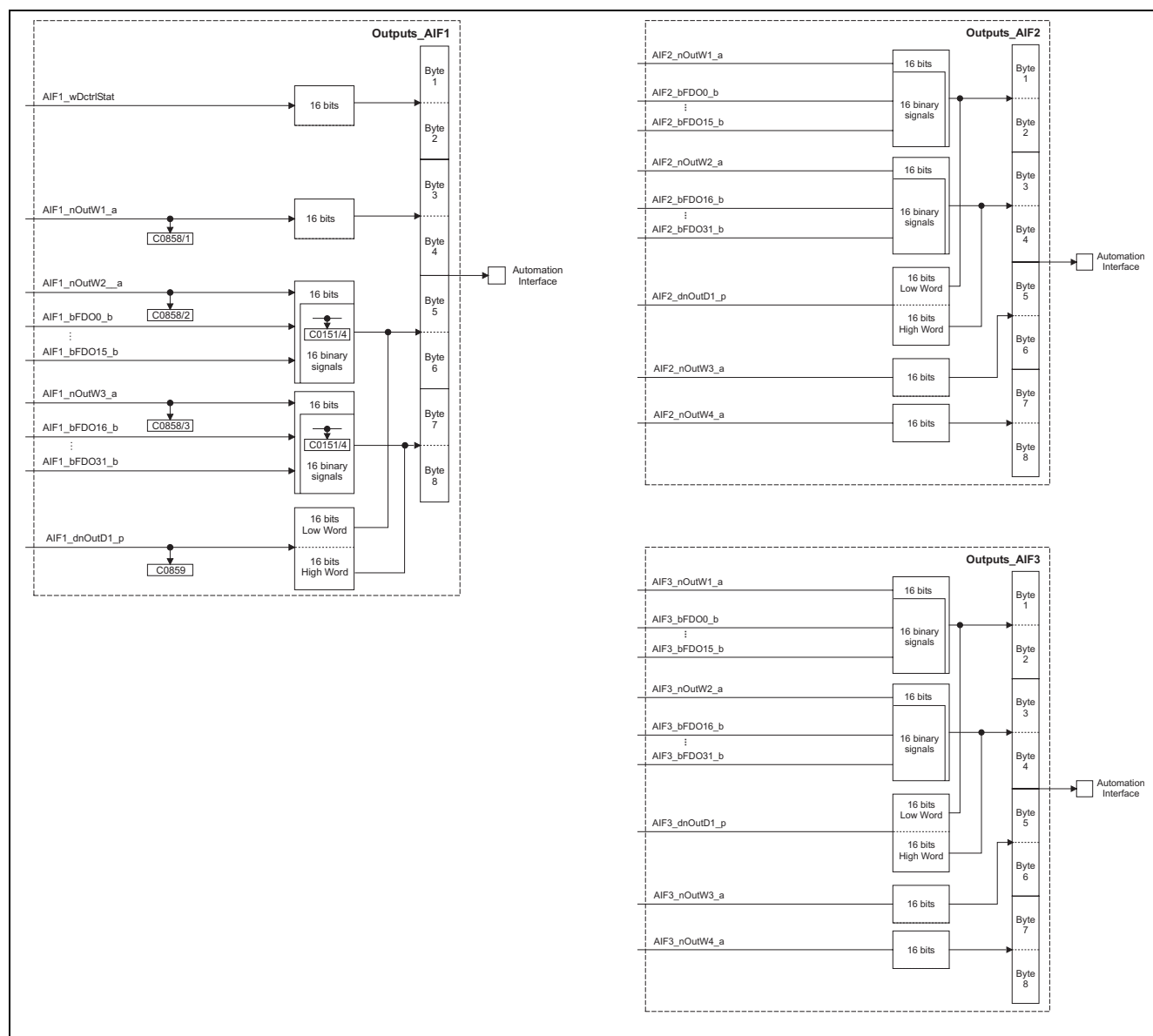


Fig. 7.6-11 Function blocks AIF-OUT1, AIF-OUT2 and AIF-OUT3

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<sub>hex</sub>) and 24575 (5FFF<sub>hex</sub>).

Conversion formula:

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

Example

| Lenze codes                                                                                                                      |  | dec                          | hex                                                                      |
|----------------------------------------------------------------------------------------------------------------------------------|--|------------------------------|--------------------------------------------------------------------------|
| <ul style="list-style-type: none"><li>• Addressing of Lenze codes via offset:<br/>– Example for operating mode L-C0001</li></ul> |  | Index = 24575 - LENZE CODENO | Index <sub>hex</sub> = 5FFF <sub>hex</sub> - LENZE-CODENO <sub>hex</sub> |
|                                                                                                                                  |  | Index = 24574 (= 24575 - 1)  | Index <sub>hex</sub> = 5FFE <sub>hex</sub> (= 5FFF <sub>hex</sub> - 1)   |

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

## 2175 (CANopen) fieldbus module

7

### Data transfer

7.6

### Parameter data channel

7.6.8

#### 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:

| 82XX                                                                                                                                                                                                                 | 8200 vector                                                                                                                                                                                                         | 93XX                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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:                                                          |                                                                                                                                                                                                                     | 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.<br><br>At first the parameter sets 2 - 4 must be activated, before the parameters included in them can be changed. |
| <ul style="list-style-type: none"><li>Offset 0 addresses parameter set 1 with the Lenze codes L-C0000 to L-C1999</li><li>Offset 2000 addresses parameter set 2 with the Lenze codes L-C2000 to L-C3999</li></ul>     |                                                                                                                                                                                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| no additional parameter sets available.                                                                                                                                                                              | <ul style="list-style-type: none"><li>Offset 4000 addresses parameter set 3 with the Lenze codes L-C4000 to L-C5999</li><li>Offset 6000 addresses parameter set 4 with the Lenze codes L-C6000 to L-C7999</li></ul> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| If a parameter is available only once (see the Operating Instructions for 82XX or 8200 vector), use the code digit offset 0.                                                                                         |                                                                                                                                                                                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Example for L-C0011 (maximum field frequency):<br>L-C0011 in parameter set 1: Lenze code = 11<br>L-C0011 in parameter set 2: Lenze code = 2011                                                                       |                                                                                                                                                                                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| -                                                                                                                                                                                                                    | L-C0011 in parameter set 3: Lenze code = 4011<br>L-C0011 in parameter set 4: Lenze code = 6011                                                                                                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Parameter changes:<br>82XX: Automatic saving in the controller<br>8200 vector: Automatic saving is factory-set (can be switched off with L-C0003)<br>Process data changes:<br>82XX, 8200 vector: no automatic saving |                                                                                                                                                                                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |



