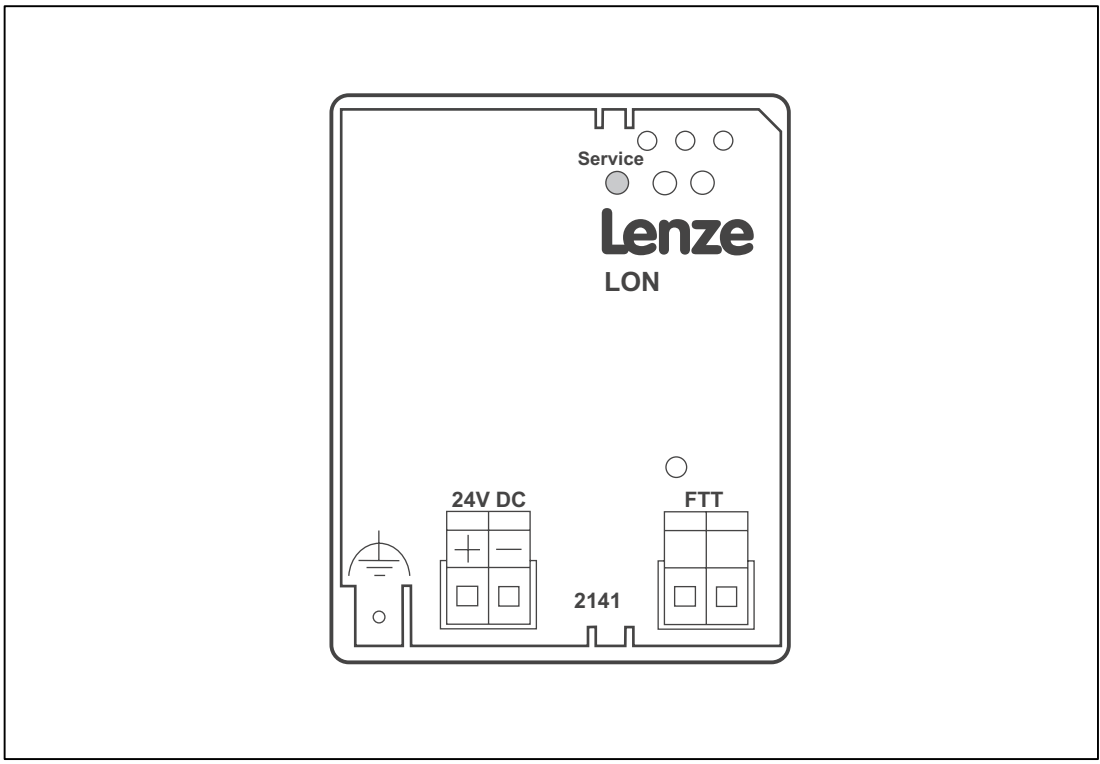
Show/Hide Bookmarks

*EDBMF2141*
455968

Lenze

# *Operating Instructions*




## *Fieldbus module type 2141*

*LON (Local Operating Network)*

These Operating Instructions are valid for fieldbus modules with the following nameplates:

2141 IB. 0x. 0x. LON

In connection with the unit series as from the nameplate data:

| | | | | | |
|---|---|---|---|---|---|
| | 820X | E./C. | 2x. | 1x. | (8201 - 8204) |
| | 821X | E./C. | 2x. | 2x. | (8211 - 8218) |
| | 822X | E. | 1x. | 1x. | (8221 - 8227) |
| | 824X | E.C. | 1x. | 1x. | (8241 - 8246) |
| | 82EVxxxxxBxxxXX | | Vx | 1x | 8200 vector |
| EPL | 10200 | I.T. | 1x. | 1x. | (Drive PLC) |
| | 93XX | E./C. | 2x. | 1x. | (9321 - 9333) |
| | 93XX | E./C. I.T. | 2x. | 1x. | (9300 Servo PLC) |

Type

Design:
E = Built-in unit IP20
IB = Module

Hardware version and index

Software version and index

Variant

Explanation

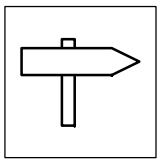
Important:

These Operating Instructions are only valid together with the corresponding Operating Instructions for the 82XX, 8200 vector, 93XX and 9300 Servo PLC and Drive PLC.

**What is new?**

| Material no. | Version | Important | Contents |
|---|---|---|---|
| 455968 | 1.0 08/02 TD02 | 1st edition | |

# Contents




**1 Preface and general information . . . . . 1-1**

1.1 How to use these Operating Instructions . . . . . 1-1
1.1.1 Terminology used . . . . . 1-1
1.2 Packing list . . . . . 1-1
1.2.1 Legal regulations . . . . . 1-2

**2 Safety information . . . . . 2-1**

2.1 Persons responsible for the safety . . . . . 2-1
2.2 General safety information . . . . . 2-1
2.3 Layout of the safety information . . . . . 2-2