**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 configure 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<br>Low byte | Index<br>High byte | Subindex | Data 1        | Data 2    | Data 3    | Data 4    |
|                          |                   |                    |          | 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<br>(5th - 8th byte) | 2 byte data<br>(5th + 6th byte) | 1 byte data<br>(5th byte) |                      |
|                                                                                | hex                             | hex                             | hex                       |                      |
| Write request<br>(Send parameters to drive)                                    | 23                              | 2B                              | 2F                        | Writing not possible |
| Write Response<br>(Controller response to the write request (acknowledgement)) | 60                              | 60                              | 60                        |                      |
| Read Request<br>(Request to read a parameter from the drive)                   | 40                              | 40                              | 40                        | 40                   |
| Read Response<br>(Response to the read request with an actual value)           | 43                              | 4B                              | 4F                        | 41                   |
| Error response<br>(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):<br>Index Low Byte = $F3_{hex}$<br>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:<br>$24575 - 12 - 2000 = 22563 = 5823_{hex}$ | According to the left-justified Intel data format the entries are (see chapter 5.6.8):<br>Index Low Byte = $23_{hex}$<br>Index High Byte = $58_{hex}$ |

# 2175 (CANopen) fieldbus module

7

## Data transfer

7.6

## Parameter data channel

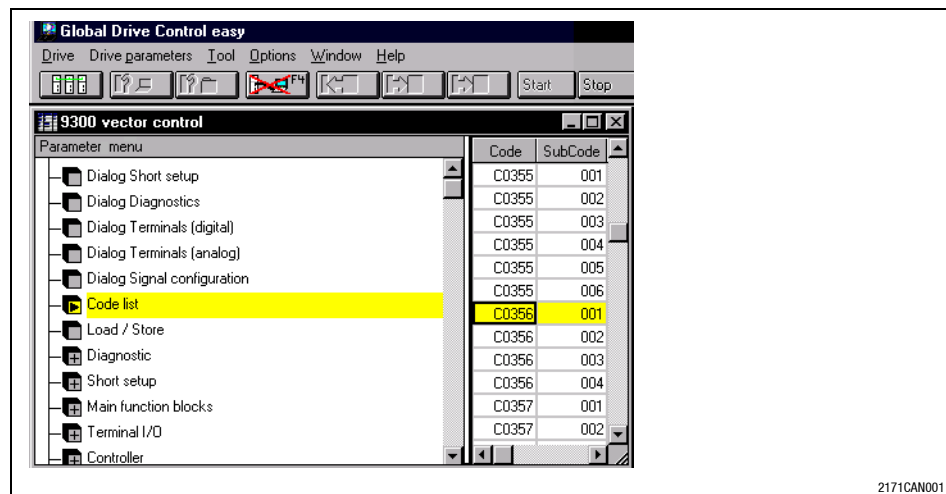
7.6.8

“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<sub>hex</sub> (1 - 4<sub>dec</sub>)



Data  
(Data 1 ... data 4)

|                                           |                                        |  |           |    |    |
|-------------------------------------------|----------------------------------------|--|-----------|----|----|
| Details about the parameter data telegram | Parameter value (length 1)             |  | 00        | 00 | 00 |
|                                           | Parameter value (length 2)<br>Low byte |  | High byte | 00 | 00 |
|                                           | Parameter value (length 4)<br>Low Word |  |           |    |    |

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<sub>dec</sub>.

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<sub>dec</sub> is to be considered.

## Fault messages

| Details about the parameter data telegram | Command | Index<br>Low byte | Index<br>High byte | Subindex | error<br>code |
|-------------------------------------------|---------|-------------------|--------------------|----------|---------------|
|-------------------------------------------|---------|-------------------|--------------------|----------|---------------|

Byte 1:

In the **Command** byte shows in code **128<sub>dec</sub> or 80<sub>hex</sub>** that an error has occurred.

Byte 2, 3 and 4:

In the **Index byte** and **Subindex** the index and subindex of that code in which the error has occurred, 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<sub>hex</sub> 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) |

2175 (CANopen) fieldbus module

Data transfer  
Examples for parameter data telegrams

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 <sub>hex</sub> |
|---------|---------------------|

- Index calculation

|                             |                                                  |
|-----------------------------|--------------------------------------------------|
| Index = 24575 - code number | Index = 24575 - 61 = 24514 = 5FC2 <sub>hex</sub> |
|-----------------------------|--------------------------------------------------|

Telegram to drive:

| Identifier | Command           | Index Low Byte    | Index High Byte   | Subindex | Data 1 | Data 2 | Data 3 | Data4 |
|------------|-------------------|-------------------|-------------------|----------|--------|--------|--------|-------|
| 1541       | 40 <sub>hex</sub> | C2 <sub>hex</sub> | 5F <sub>hex</sub> | 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<sub>hex</sub>

Index of the read request

5FC2<sub>hex</sub>

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 <sub>hex</sub> | C2 <sub>hex</sub> | 5F <sub>hex</sub> | 00       | B0 <sub>hex</sub> | 8F <sub>hex</sub> | 06 <sub>hex</sub> | 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 <sub>hex</sub> |
|---------|---------------------|

- Index calculation

|                             |                                                  |
|-----------------------------|--------------------------------------------------|
| Index = 24575 - code number | Index = 24575 - 12 = 24563 = 5FF3 <sub>hex</sub> |
|-----------------------------|--------------------------------------------------|

- Subindex: 0
- Calculation of the acceleration time

|                         |                                                      |
|-------------------------|------------------------------------------------------|
| Acceleration-time value | 20 s · 10.000 = 200.000 = 00 03 0D 40 <sub>hex</sub> |
|-------------------------|------------------------------------------------------|

- Telegram to drive

| Identifier | Command           | Index Low Byte    | Index High Byte   | Subindex | Data 1            | Data 2            | Data 3            | Data4 |
|------------|-------------------|-------------------|-------------------|----------|-------------------|-------------------|-------------------|-------|
| 1537       | 23 <sub>hex</sub> | F3 <sub>hex</sub> | 5F <sub>hex</sub> | 00       | 40 <sub>hex</sub> | 0D <sub>hex</sub> | 03 <sub>hex</sub> | 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 <sub>hex</sub> | F3 <sub>hex</sub> | 5F <sub>hex</sub> | 00       | 00     | 00     | 00     | 00    |

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

Command = write response (controller response (acknowledgement)) = 60<sub>hex</sub>

# 2175 (CANopen) fieldbus module

7

7.6

7.6.9

## Data transfer

### Examples for parameter data telegrams

#### 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 alphanumeric 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<sub>hex</sub> or 41<sub>hex</sub>) 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 <sub>hex</sub> | 37 <sub>hex</sub> | 5F <sub>hex</sub> | 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<sub>hex</sub>

#### • Response including the block length (14 characters)

| 1st byte          | 2nd byte          | 3rd byte          | 4th byte | 5th byte          | 6th byte | 7th byte | 8th byte |
|-------------------|-------------------|-------------------|----------|-------------------|----------|----------|----------|
| 41 <sub>hex</sub> | 37 <sub>hex</sub> | 5F <sub>hex</sub> | 00       | 0E <sub>hex</sub> | 00       | 00       | 00       |

1st byte: 41 read response. The entry 41<sub>hex</sub> implies that it is a block telegram.

2nd/3rd byte: see above

5th byte: 0E (=14<sub>dec</sub>) 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 <sub>hex</sub> | 00       | 00       | 00       | 00       | 00       | 00       | 00       |

1st byte: 60<sub>hex</sub>

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<sub>hex</sub> (=0110 0000<sub>bin</sub>), then with command 70<sub>hex</sub> (=0111 0000<sub>bin</sub>), after this again with 60<sub>hex</sub> etc. The response is send accordingly. It is alternating because of a toggle bit. The process is completed by command 11<sub>hex</sub> (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 <sub>hex</sub> | 32 <sub>hex</sub> | 53 <sub>hex</sub> | 38 <sub>hex</sub> | 32 <sub>hex</sub> | 31 <sub>hex</sub> | 32 <sub>hex</sub> |

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 <sub>hex</sub> | 00       | 00       | 00       | 00       | 00       | 00       | 00       |

1st byte: 70<sub>hex</sub> (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 <sub>hex</sub> | 56 <sub>hex</sub> | 5F <sub>hex</sub> | 31 <sub>hex</sub> | 34 <sub>hex</sub> | 30 <sub>hex</sub> | 30 <sub>hex</sub> | 30 <sub>hex</sub> |

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)

## 2175 (CANopen) fieldbus module

7

### Data transfer

7.6

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

7.6.10

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<br>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.



# 2175 (CANopen) fieldbus module

*CANopen objects and Lenze codes*  
*Communication relevant Lenze codes*

## 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  $\geq$  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  $\geq$  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  $\neq$  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 <sub>d</sub> =<br>58ED <sub>h</sub> | -                 | -         | 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 <sub>d</sub> =<br>58EC <sub>h</sub> |                   |           | 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 <sub>h</sub> =<br>22725 <sub>d</sub> | 1                 | 1 [1] 6<br>3 | FIX32     |
| L-C2350 | -       | 56D1 <sub>h</sub> =<br>22225 <sub>d</sub> | 1                 | 1 [1] 6<br>3 | 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).

# 2175 (CANopen) fieldbus module

## CANopen objects and Lenze codes

### Description of communication relevant Lenze codes

L-C1851/L-C2351:  
baud rate

| Code    | Subcode | Index                                     | Possible settings |           | Data type |
|---------|---------|-------------------------------------------|-------------------|-----------|-----------|
|         |         |                                           | Lenze             | Selection |           |
| L-C1851 | -       | 58C4 <sub>h</sub> =<br>22724 <sub>d</sub> | 0                 | 0 .... 6  | FIX32     |
| L-C2351 | -       | 56D0 <sub>h</sub> =<br>22224 <sub>d</sub> |                   | 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 <sub>h</sub> =<br>22723 <sub>d</sub> | 0                 | 0 = Slave operation<br>1 = Master operation | FIX32     |
| L-C2352 | -       | 56CF <sub>h</sub> =<br>22223 <sub>d</sub> |                   |                                             |           |

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.

# 2175 (CANopen) fieldbus module

## CANopen objects and Lenze codes

### Description of communication relevant Lenze codes

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

| Code    | Subcode            | Index                                  | Possible settings |           | Data type | Explanation                                                                                                                                                                                         |
|---------|--------------------|----------------------------------------|-------------------|-----------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|         |                    |                                        | Lenze             | Selection |           |                                                                                                                                                                                                     |
| L-C1853 | /1<br>CAN-IN1/OUT1 | 58C2 <sub>h</sub> = 22722 <sub>d</sub> | 0                 | 0 [1] 3   | FIX32     | 0: Addressing to CANopen (Default identifier)<br>1: Addressing to L-C1854/<br>L-C2354<br>2 : Addressing to LENZE system bus<br>3 : Addressing to CANopen index 14Xx <sub>h</sub> /18Xx <sub>h</sub> |
|         | /2<br>CAN-IN2/OUT2 |                                        |                   |           |           |                                                                                                                                                                                                     |
| L-C2353 | /3<br>CAN-IN3/OUT3 | 56CE <sub>h</sub> = 22222 <sub>d</sub> |                   | 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<sub>h</sub>/18Xx<sub>h</sub> (see above: value 3), if the CANopen indices 1400<sub>h</sub>, 1401<sub>h</sub>, 1402<sub>h</sub>, 1800<sub>h</sub>, 1801<sub>h</sub> or 1802<sub>h</sub> are assigned with a new value.

## 2175 (CANopen) fieldbus module

### CANopen objects and Lenze codes

#### Description of communication relevant Lenze codes

##### Calculation of the identifiers

| PDO          | Value L-C1853 / L - C2353                  |                                                   |                                 |                       |                                      |
|--------------|--------------------------------------------|---------------------------------------------------|---------------------------------|-----------------------|--------------------------------------|
|              | 0<br>to CANopen<br>(default<br>identifier) | 1<br>via code L-1854/L-2354<br>Default<br>setting | 2<br>via<br>Lenze system<br>bus | 3<br>to CANopen index |                                      |
| CAN-IN1      | 512 + node<br>address                      | 384 + C1854/1<br>or<br>384 + C2354/1              | 384 + 129                       | 512 + node<br>address | Index 1400 <sub>h</sub> , subindex 1 |
| CAN-IN2      | 768 + node<br>address                      | 384 + C1854/3<br>or<br>384 + C2354/3              | 384 + 257                       | 640 + node<br>address | Index 1401 <sub>h</sub> , subindex 1 |
| CAN-IN3      | 1024 + node<br>address                     | 384 + C1854/5<br>or<br>384 + C2354/5              | 384 + 385                       | 768 + node<br>address | Index 1402 <sub>h</sub> , subindex 1 |
| CAN-<br>OUT1 | 384 + node<br>address                      | 384 + C1854/2<br>or<br>384 + C1854/2              | 384 + 1                         | 384 + node<br>address | Index 1800 <sub>h</sub> , subindex 1 |
| CAN-<br>OUT2 | 640 + node<br>address                      | 384 + C1854/4<br>or<br>384 + C1854/4              | 384 + 258                       | 641 + node<br>address | Index 1801 <sub>h</sub> , subindex 1 |
| CAN-<br>OUT3 | 896 + node<br>address                      | 384 + C1854/6<br>or<br>384 + C1854/6              | 384 + 386                       | 769 + node<br>address | Index 1802 <sub>h</sub> , 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<sub>hex</sub>) + 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<sub>hex</sub>/18XX<sub>hex</sub>

If the subcode has the value 3, this makes clear that the identifiers have been changed via the CANopen indices 14XX<sub>hex</sub>/18XX<sub>hex</sub>. Now, the identifier is developed from the CANopen indices.