**3 Technical data . . . . . 3-1**

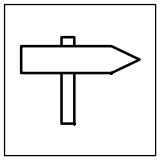
3.1 Features of the 2141 fieldbus module . . . . . 3-1
3.2 General data / application conditions . . . . . 3-1
3.3 Rated data . . . . . 3-2
3.4 Dimensions . . . . . 3-2
3.5 Communication times . . . . . 3-3
3.5.1 Processing time in the basic unit . . . . . 3-3
3.5.2 Number of devices connected to the bus . . . . . 3-3

**4 Installation . . . . . 4-1**

4.1 Elements at the front of the 2141 fieldbus module . . . . . 4-1
4.1.1 Connection of external voltage supply . . . . . 4-2
4.1.2 Connection for LON bus . . . . . 4-2
4.1.3 Data for connection terminals . . . . . 4-2
4.2 Mechanical installation . . . . . 4-3
4.3 Electrical installation . . . . . 4-4
4.3.1 Wiring to a host . . . . . 4-4
4.3.2 Voltage supply . . . . . 4-5

**5 Commissioning . . . . . 5-1**

5.1 First switch-on . . . . . 5-1
5.2 Configure the controller for the communication with the 2141 fieldbus module . . . . . 5-2
5.2.1 Protection against uncontrolled restart . . . . . 5-2

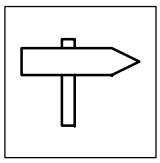


# Contents


**6 Parameter setting . . . . . 6-1**

6.1 Parameter data channel configuration . . . . . 6-2
6.1.1 Parameter data . . . . . 6-2
6.1.2 Network variable for the parameter data channel . . . . . 6-3
6.1.2.1 Structure of network variables . . . . . 6-3
6.2 Overview: Network variable and configuration variable . . . . . 6-1
6.2.1 Overview: Node Object . . . . . 6-1
6.2.2 Overview: Network variable for VSD control . . . . . 6-2
6.2.3 Overview: Network variable for AIF-CTRL control . . . . . 6-2
6.2.4 Overview: Configuration variable for VSD control . . . . . 6-3
6.3 Node Object . . . . . 6-4
6.3.1 Inhibit / Enable all objects . . . . . 6-4
6.3.2 Status messages . . . . . 6-4
6.4 VSD control VSD . . . . . 6-5
6.4.1 Network variable (VSD) . . . . . 6-5
6.4.1.1 Speed setpoint . . . . . 6-5
6.4.1.2 Actual speed value . . . . . 6-6
6.4.1.3 Scaling of the speed setpoint . . . . . 6-7
6.4.1.4 Motor current . . . . . 6-8
6.4.1.5 Motor voltage . . . . . 6-9
6.4.1.6 Operating time . . . . . 6-10
6.4.2 Configuration variable (VSD) . . . . . 6-11
6.4.2.1 Location label . . . . . 6-11
6.4.2.2 Receive Heartbeat Time . . . . . 6-12
6.4.2.3 Maximum Send Time . . . . . 6-13
6.4.2.4 Minimum Send Time . . . . . 6-14
6.4.2.5 Maximum speed . . . . . 6-15
6.4.2.6 Minimum speed . . . . . 6-16
6.4.2.7 Rated speed value . . . . . 6-17
6.4.2.8 Rated frequency . . . . . 6-18
6.4.2.9 Acceleration time . . . . . 6-19
6.4.2.10 Deceleration time . . . . . 6-20
6.4.2.11 Receive heartbeat for AIF-CTRL . . . . . 6-21
6.4.2.12 Monitoring reaction for AIF-CTRL . . . . . 6-22
6.4.2.13 Default value for nviDrvSpeedScale . . . . . 6-23
6.5 Device control AIF-CTRL . . . . . 6-24
6.5.1 General information . . . . . 6-24
6.5.1.1 Output variable . . . . . 6-24
6.5.1.2 Input variable . . . . . 6-25
6.5.2 Network variable for 82XX controllers . . . . . 6-26
6.5.2.1 Status word for 82XX . . . . . 6-27
6.5.2.2 Control word for 82XX . . . . . 6-29
6.5.3 Network variable for 8200 vector controllers . . . . . 6-30
6.5.3.1 Status word for 8200 vector . . . . . 6-32
6.5.3.2 Control word for 8200 vector . . . . . 6-33
6.5.4 Network variable for 93XX controllers . . . . . 6-35
6.5.4.1 Status word for 93XX . . . . . 6-37
6.5.4.2 Control word for 93XX . . . . . 6-40
6.5.5 Network variable for 9300 Servo PLC and Drive PLC . . . . . 6-42

# Contents




**7 Troubleshooting and fault elimination . . . . . 7-1**

7.1 No communication with the controller. . . . . 7-1

7.2 Controller does not execute write job . . . . . 7-1

**8 Appendix . . . . . 8-1**

8.1 Starting Global Drive Control (GDC) from LONMakert . . . . . 8-1

8.1.1 Installation instructions for the plug-In . . . . . 8-1

8.1.2 Open GDC from the LONMakert . . . . . 8-2

8.2 Configuration of the LON network . . . . . 8-3

8.2.1 Configuration of the LON network . . . . . 8-3

8.2.2 Working with network variables . . . . . 8-5

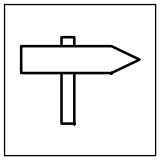
8.2.3 Registration of new data types . . . . . 8-8

8.3 List of abbreviations . . . . . 8-9

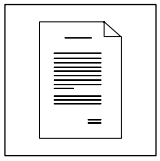
8.4 Glossary . . . . . 8-10

8.5 More information sources . . . . . 8-10

8.6 Table of keywords . . . . . 8-11



# Contents



# 1 Preface and general information

## 1.1 How to use these Operating Instructions

- These Operating Instructions are intended for safety-relevant working on and with the 2141 fieldbus module. They contain safety information which must be observed.
- All personnel working on and with the 2141 fieldbus module must have these Operating Instructions available and observe the information and notes relevant for them.
- The Operating Instructions must always be complete and perfectly readable.