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

## 2175 (CANopen) fieldbus module

7

### CANopen objects and Lenze codes

7.7

### Description of communication relevant Lenze codes

7.7.2

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

| Code    | Subcode      | Index                                     | Possible settings |           | Data type |
|---------|--------------|-------------------------------------------|-------------------|-----------|-----------|
|         |              |                                           | Lenze             | Selection |           |
| L-C1854 | /1 CAN-IN1   | 58C1 <sub>h</sub> =<br>22721 <sub>d</sub> | /1:               | 0         | FIX32     |
|         | /2 CAN-OUT1  |                                           | 129               | [1]       |           |
|         | /3* CAN-IN2  |                                           | /2: 1             |           |           |
|         | /4* CAN-OUT2 |                                           | /3:               |           |           |
|         | /5* CAN-IN3  |                                           | 257*              |           |           |
| L-C2354 | /6* CAN-OUT3 | 56CD <sub>h</sub> =<br>22221 <sub>d</sub> | /4:               | 0         | 513       |
|         |              |                                           | 258*              | [1]       |           |
|         |              |                                           | /5:               |           |           |
|         |              |                                           | 385*              |           |           |
|         |              |                                           | /6:               |           |           |
|         |              |                                           | 386*              |           |           |

\*) 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   | 58C0 <sub>h</sub> =<br>22720 <sub>d</sub> | -                 | 0         | [1] 2047  | FIX32 |
|         | /2 CAN-OUT1  |                                           |                   |           |           |       |
|         | /3* CAN-IN2  |                                           |                   |           |           |       |
| L-C2355 | /4* CAN-OUT2 | 56CC <sub>h</sub> =<br>22220 <sub>d</sub> |                   |           |           |       |
|         | /5* CAN-IN3  |                                           |                   |           |           |       |
|         | /6* CAN-OUT3 |                                           |                   |           |           |       |

\*) 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*

## 2175 (CANopen) fieldbus module

### CANopen objects and Lenze codes

#### Description of communication relevant Lenze codes

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

| Code    | Subcode                 | Index                                     | Possible settings |                | Data type |
|---------|-------------------------|-------------------------------------------|-------------------|----------------|-----------|
|         |                         |                                           | Lenze             | Selection      |           |
| L-C1856 | /1 Boot up time         | 58BF <sub>h</sub> =<br>22719 <sub>d</sub> | 1:                | 0 [1 ms] 65535 | FIX32     |
|         | /2 cycle times CAN-OUT1 |                                           | 3000 ms           |                |           |
|         | /3 cycle times CAN-OUT2 |                                           | 2 .. 5: 0 ms      |                |           |
| L-C2356 | /4 cycle times CAN-OUT3 | 56CB <sub>h</sub> =<br>22219 <sub>d</sub> |                   |                | FIX32     |
|         | /5 Sync-Tx cycle times  |                                           |                   |                |           |

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<sub>h</sub> “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<sub>h</sub> is rounded to an integral multiple of 1000 μs. If the CANopen index 1006<sub>h</sub> is read, the contents of this code in [μs] will be sent as response.

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

## 2175 (CANopen) fieldbus module

7

### CANopen objects and Lenze codes

7.7

### Description of communication relevant Lenze codes

7.7.2

L-C1857/L-C2357:  
Monitoring time

| Code    | Subcode                                     | Index                                     | Possible settings |           |              | Data type |
|---------|---------------------------------------------|-------------------------------------------|-------------------|-----------|--------------|-----------|
|         |                                             |                                           | Lenze             | Selection |              |           |
| L-C1857 | /1 CAN-IN1<br>/2 CAN-IN2                    | 58BE <sub>h</sub> =<br>22718 <sub>d</sub> | 3000<br>ms        | 0         | [1 ms] 65535 | FIX32     |
| L-C2357 | /3 CAN-IN3<br>/4 BUS-OFF<br>monitoring time | 56CA <sub>h</sub> =<br>22218 <sub>d</sub> |                   |           |              |           |



#### 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 <sub>h</sub> =<br>22716 <sub>d</sub> | -                 | 0 [1] 1023 | U16       |
| L-C2359 | -       | 56C8 <sub>h</sub> =<br>22216 <sub>d</sub> |                   | 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<sub>h</sub> (97<sub>d</sub>), which is displayed when reading the code L-C1859 or L-C2359.

## 2175 (CANopen) fieldbus module

### CANopen objects and Lenze codes

#### Description of communication relevant Lenze codes

**L-C1860:**  
Display of the current DIP switch position

| Code    | Subcode | Index                                     | Possible settings |            | Data type |
|---------|---------|-------------------------------------------|-------------------|------------|-----------|
|         |         |                                           | Lenze             | Selection  |           |
| L-C1860 | -       | 58BB <sub>h</sub> =<br>22715 <sub>d</sub> | -                 | 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 <sub>h</sub> =<br>22708 <sub>d</sub> | 128               | 0 [1] 2047 | FIX32     |
| L-C2367 | -       | 56C0 <sub>h</sub> =<br>22208 <sub>d</sub> |                   |            | 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<sub>h</sub> "COB-ID SYNC message" directly influences this code. The identifier of a value re-entered into the index 1005<sub>h</sub> will also be taken over from the code L-C1867/L-C2367.

When reading the CANopen index 1005<sub>hex</sub> (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 <sub>h</sub> =<br>22707 <sub>d</sub> | 128               | 0 [1] 2047 | FIX32     |
| L-C2368 | -       | 56BF <sub>h</sub> =<br>22207 <sub>d</sub> |                   |            | 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<sub>h</sub> “COB-ID SYNC message” directly influences this code. The identifier of a value that has been reentered into the index 1005<sub>h</sub> will also be taken over from the code L-C1868/L-C2368. When reading the index 1005<sub>h</sub> the value saved here will be displayed.

## 2175 (CANopen) fieldbus module

### CANopen objects and Lenze codes

#### Description of communication relevant Lenze codes

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<br>/2* CAN-IN2 | 58AE <sub>h</sub> =<br>22702 <sub>d</sub> | 1                 | 0 [1]     | 240 | FIX32     |
| L-C2373 | /3* CAN-IN3               | 56BA <sub>h</sub> =<br>22202 <sub>d</sub> |                   | 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<sub>h</sub>, 1401<sub>h</sub> and 1402<sub>h</sub> (receive PDO communication parameter), subindex 2 each (transmission type), are directly mapped on the subcodes of code L-C1873 / L-C2373.

|                                              |   |                                 |
|----------------------------------------------|---|---------------------------------|
| Index 14XX <sub>h</sub> , subindex 2 = 1-240 | = | Code L-C1873, subcode X = 1-240 |
| Index 14XX <sub>h</sub> , subindex 2 = 254   |   | Code L-C1873, subcode 0         |

An exception is the value 0, which is not directly mapped on the CANopen indices 1400<sub>hex</sub>, 1401<sub>hex</sub> and 1402<sub>hex</sub>. The value 0 is mapped under CANopen index 14XX<sub>hex</sub>, 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.

2175 (CANopen) fieldbus module

CANopen objects and Lenze codes

Description of communication relevant Lenze codes

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

| Code    | Subcode                         | Index                                     | Possible settings |           | Data type    |
|---------|---------------------------------|-------------------------------------------|-------------------|-----------|--------------|
|         |                                 |                                           | Lenze             | Selection |              |
| L-C1874 | /1 = CAN-OUT1<br>/2* = CAN-OUT2 | 58AD <sub>h</sub> =<br>22701 <sub>d</sub> | 1                 | 1 [1]     | 240<br>FIX32 |
| L-C2374 | /3* = CAN-OUT3                  | 56B9 <sub>h</sub> =<br>22201 <sub>d</sub> |                   |           | 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<sub>h</sub>, 1801<sub>h</sub> and 1802<sub>h</sub> (with subindex 2, “transmission type”) are directly mapped on the subcodes of code L-C1874 or L-C2374.

|                                              |   |                                            |
|----------------------------------------------|---|--------------------------------------------|
| Index 18XX <sub>h</sub> , subindex 2 = 1-240 | = | Code L-C1874 or L-C2374, subcode X = 1-240 |
| Index 18XX <sub>h</sub> , subindex 2 =252    |   | Code L-C1875, subcode X = 0                |
| Index 18XX <sub>h</sub> , subindex 2 = 254   |   | Code L-C1875, subcode X = 1                |
|                                              |   | Code L-C1875, subcode X = 2                |

If for instance the index 1802<sub>h</sub> is read, the value in code L-C1874 / L-C2374, subcode 3 comes back as an answer. If the index 1801<sub>h</sub> 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                  | 58AC <sub>h</sub> =<br>22700 <sub>d</sub> | /1: 0<br>/2: 1<br>/3: 1 | 0 [1]     | 3 FIX32   |
| L-C2375 | /2* CAN-OUT2<br>/3* CAN-OUT3 | 56B8 <sub>h</sub> =<br>22200 <sub>d</sub> | /1: 0<br>/2: 0<br>/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.

# 2175 (CANopen) fieldbus module

## CANopen objects and Lenze codes Description of communication relevant Lenze codes



**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<sub>hex</sub>, 1801<sub>hex</sub> or 1802<sub>hex</sub>, subindex 2 each.

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

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

When describing CANopen index 1800<sub>hex</sub>, 1801<sub>hex</sub> or 1802<sub>hex</sub> the same relation prevails, see the table below or under the description of the CANopen indices 18XX<sub>hex</sub>.

| 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                                                           |

## 2175 (CANopen) fieldbus module

### CANopen objects and Lenze codes

#### Description of communication relevant Lenze codes

#### 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 | 58AB <sub>h</sub> = | 65535             | 0 [1]     | FIX32     |
|         | /2 CAN-OUT1.W2 | 22699 <sub>d</sub>  |                   |           |           |
| L-C2376 | /3 CAN-OUT1.W3 | 56B7 <sub>h</sub> = |                   |           | FIX32     |
|         | /4 CAN-OUT1.W4 | 22199 <sub>d</sub>  |                   |           |           |

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<sub>hex</sub> (see "MASK"). Please note the field marked in grey.

1. cycle

Result after 1. cycle: The PDO is transmitted

|               | CAN-OUT 1.W3 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---------------|--------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| <b>MASK</b>   | 0            | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| <b>Data</b>   | 1            | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| <b>Result</b> | 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 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---------------|--------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| <b>MASK</b>   | 0            | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| <b>Data</b>   | 1            | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| <b>Result</b> | 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.

2175 (CANopen) fieldbus module

CANopen objects and Lenze codes

Description of communication relevant Lenze codes

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 | 58AA <sub>h</sub> = | 65535             | 0 [1] 65535 | FIX32     |
|         | /2 CAN-OUT2.W2 | 22698 <sub>d</sub>  |                   |             |           |
| L-C2377 | /3 CAN-OUT2.W3 | 56B6 <sub>h</sub> = |                   |             |           |
|         | /4 CAN-OUT2.W4 | 22198 <sub>d</sub>  |                   |             |           |

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 | 58A9 <sub>h</sub> = | 65535             | 0 [1] 65535 | FIX32     |
|         | /2 CAN-OUT3.W2 | 22697 <sub>d</sub>  |                   |             |           |
| L-C2378 | /3 CAN-OUT3.W3 | 56B5 <sub>h</sub> = |                   |             |           |
|         | /4 CAN-OUT3.W4 | 22197 <sub>d</sub>  |                   |             |           |

See L-C1876 / L-C2376.

2175 (CANopen) fieldbus module

CANopen objects and Lenze codes

Description of communication relevant Lenze codes

L-C1882/L-C2382:  
Monitoring reaction

| Code    | Subcode                                                           | Index                                  | Possible settings |                                                              | Data type |
|---------|-------------------------------------------------------------------|----------------------------------------|-------------------|--------------------------------------------------------------|-----------|
|         |                                                                   |                                        | Lenze             | Selection                                                    |           |
| L-C1882 | /1 response CAN-IN1<br>/2 response CAN-IN2<br>/3 response CAN-IN3 | 58A5 <sub>h</sub> = 22693 <sub>d</sub> | 0                 | 0<br>0: no response<br>1: Controller inhibit<br>2: Quickstop | 2 FIX32   |
| L-C2382 | /4 response BUS-OFF<br>/5 response Life Guarding Event            | 56B1 <sub>h</sub> = 22193 <sub>d</sub> |                   |                                                              | 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<sub>h</sub> "guard time" and 100D<sub>h</sub> "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<sub>h</sub> "guard time" it is possible to enter a time in milliseconds. Under index 100D<sub>h</sub> "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.

## 2175 (CANopen) fieldbus module

7

### CANopen objects and Lenze codes

7.7

### Description of communication relevant Lenze codes

7.7.2

L-C2120:  
AIF control byte

| Code    | Subcode | Index                                     | Possible settings |                                                                                                                                                                                                                                                               | Data type |
|---------|---------|-------------------------------------------|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
|         |         |                                           | Lenze             | Selection                                                                                                                                                                                                                                                     |           |
| L-C2120 | -       | 22455 <sub>d</sub> =<br>57B7 <sub>h</sub> | 0                 | 0 = No command<br>1 = Update codes L-23XX and CAN re-initialisation $\equiv$ Reset node<br>2 = Update codes L-C23XX<br>10 = L-C2356/1...4 re-read<br>11 = L-C2357 re-read<br>12 = L-C2375 re-read<br>13 = L-C2376 ... L-C2378 re-read<br>14 = L-C2382 re-read | FIX32     |

With 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 PDO                                                               |
| 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 <sub>d</sub> =<br>57B6 <sub>h</sub> | 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<sub>hex</sub>:  
Device type

The CANopen index 1000<sub>hex</sub> 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<sub>hex</sub> (2175IB).

| Index [hex] | Subindex | Name        | Data type | Value range                 | Rights |
|-------------|----------|-------------|-----------|-----------------------------|--------|
| 1000        | 0        | Device type | U32       | 0 ... (2 <sup>32</sup> - 1) | ro     |

Bit assignment in the telegram data

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

1001<sub>hex</sub>:  
Error Register

Reading the fault 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<br>1: device TRIP/ fault CANopen module<br>Fault analysis: see L-C0161 in the Instructions of the corresponding controller |

1003<sub>hex</sub>:  
Pre-defined Error Field

Fault history

| 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 <sup>32</sup> - 1) | ro     |

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.

# 2175 (CANopen) fieldbus module

7

7.7

7.7.3

## CANopen objects and Lenze codes

### Implemented CANopen objects

1005<sub>hex</sub>:  
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 <sup>32</sup> - 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<sub>hex</sub>).

#### Identifier description

For the receipt of PDOs the value 80<sub>hex</sub> 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<sub>hex</sub>. 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 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     | The device does not create any SYNC telegrams                                                       |
|            | 1     | The device creates SYNC telegrams                                                                   |
| 31         | X     | optional                                                                                            |

1006<sub>hex</sub>:  
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 <sup>32</sup> - 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<sub>hex</sub>:  
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<sub>hex</sub>:  
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<sub>hex</sub>:  
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.