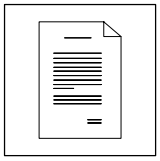
These Operating Instructions contain the most important technical data and describe the installation of the 2141 fieldbus module. They are only valid in combination with the Operating Instructions of the corresponding controller.

### 1.1.1 Terminology used

| | |
|---|---|
| **Controller** | In the following, the term ”controller” is used for ”93XX servo inverters” or ”82XX frequeny inverters”. |
| **Drive system** | In the following, the term ”drive system” is used for drive systems with fieldbus modules or other Lenze drive components. |
| **Field bus module** | In the following text, the term ”fieldbus module” is used for the fieldbus module type 2141 LON. |
| **Cxxx/y** | Subcode y of code Cxxx (z.B. C0410/3 = subcode 3 of code C0410) |
| **Xk/y** | Terminal strip Xk/terminal y (z.B. X3/28 = terminal 28 on terminal strip X3) |
| (📖xx-yyy) | Cross reference (chapter - page) |

## 1.2 Packing list

| **Scope of supply** | **Important** |
|---|---|
| • 1 2141 fieldbus module with housing (enclosure IP20)<br>• 1 M3 fixing screw<br>• 1 2-pole plug connector for voltage supply<br>• 1 2-pole plug connector for LON<br>• 1 Mounting Instructions<br>• 1 floppy | After the delivery has been received, check immediately whether the items supplied match the accompanying papers. Lenze does not accept any liability for deficiencies claimed subsequently.<br>Claim<br>• visible transport damage immediately to the forwarder<br>• visible deficiencies/incompleteness immediately to your Lenze representative. |





## 1.2.1 Legal regulations

| | | | |
|---|---|---|---|
| **Labelling** | **Nameplate** | **CE mark** | **Manufacturer** |
| | Lenze 2141 fieldbus modules are unambiguously identified by their nameplates. | Conforms to the EC Low Voltage Directive | Lenze Drive Systems GmbH<br>Postfach 101352<br>D-31763 Hameln |
| **Application as directed** | **Fieldbus module 2141**<br>• Operate the fieldbus module only under the conditions prescribed in these Operating Instructions.<br>• The fieldbus module is an additional module and can be optionally attached to the Lenze controller series 820X, 821X, 822X, 8200 vector and 93XX. The 2141 fieldbus module links these Lenze controllers with the standardized serial communication system LON.<br>• The fieldbus module must be attached and electrically connected so that it complies with its function and does not cause any hazards when attached and operated as instructed.<br>• Observe all notes given in chapter „Safety information“ 🕮 2-1.<br>• Please observe all information given in these Operating Instructions.This means:<br>– Read these Operating Instructions carefully before you start to work with the system.<br>– These Operating Instructions must always be available during operation of the fieldbus module.<br>**Any other use shall be deemed as inappropriate!** | | |
| **Liability** | • The information, data, and notes in these instructions met the state of the art at the time of printing. Claims referring to drive systems which have already been supplied cannot be derived from the information, illustrations, and descriptions given in these Operating Instructions.<br>• The specifications, processes, and circuitry described in these instructions are for guidance only and must be adapted to your own specific application. Lenze does not take responsibility for the suitability of the process and circuit proposals.<br>• The specifications in these Instructions describe the product features without guaranteeing them.<br>• Lenze does not accept any liability for damage and operating interference caused by:<br>– disregarding these Instructions<br>– unauthorized modifications to the controller<br>– operating errors<br>– improper working on and with the controller | | |
| **Warranty** | • Warranty conditions: see Sales and Delivery Conditions of Lenze Drive Systems GmbH.<br>• Warranty claims must be made to Lenze immediately after detecting the deficiency or fault.<br>• The warranty is void in all cases where liability claims cannot be made. | | |
| **Disposal** | **Material** | **recycle** | **dispose** |
| | Metal | ● | - |
| | Plastic | ● | - |
| | Assembled PCBs | - | ● |
| | Operating Instructions | ● | |

# 2 Safety information

## 2.1 Persons responsible for the safety

**Operator**

- An operator is any natural or legal person who uses the drive system or on behalf of whom the drive system is used.
- The operator or his safety personnel is obliged
  - to ensure the compliance with all relevant regulations, instructions and legislation.
  - to ensure that only skilled personnel works on and with the 2102IB fieldbus module.
  - to ensure that the personnel has the Operating Instructions available for all corresponding works.
  - to ensure that all unqualified personnel are prohibited from working on and with the drive system.

**Qualified personnel**

Qualified personnel are persons who - because of their education, experience, instructions and knowledge about corresponding standards and regulations, rules for the prevention of accidents and operating conditions - are authorized by the person responsible for the safety of the plant to perform the required actions and who are able to recognize potential hazards.
(Definition for skilled personnel to VDE105 or IEC364)

## 2.2 General safety information

- These safety notes do not claim to be complete. In case of questions and problems please contact your Lenze representative.
- At the time of delivery the fieldbus module meets the state of the art and ensures basically safe operation.
- The indications given in these Operating Instructions refer to the stated hardware and software versions of the fieldbus modules.
- The fieldbus module is hazardous if:
  - unqualified personnel works on and with the fieldbus module.
  - the fieldbus module is used inappropriately.
- The processing notes and circuit sections shown in these Operating Instructions are proposals which cannot be transferred to other applications without being tested and checked.
- Ensure by appropriate measures that neither personal injury nor damage to property may occur in the event of failure of the fieldbus module.
- The drive system must only be operated when no faults occur.
- Retrofittings, modifications or redesigns are basically prohibited. Lenze must be contacted in all cases.
- The fieldbus module is electrical equipment intended for use in industrial high-power plants. The fieldbus module must be tightly screwed to the corresponding controller during operation. In addition, all measures described in the Operating Instructions of the controller used must be taken. Example: Fastening of covers to ensure protection against contact.



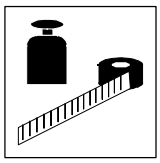
## 2.3 Layout of the safety information

All safety information given in these Instructions have got the same structure:

**Signal word** (indicates the severity of danger)

Note (describes the danger and informs the reader how to avoid danger)

| | Icons used | | Signal words | |
|---|---|---|---|---|
| **Warning of danger to persons** |  | Warning of hazardous electrical voltage | **Danger!** | Warns of **impending dangerr**.<br>Consequences if disregarded:<br>Death or severe injuries. |
| |  | Warning of a general danger | **Warning!** | Warns of **potential, very hazardous situations**.<br>Consequences if disregarded:<br>Death or severe injuries. |
| | | | **Caution!** | Warns of **potential, hazardous situations**.<br>Consequences if disregarded:<br>Light or minor injuries. |
| **Warning of damage to materials** |  | | **Stop!** | Warns of **possible damage to material**.<br>Consequences if disregarded:<br>Damage of the controller/drive system or its environment. |
| **Other notes** |  | | **Tip!** | It designates general, useful notes.<br>If you follow the tip, handling of the controller/drive system will be easier. |



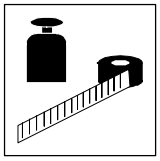
# 3 Technical data

## 3.1 Features of the 2141 fieldbus module

- Attachable additional module for the following devices:
  – 82XX
  – 8200 vector
  – Drive PLC
  – 93XX
  – 9300 Servo PLC
- LON connection module with the LONMARK® Functional Profile “Variable Speed Motor Drive”
- Access to all Lenze parameters
- Internal and external 24-V-DC voltage supply possible with devices listed above (exception 820X: only external supply)

## 3.2 General data / application conditions

| Field | Values |
|---|---|
| Order name | EMF2141IB |
| Network topology | Free topology (line, tree, star, ring) |
| Possible number of nodes | 64 |
| Max. cable length | 2700 m with bus topology<br>500 m if the topology is mixed |
| Baud rate | 78 kBit/s |
| Ambient temperature | during operation: 0° C ... 55° C<br>during transport: -25° C ... 70° C<br>during storage: -25° C ... 60 °C |
| Permissible humidity | Class 3K3 to EN 50178 (without condensation, average relative humidity 85%) |





## 3.3 Rated data

| Field | Values | | |
|---|---|---|---|
| Voltage supply | 24 V DC ± 10 %; max. 120 mA | | |
| Communication profile | LONMARK® Functional Profile “Variable Speed Motor Drive” | | |
| Communication medium | FTT - 10 A (Free Topology Transceiver) | | |
| Insulation voltage between bus and ... | | **Rated insulation voltage** | **Insulation type** |
| • Earth reference / PE | | 50 V AC | Mains isolation |
| • External supply (terminal 39/59) | | 50 V AC | Mains isolation |
| • Power stage | | | |
| – 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 (with internal supply) | | 0 V AC | No mains isolation |
| – 8200 vector (with external supply | | 100 V AC | Basic insulation |
| – 821X | | 50 V AC | Mains isolation |
| – 822X | | 270 V AC | Basic insulation |
| – 93XX | | 270 V AC | Basic insulation |
| Degree of pollution | VDE0110, part 2, pollution degree 2 | | |