2175 (CANopen) fieldbus module

CANopen objects and Lenze codes  
Implemented CANopen objects

100D<sub>hex</sub>:  
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<sub>hex</sub>:  
Store Parameters

Storage of parameters in the EEPROM.

| Index [hex] | Subindex | Name             | Data type | Value range                 | Rights |
|-------------|----------|------------------|-----------|-----------------------------|--------|
| 1010        | 0        | store parameters | U32       | 0 ... (2 <sup>32</sup> - 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<sub>hex</sub>
- false signature: 0800 0020<sub>hex</sub>



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 |

| Subindex | Rights | Explanation                                                                                                                                                                                     |                                                                                                                                                                                                                                                                                                           |
|----------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|          |        | Writing                                                                                                                                                                                         | Reading                                                                                                                                                                                                                                                                                                   |
| 0        | ro     | <ul style="list-style-type: none"><li>• If you attempt to write the following fault message occurs: 0601 0002</li></ul>                                                                         | Supported subindex = 3                                                                                                                                                                                                                                                                                    |
| 1        | rw     | <div></div> <ul style="list-style-type: none"><li>• This function is not yet supported at the moment.</li><li>• If you attempt to write the following fault message occurs: 0800 0020</li></ul> | Read memory functionality of all parameters                                                                                                                                                                                                                                                               |
| 2        |        |                                                                                                                                                                                                 | The following functionalities are possible in dependence on the controller and are described by reading the values of the bit positions 0 and 1:<br>Value 0: Storing is not carried out<br>Value 1: Storing on command<br>Value 2: Automatic storing<br>Value 3: Automatic storing and storing on command |
| 3        |        |                                                                                                                                                                                                 | Only read memory functionality of the communication parameters of the objects.<br>Only read memory functionality of the vendor-specific parameters<br>(Memory area: 6000 <sub>hex</sub> - 9FFF <sub>hex</sub> )                                                                                           |

# 2175 (CANopen) fieldbus module

CANopen objects and Lenze codes  
Implemented CANopen objects

1011<sub>hex</sub>:  
Restore Default Parameters

Loading of the default setting.



**Note!**

If this function is applied, the subindices used depend on the controller type.

| Index <small>[hex]</small> | Subindex | Name                       | Data type | Value range                 | Rights |
|----------------------------|----------|----------------------------|-----------|-----------------------------|--------|
| 1011                       | 0 ... 7  | restore default parameters | U32       | 0 ... (2 <sup>32</sup> - 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   |  |  |  |
| hex              | 64  |  |  |  | 61  |  |  |  |
| hex              | 64  |  |  |  | 6F  |  |  |  |
| hex              | 6C  |  |  |  | 6C  |  |  |  |

Bit assignment for write authorisation

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

| Subindex | Rights | Writing                                                                                                                                                                                                                                                                                          | Reading                                                                                                   |
|----------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|
| 0        | ro     | <ul style="list-style-type: none"><li>If you attempt to write the following fault message occurs:<br/>0601 0002<sub>hex</sub></li></ul>                                                                                                                                                          | Maximally available subindex dependent on the controller type:<br>7: 8200 vector/motec frequency inverter |
| 1        | rw     | <div></div> <ul style="list-style-type: none"><li>This function is not yet supported at the moment.</li><li>If you attempt to write the following fault message occurs:<br/>0800 0020<sub>hex</sub></li></ul> | 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 <sub>hex</sub> - 9FFF <sub>hex</sub> )         |
| 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                                                                       |

## 2175 (CANopen) fieldbus module

### CANopen objects and Lenze codes

#### Implemented CANopen objects

**1014<sub>hex</sub>:**  
**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 <sub>[hex]</sub> | Subindex | Name                    | Data type | Value range                 | Rights |
|------------------------|----------|-------------------------|-----------|-----------------------------|--------|
| 1014                   | 0        | COB-ID emergency object | U32       | 0 ... (2 <sup>32</sup> - 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 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<br>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<br>LENZE: Error code “10XX” |          | Error register<br>Object<br>1001 <sub>hex</sub> | Field for vendor-specific fault message |          |          |          |          |

**1015<sub>hex</sub>:**  
**Inhibit time emergency**

This object serves to select the interval between the occurrence of an internal fault and the sending of the fault message to the bus (“COB-ID Emergency object”, code: 1014<sub>hex</sub>).

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 <sub>[hex]</sub> | Subindex | Name                   | Data type | Value range | Rights |
|------------------------|----------|------------------------|-----------|-------------|--------|
| 1015                   | 0        | Inhibit time emergency | U16       | 0 ... 65535 | rw     |

#### Entry of the vendor ID

**1018<sub>hex</sub>:**  
**Identity object**

| Index <sub>[hex]</sub> | 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    |

2175 (CANopen) fieldbus module

CANopen objects and Lenze codes  
Implemented CANopen objects

1200<sub>hex</sub> / 1201<sub>hex</sub>:  
Server SDO parameters

Two objects (CAN parameter data channel 1 = 1200<sub>hex</sub> and CAN parameter data channel 2 = 1201<sub>hex</sub>) 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 (subindex 1 and 2).

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

(resulting) identifier = basic identifier + controller address = 1540<sub>dec</sub> = 0604<sub>hex</sub>

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<sub>hex</sub> + 604<sub>hex</sub> = 80 00 06 04<sub>hex</sub> (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 <sub>hex</sub> | 06 <sub>hex</sub> | 0000 0000 | 80 <sub>hex</sub> |



2175 (CANopen) fieldbus module

CANopen objects and Lenze codes

Implemented CANopen objects

1401<sub>hex</sub>:  
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 <sub>[hex]</sub> | 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 <sub>hex</sub> + 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<sub>hex</sub>.

1402<sub>hex</sub>:  
Receive PDO3\* Communication  
Parameter



**Note!**

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

Receipt of communication parameters of PDO 3

| Index <sub>[hex]</sub> | 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 <sub>hex</sub> + 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<sub>hex</sub>.

## 2175 (CANopen) fieldbus module

### CANopen objects and Lenze codes

#### Implemented CANopen objects

**1600<sub>hex</sub>:**  
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<sub>hex</sub>

Subind. 1 - 4: value 10<sub>hex</sub>

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

**1601<sub>hex</sub>:**  
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<sub>hex</sub>

Subind. 1 - 4: value 10<sub>hex</sub>

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

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

2175 (CANopen) fieldbus module

CANopen objects and Lenze codes

Implemented CANopen objects

1602<sub>hex</sub>:  
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<sub>hex</sub>  
Subind. 1 - 4: value 10<sub>hex</sub>  
\*) The object is not available for the 82XX, 8200 vector/motec and 93XX controllers.

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

1800<sub>hex</sub>:  
Transmit PDO1 Communication  
Parameter

Process data transmission

| Index <sub>[hex]</sub> | 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 <sub>hex</sub> + 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<br>1 | RTR to this PDO is permitted (Lenze)<br>RTR to this PDO is not permitted (not adjustable)<br>RTR = remote transmission request |
| 31 (MSB)     | 0<br>1 | 0: PDO active<br>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.