## 3.4 Dimensions

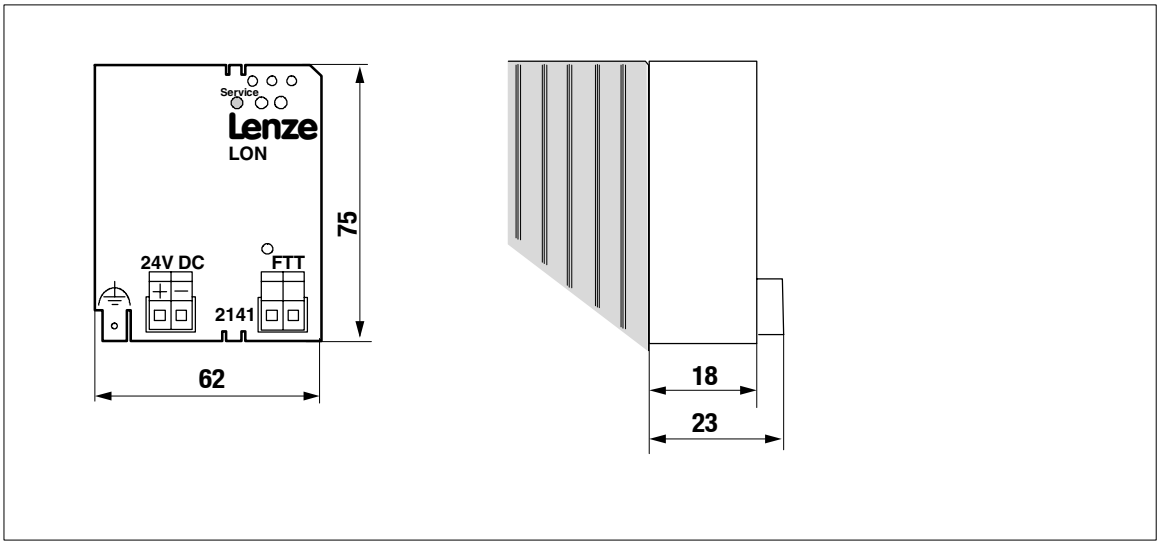
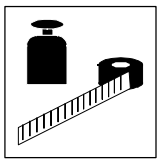



Fig. 3-1 Dimensions of the 2141 fieldbus module (all dimensions in mm)



## 3.5 Communication times

The communication times depend on

- the data transmission time which depends on
  – the baud rate
  – the user-data length
- the processing time in the basic unit (see the following chapter)

**Tip!**

You will find more information about the data-transfer time in the documentation for your host system.

### 3.5.1 Processing time in the basic unit

The processing time in the basic unit starts when a message arrives at the fieldbus module (action) and ends with the response of the basic unit to this message.

**Processing time 820X**

| Action | Processing step | Max. time required by the controller |
|---|---|---|
| • Writing of control word or setpoint, if the value has changed | Writing of status word and actual value | 200 ms +40 ms tolerance |
| • Alternating reading of status word and actual value | Control word or setpoint reading | 27 ms +48 ms Toleranz |
| | Control word and setpoint reading | 54 ms +56 ms tolerance |
| • Processing of parameter access if there is a service | Read parameter | 55 ms +48 ms tolerance |
| | Parameter writing | 108 ms +32 ms tolerance |

Tab. 3-1 Processing time: 820X

**Processing time: 821X / 8200 vector / 822X**

Parameter data and process data are independent of each other.

- Parameter data: 30 ms + 20 ms tolerance
- Process data: approx. 3 ms + 2 ms tolerance

**Processing time: 9300 servo inverter**

Parameter data and process data are independent of each other.

- Parameter data: approx. 30 ms + 20 ms tolerance (typical)
  – The processing time can be longer for some codes (see 9300 Manual).
- Process data: approx. 3 ms + 2 ms tolerance

**Processing time: Drive PLC / 9300 Servo PLC**

Parameter data and process data are independent of each other.

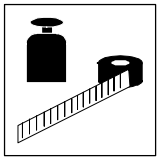
- Parameter data: approx. 30 ms + 20 ms tolerance
- Process data depend on the process image.
  The process image is an update of process data made after every task cycle.

### 3.5.2 Number of devices connected to the bus

The maximum bus size depends on

- the baud rate used
- the number of repeaters

For more detailed information see the documenation for the control system.



# Technical data

# 4 Installation

## 4.1 Elements at the front of the 2141 fieldbus module

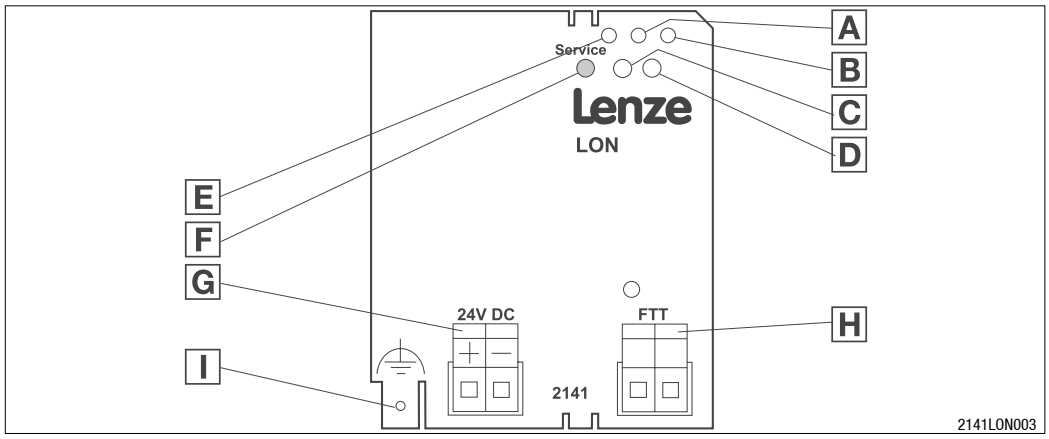
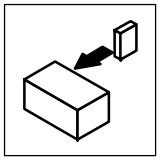



Fig. 4-1 Front view of the 2141 LON fieldbus module

| Pos. | Name/meaning |
|---|---|
| A | **Green bus LED** indicates the voltage supply status for the 2141 fieldbus module.<br>ON: 2141 fieldbus module is supplied with voltage and connected to the controller.<br>OFF: 2141 is not supplied with voltage; controller and external voltage supply are switched off.<br>BLINKING:<br>2141 fieldbus module is supplied with voltage but not connected to the controller. Possible reasons: Controller is switched off, in initialisation or not connected. |
| B | **Yellow bus LED** indicates the communication status for the 2141 fieldbus module.<br>OFF: 2141 fieldbus module has not been initialised yet.<br>BLINKING: LON communication is not active |
| C/ D | **Green or red DRIVE-LED**<br>Operating status of 82XX, 8200 vector or 93XX (see Operating Instructions for controller) |
| E | **Service LED**<br>OFF: Normal operating status<br>BLINKING: Fieldbus module not configured yet<br>ON: No program available in Neuron |
| F | **Service key**<br>e.g. for signing on a fieldbus module with the configuration tool " LONMaker™ for Windows® " 🕮 8-1 |
| G | **Two-pole plug for external voltage supply of the 2141 fieldbus module**<br>+: Input voltage (see chapter: Technical Data)<br>- : Reference potential for external voltage (GND) |
| H | **Two-pole plug for LON communication**<br>Max. cable cross-section: 1.5 $mm^2$ |
| I | Only for 820X, 821X: Additional PE shield cable avoids EMC related communication interference |





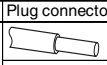
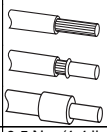
## 4.1.1 Connection of external voltage supply

| Name | Explanation |
|---|---|
| 24 V DC /+ | External voltage supply +24 V DC/ ± 10 %, 120 mA max. |
| 24 V DC /- | GND; reference for external voltage supply |

## 4.1.2 Connection for LON bus

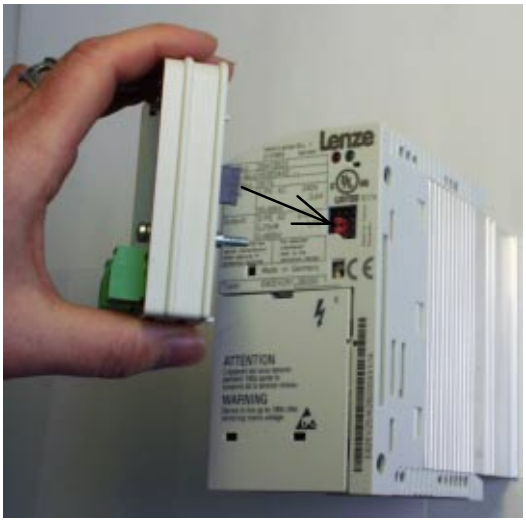
| Name | Explanation |
|---|---|
| FTT / | Connection for LON bus, two cores (twisted pair) |

## 4.1.3 Data for connection terminals

| | |
|---|---|
| **Electrical connection** | Plug connector with threaded terminal end |
| **Possible connections** |  rigid: 1.5 mm² (AWG 16) |
| |  flexible:<br>without wire crimp cap<br>1.5 mm² (AWG 16)<br>with wire crimp cap,without plastic sleeve<br>1.5 mm² (AWG 16)<br>with wire crimp cap, with plastic sleeve<br>1.5 mm² (AWG 16) |
| **Tightening torque** | 0.5 Nm (4.4 lb-in) |
| **Bare end** | 7 mm |

## 4.2 Mechanical installation

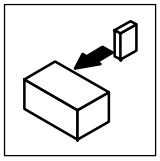
- Plug the fieldbus module onto the controller (here: 8200 vector)




2141LON001

- Use the fixing screw to mount the fieldbus module.


2141LON002





## 4.3 Electrical installation

### 4.3.1 Wiring to a host


**Stop!**

An additional mains isolation is required, if

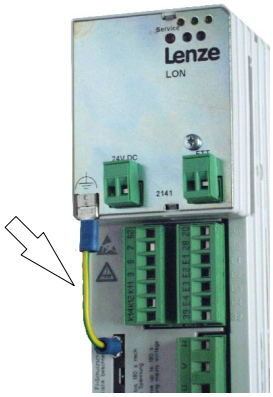
- a 820X or 821X controller is connected to a host and
- a safe mains isolation (double basic insulation) to VDE 0160 is necessary.

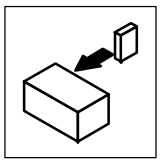
For this, it is, for instance, possible to use an interface module which provides an additional mains isolation (see corresponding manufacturer's data).

For wiring, the electrical isolation of the supply voltage must be taken into account.