# 2175 (CANopen) fieldbus module

## CANopen objects and Lenze codes Implemented CANopen objects

1801<sub>hex</sub>:  
Transmit PDO2\* Communication  
Parameter

Process data transmission



**Note!**

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

| Index <small>[hex]</small> | 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 <sub>hex</sub> + 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<br>1 | RTR to this PDO is permitted (Lenze)<br>RTR to this PDO is not permitted (not adjustable)<br>RTR = remote transmission request |
| 31 (MSB)     | 0<br>1 | 0: PDO active<br>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<sub>hex</sub>:  
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 <sub>hex</sub> + 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<br>1 | RTR to this PDO is permitted (Lenze)<br>RTR to this PDO is not permitted (not adjustable)<br>RTR = remote transmission request |
| 31 (MSB)     | 0<br>1 | 0: PDO active<br>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                           |

2175 (CANopen) fieldbus module

CANopen objects and Lenze codes

Implemented CANopen objects

1A00<sub>hex</sub>:  
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<sub>hex</sub> (max. supported subindex)  
Subind. 1 - 4: In case of read requests regarding this object the value 10<sub>hex</sub> will be sent back.

| Index <sub>[hex]</sub> | 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<sub>hex</sub>:  
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<sub>hex</sub> (max. supported subindex)  
Subind. 1 - 4: In case of read requests regarding this object the value 10<sub>hex</sub> will be sent back.  
\*) The object is not available for the 82XX, 8200 vector/motec and 93XX controllers.

| Index <sub>[hex]</sub> | 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<sub>hex</sub>:  
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<sub>hex</sub> (max. supported subindex)

Subind. 1 - 4: In case of read requests regarding this object the value 10<sub>hex</sub> will be sent back.

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

| Index <sub>[hex]</sub> | 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 |        |

2175 (CANopen) fieldbus module

Troubleshooting

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<br>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<br>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.<br>A voltage of 24 V +10 % must be applied.                        |
|                                               |                                                                                                                           | The fieldbus module has not been initialised with the controller yet.<br>Possibility 1: Controller not switched on (see fault possibility 1).<br>Possibility 2: Check the connection to the controller |
| Does the controller receive telegrams?        | The LED "Status bus connection" at the fieldbus module<br>7.4-1 must blink green when communicating with host.            | Check wether the connection corresponds to the instrucion given in chapter "electrical installation".<br>Check whether host sends telegrams and uses the appropriate interface.                        |
|                                               |                                                                                                                           | Has the available device address already been assigned?<br>Check the setting of the other participants in the DeviceNet.                                                                               |



# 2175 (CANopen) fieldbus module

Appendix  
Code table

## 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            | Subcode 1 of code C1853                                                                                                                       |
|           | 2 .. 4       | 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<br>FIX32: Fixed value 4 bytes (= 32 bits)<br>U16: Unsigned Integer 2 bytes (= 16 bits) |

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

| Code    | Subcode               | Index                                  | Possible settings                                 |                                                                                                                                                                              | Data type | Name                                 |
|---------|-----------------------|----------------------------------------|---------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|--------------------------------------|
|         |                       |                                        | Lenze                                             | Selection                                                                                                                                                                    |           |                                      |
| L-C1810 | -                     | 22765 <sub>d</sub> = 58ED <sub>h</sub> | -                                                 | -                                                                                                                                                                            | VS        | Software product code                |
| L-C1811 | -                     | 22764 <sub>d</sub> = 58EC <sub>h</sub> |                                                   |                                                                                                                                                                              | VS        | Software creation date               |
| L-C1850 | -                     | 58C5 <sub>h</sub> = 22725 <sub>d</sub> | 1                                                 | 1 [1] 63                                                                                                                                                                     | FIX32     | Node address                         |
| L-C1851 | -                     | 58C4 <sub>h</sub> = 22724 <sub>d</sub> | 0                                                 | 0 = 500 kbits/s, 1 = 250 kbits/s<br>2 = 125 kbits/s, 3 = 50 kbits/s<br>4 = 1000 kbits/s, 5 = 20 kbits/s<br>6 = 10 kbits/s                                                    | FIX32     | baud rate                            |
| L-C1852 | -                     | 58C3 <sub>h</sub> = 22723 <sub>d</sub> | 0                                                 | 0 = Slave operation<br>1 = Master operation                                                                                                                                  | FIX32     | Master/slave operation               |
| L-C1853 | /1 ... /3             | 58C2 <sub>h</sub> = 22722 <sub>d</sub> | 0                                                 | 0 = Addressing to CANopen<br>1 = Addressing to L-C1854/L-C2354<br>2 = Addressing to LENZE system bus<br>3 = Addressing to CANopen index 14Xx <sub>h</sub> /18XX <sub>h</sub> | FIX32     | Addressing CAN-INx/ CAN-OUTx         |
| L-C1854 | /1 ... /2 /3* ... /6* | 58C1 <sub>h</sub> = 22721 <sub>d</sub> | /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 <sub>h</sub> = 22720 <sub>d</sub> | 0                                                 | 0 [1] 2047                                                                                                                                                                   | FIX32     | Display of resulting identifiers     |
| L-C1856 | /1 ... /5             | 58BF <sub>h</sub> = 22719 <sub>d</sub> | /1: 3000 ms /2 ... /5: 0 ms                       | 0 35 [1 ms] 655                                                                                                                                                              | FIX32     | Boot up and cycle times              |
| L-C1857 | /1 ... /4             | 58BE <sub>h</sub> = 22718 <sub>d</sub> | 3000 ms                                           | 0 35 [1 ms] 655                                                                                                                                                              | FIX32     | Monitoring time                      |
| L-C1859 | -                     | 58BC <sub>h</sub> = 22716 <sub>d</sub> | -                                                 | 0 [1] 1023                                                                                                                                                                   | U16       | Display of DIP switch position       |