**Tip!**

The communciation of controllers 820X and 821X may be interfered by electromagnetic radiation. If necessary, use an additional PE shield cable .





## 4.3.2 Voltage supply

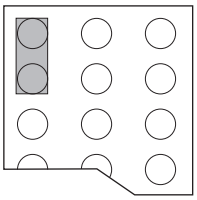
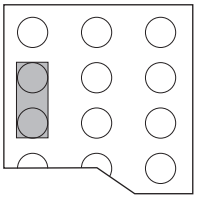
- Internal voltage supply
  – If possible, use all devices without external voltage supply because of EMC-relevant reasons.
- External voltage supply (📖 4-1)
  – Absolutely required for 820X controllers.
  – If the 2141 fieldbus module is to be able to communicate even if the basic device is switch off.
- If the bus system is distributed over several control cabinets, install a separate power supply in every control cabinet to supply all devices/fieldbus modules.
  Install an equipotential bonding conductor between the control cabinets.

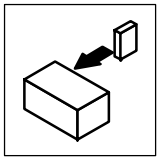
### Tip!

**Internal voltage supply of the fieldbus module connected to a 8200 vector**

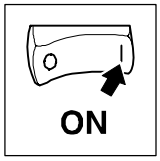
Controllers with an extended AIF interface (front of the 8200 vector) can be internally supplied. The part of the drawing highlighted in grey shows the jumper position.

With Lenze setting, the fieldbus module is not internally supplied.
For internal voltage supply, put the jumper in the position indicated below.

| **Lenze setting**<br>(only external voltage supply) | **Internal voltage supply** |
|---|---|
|  |  |



# Installation




# 5 Commissioning

## 5.1 First switch-on

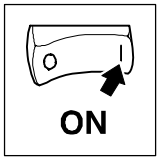
**Please do not change the switch-on sequence explained below!**


### Stop!

Prior to first switch-on of the controller, check the wiring for

- Completeness
- Short circuit
- Earth fault

| Step | Procedure | Remarks |
|---|---|---|
| 1. | Connect the basic unit and if available external supply for the fieldbus module. | • Onle of the two operating status LEDs at the controller, see 🕮 4-1, must be on or blinking. If this is not the case, see 🕮 7-1<br>• The green Vcc display must also be on, see 🕮 4-1. If this is not the case, see 🕮 7-1. |
| 2. | Configuration of controllers connected to the bus | Configure the controllers connected to the bus by means of an appropriate configuration tool<br>→ e.g. LONMAKER™ for Windows®, see 🕮 8-1 |
| 3. | It should now be possible to communicate with the drive. | The yellow bus LED is blinking, see 🕮 4-1, when data is exchanged between the 2141 fieldbus module and other controllers connected to the bus. |
| 4. | Drive-specific settings | → Operating Instructions of the basic unit |
| 5. | Enable controller | → Terminal controller inhibit (CINH) = HIGH |
| 6. | Setpoint selection | → The drive should be running now. |






## 5.2 Configure the controller for the communication with the 2141 fieldbus module

| | |
|---|---|
| **82XX / 8200 vector** | 1. The drive can be controlled through LON, if the Lenze parameter “Operating mode” (C0001) is changed from 0 to 3. This change can be made by using the keypad or directly via LON.<br>• Example:<br>For LON Write (C0001=3), subindex: 0, value = 30000.<br>Please see the information about how to change parameter values of a code in chapter 6.1.2.1.<br>2. Terminal 28 (controller enable) is always active and must be HIGH for LON operation (see the corresponding Operating Instructions). Otherwise, the controller cannot be enabled by the LON.<br>3. 821X, 8200vector and 822X:<br>The QSP function (quick stop) is always active.<br>A input terminal configured for QSP (default setting: not assigned) must be HIGH during LON operation (see Operating Instructions for the controller).<br>4. The controller now accepts control and parameter-setting data from the LON. |
| **93XX** | 1. If you want to control the drive via LON, configure the Lenze parameter “Signal configuration” (C0005) for xxx3. This change can be carried out using the 9371BB keypad or the LON. Select the signal configuration 1013 for first commissioning.<br>• Example:<br>For LON Write (C0005=1013), subindex: 0, value: 10130000.<br>Please see the information about how to change parameter values of a code in chapter 6.1.2.1.<br>2. Set the parameter C0142 to 0.<br>Please read the next TIP for code C0142.<br>3. Terminal 28 (RFR = controller enable) is always active and must be HIGH for LON operation (see Operating Instructions for the controller). Otherwise, the controller cannot be enabled by the LON.<br>4. With the signal configuration C0005=1013, the QSP function (quick stop) and the CW/CCW changeover are assigned to the input terminals E1 and E2, and thus they are always active.<br>E1 must be HIGH for LON operation (see 93XX Operating Instructions).<br>With the signal configuration C0005=xx13, terminal A1 is configured as voltage output. Therefore, only the following terminals must be connected via cables:<br>• X5.A1 with X5.28 (ctrl. enable)<br>• X5.A1 with X5.E1 (CW/QSP)<br>5. The controller now accepts control and parameter-setting data from the LON. |