## 2175 (CANopen) fieldbus module

Appendix  
Code table

| Code    | Subcode        | Index                                     | Possible settings         |                                                                    | Data type | Name                                       |
|---------|----------------|-------------------------------------------|---------------------------|--------------------------------------------------------------------|-----------|--------------------------------------------|
|         |                |                                           | Lenze                     | Selection                                                          |           |                                            |
| L-C1860 | -              | 58BB <sub>h</sub> =<br>22715 <sub>d</sub> | -                         | 0 [1] 1023                                                         | U16       | Display of the current DIP switch position |
| L-C1867 | -              | 58B4 <sub>h</sub> =<br>22708 <sub>d</sub> | 128                       | 0 [1] 2047                                                         | FIX32     | Sync Rx identifier                         |
| L-C1868 | -              | 58B3 <sub>h</sub> =<br>22707 <sub>d</sub> | 128                       | 0 [1] 2047                                                         | FIX32     | Sync Tx identifier                         |
| L-C1873 | /1<br>/2*, /3* | 58AE <sub>h</sub> =<br>22702 <sub>d</sub> | 1                         | 0 [1] 240                                                          | FIX32     | Sync rate CAN-IN1 ... CAN-IN3              |
| L-C1874 | /1<br>/2*, /3* | 58AD <sub>h</sub> =<br>22701 <sub>d</sub> | 1                         | 1 [1] 240                                                          | FIX32     | Sync rate CAN-OUT1 ... CAN-OUT3            |
| L-C1875 | /1<br>/2*, /3* | 58AC <sub>h</sub> =<br>22700 <sub>d</sub> | /1: 0<br>/2: 1*<br>/3: 1* | 0 [1] 3                                                            | FIX32     | Tx mode CAN-OUT1 ... CAN-OUT3              |
| L-C1876 | /1 ... /4      | 58AB <sub>h</sub> =<br>22699 <sub>d</sub> | 65535                     | 0 [1] 65535                                                        | FIX32     | Masks CAN-OUT1                             |
| L-C1877 | /1 ... /4      | 58AA <sub>h</sub> =<br>22698 <sub>d</sub> | 65535                     | 0 [1] 65535                                                        | FIX32     | Masks CAN-OUT2                             |
| L-C1878 | /1 ... /4      | 58A9 <sub>h</sub> =<br>22697 <sub>d</sub> | 65535                     | 0 [1] 65535                                                        | FIX32     | Masks CAN-OUT3                             |
| L-C1882 | /1 ... /5      | 58A5 <sub>h</sub> =<br>22693 <sub>d</sub> | 0                         | 0 [1] 2<br>0: no response<br>1: Controller inhibit<br>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 <sub>d</sub> =<br>58ED <sub>h</sub> | -                 | -                                                                                                                                                                                                                                                                           | VS        | Software product code        |
| L-C1811 | -         | 22764 <sub>d</sub> =<br>58EC <sub>h</sub> |                   |                                                                                                                                                                                                                                                                             | VS        | Software creation date       |
| L-C2120 | -         | 22455 <sub>d</sub> =<br>57B7 <sub>h</sub> | 0                 | 0 = No command<br>1 = Update codes L-23XX and CAN re-initialisation = Reset node<br>2 = Update codes L-C23XX<br>10 = L-C2356/1...4 re-read<br>11 = L-C2357 re-read<br>12 = L-C2375 re-read<br>13 = L-C2376 ... L-C2378 re-read<br>14 = L-C2382 re-read<br>15 = not assigned | FIX32     | AIF control byte             |
| L-C2121 | -         | 22454 <sub>d</sub> =<br>57B6 <sub>h</sub> | 0                 | 0 [1] 255                                                                                                                                                                                                                                                                   | FIX32     | AIF status byte              |
| L-C2350 | -         | 56D1 <sub>h</sub> =<br>22225 <sub>d</sub> | 1                 | 1 [1] 63                                                                                                                                                                                                                                                                    | FIX32     | Node address                 |
| L-C2351 | -         | 56D0 <sub>h</sub> =<br>22224 <sub>d</sub> | 0                 | 0 = 500 kbits/s, 1 = 250 kbits/s<br>2 = 125 kbits/s, 3 = 50 kbits/s<br>4 = 1000 kbits/s                                                                                                                                                                                     | FIX32     | baud rate                    |
| L-C2352 | -         | 56CF <sub>h</sub> =<br>22223 <sub>d</sub> | 0                 | 0 = Slave operation<br>1 = Master operation                                                                                                                                                                                                                                 | FIX32     | Master/slave operation       |
| L-C2353 | /1 ... /3 | 56CE <sub>h</sub> =<br>22222 <sub>d</sub> | 0                 | 0 = Addressing to CANopen<br>1 = Addressing to L-C1854/L-C2354<br>2 = Addressing to LENZE system bus                                                                                                                                                                        | FIX32     | Addressing CAN-INx/ CAN-OUTx |

# 2175 (CANopen) fieldbus module

7

7.9

7.9.1

## Appendix

### Code table

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



# 2175 (CANopen) fieldbus module

## Index

## 7.10 Index

### 0 ... 9

**8200 inverter series**, 7.6-34

**8200 vector**

- Control word, 7.6-15
- Status word, 7.6-16

**82XX, Status word**, 7.6-16

**82XX**

- Control word, 7.6-15
- Status word, 7.6-16

**93XX**

- Control word, 7.6-18
- Status word, 7.6-21

### A

**Address setting**, 7.5-2

**AIF-IN, Function block**, 7.6-19

**AIF-OUT, Function block**, 7.6-21

**Appendix**, 7.9-1

**Application conditions**, 7.3-1

**Application range**, 7.2-2

### B

**Baud rate, setting**, 7.5-2

**baud rate, Function module system bus (CAN)**. *Siehe* baud rate

**Bus cable length**, 7.4-7

### C

**CANopen objects**, 7.7-1

**Code numbers, Access via the field bus module**, 7.6-26

**Code numbers / index, Conversion**, 7.6-26

**Code table**, 7.9-1

**Codes, Lenze**, 7.6-26

**Commissioning**, 7.5-1

**Components of the communication module**, 7.4-1

**Connection**

- Connections of the fieldbus module, 7.4-3
- Plug-in terminal (5 pole), 7.4-3

**Control word**, 7.6-8

- 8200 vector, 7.6-15
- 82XX, 7.6-15
- 93XX, 7.6-18, 7.6-21

### D

**Data transfer**, 7.6-1

**Device address setting**, 7.5-2

**Dimensions**, 7.3-4

**DIP switch settings / CANopen**, 7.5-1

**Double insulation**, 7.4-5

### E

**Electrical installation** , 7.4-3

**Example**

- Read block parameters, 7.6-33
- Read parameter, 7.6-31
- Write parameter, 7.6-32

### F

**Fieldbus module, Connections**, 7.4-3

**Fieldbus module 2175 (CANopen)**, 7.1-1

**Frequency setpoint**, 7.6-8

**Function module system bus (CAN), baud rate**, 7.4-7

### G

**General data**, 7.3-1

**General information**, 7.2-1

### H

**Hardware version, Type code**, 7.2-1

### I

**Identification**, 7.2-1

**Index**, 7.10-1

- Conversion, 7.6-26

**Installation**, 7.4-1

### L

**Lenze codes**, 7.6-26, 7.7-1

### M

**Mains isolation**, 7.4-5

**Mechanical installation** , 7.4-2

### N

**Network manager (NMT)**, 7.6-4

## P

Parameter channel, 7.4-6

Parameter data channel, 7.6-26

Parameter sets, 7.6-27

- Lenze, 7.6-27

Parameters

- Control word (C0135), 7.6-8
- Frequency setpoint (C0046), 7.6-8
- L-C0142, 7.5-4

Plug-in terminal for external supply, Connections, 7.4-3

Process data channel, 7.4-6

Process data telegram

- from the controller, 7.6-12
- to the controller, 7.6-11

Process data transfer, 7.6-9

Process-data assignment, AIF-CTRL

- 9300 Servo PLC, 7.6-22
- Drive PLC, 7.6-22

Processing times

- 8200, 7.3-2
- 8200 vector, 7.3-3
- 821X, 7.3-3
- 822X, 7.3-3

## R

Rated data, 7.3-1

## S

Setpoint source, 7.6-8

Setting of participant address, 7.5-2

Software version, Type code, 7.2-1

System bus (CAN), Technical data, Communication times, 7.3-2

## T

Technical data, 7.3-1

Telegram run time, 7.3-3

Troubleshooting, 7.8-1

Type code, 7.2-1

## U

User data, 7.6-28

## V

Voltage supply, external, 7.4-3