### 5.2.1 Protection against uncontrolled restart

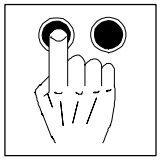
**Tip!**

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

- The start condition can be changed, i.e. the drive can be inhibited by setting C0142 = 0 for the event that
  - – the corresponding controller sets a fault „Message“
  - – the fault is active for more than 0.5 s

Parameter function:

- 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“ (CHIN, e.g. at terminal X5/28)
- C0142 = 1
  - – Uncontrolled restart of the controller possible



# 6 Parameter setting

### General information on parameter setting

Two different data types are transferred by the 2141 fieldbus module:

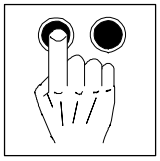
- Parameter data
- Process data,

Different communication channels are assigned to parameter and process data:

| Telegram type | | Communication channel | |
|---|---|---|---|
| **Parameter data** | These are, for instance,<br>• Operationg parameters<br>• diagnostics information<br>• motor data<br>In general, the parameter transfer is not as time-critical as the tranfer of process data. | **Parameter data channel (chapter 6.1)** | • Enables the access to all Lenze codes.<br>• Parameter changes are normally saved in the controller (observe C0003).<br>• If the parameter channel is active, it assigns 4 words of the input and output process data. 🕮 6-2 ) |
| **Process data** | These are, for instance,<br>• Setpoints<br>• Actual values<br>Exchange between host and controller required as fast as possible. Small amounts of data, which are transferred cyclically. | **Process-data channel (chapter 6.2)** | • The controller can be controlled using the process data. ( 🕮 6-7 ).<br>• The host has direct access to the process data. In the PLC, for instance, the data are directly assigned to the I/O area.<br>• Process data are<br>– not stored in the controller.<br>– cyclically transferred between the host and the controller (continuous exchange of current input and output data). |

Tab. 6-1 Division of parameter data and process data into different communication channels

The communication protocol is only described as much as it is needed for building up the network with Lenze controllers.





# 6.1 Parameter data channel configuration

## 6.1.1 Parameter data

Parameter data are addressed through Lenze codes. You find a code table in the Operating Instructions for your controller.

### Lenze parameter sets

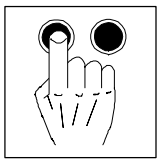
Parameter sets are for special code saving which is necessary because of different configurations for different application processes.

The following table informs about the number of parameter sets and how to address them.

<table>
<tr><th>82XX</th><th>8200 vector</th><th>93XX</th></tr>
<tr><td colspan="2">82XX and 8200 vector controllers have 2 and 4 parameter sets. They can be directly addressed through LON.<br>They are addressed by means of a code-digit offset:<br>• Offset 0 addesses parameter set 1 with Lenze codes C0000 to C1999<br>• Offset 2000 addresses parameter set 2 with Lenze codes C2000 to C3999</td><td rowspan="4">93XX controllers have 4 parameter sets (depending on the variant) for non-volatile saving.<br>Another parameter set is in the user memory of the controller. This is the current parameter set.<br>After switching on the controller, parameter set 1 is automatically loaded into the current parameter set.<br>Changes of the current parameter set will be lost after switching off the controller.<br>The current parameter set is stored in code C0003.<br>Only the current parameter set can be directly addressed through LON.<br>The 93XX codes are listed in the Operating Instructions and the Manual.</td></tr>
<tr><td>No additional parameter sets available.</td><td>• Offset 4000 addresses parameter set 3 with the Lenze codes C4000 to C5999<br>• Offset 6000 addresses parameter set 4 with the Lenze codes C6000 to C7999</td></tr>
<tr><td colspan="2">If a parameter is available only once (see Operating Instructions for 82XX or 8200 vector), use the code digit offset 0.</td></tr>
<tr><td colspan="2">Example for C0011 (maximum field frequency):<br>C0011 in parameter set 1: Lenze code = 11<br>C0011 in parameter set 2: Lenze code = 2011</td></tr>
<tr><td>-</td><td>C0011 in parameter set 3: Lenze code = 4011<br>C0011 in parameter set 4: Lenze code = 6011</td><td></td></tr>
<tr><td colspan="3">Parameter changes</td></tr>
<tr><td>Automatic saving in the controller</td><td>Automatic saving (can be switched off under C0003)</td><td>Changes must be saved in code C0003.<br>All changes made without storing them will be lost after the controller is switched off.</td></tr>
<tr><td colspan="3">Process data changes</td></tr>
<tr><td>No automatic saving</td><td>No automatic saving</td><td>No automatic saving</td></tr>
</table>


**Stop!**

- For 8200 vector
  – Please note that it is not allowed to write parameter data to the EEPROM.
  – If you want to cyclically change parameter data, the code must be configured as follows after every mains switching: C0003 = 0.
- For 82XX
  – Please note that it is not allowed to cyclically write parameter data to the EEPROM.



## 6.1.2 Network variable for the parameter data channel

The network variables `nviParamReq`, `nvoParamRes` and `nvoLongRes`

- enable parameter setting and diagnostics.
- allow access to all Lenze parameters (codes).

The input network variable `nviParamReq` allows to send a read or write request to the drive.

With parameters with a length of up to 4 bytes, the response is output through the output network variable `nvoParamRes`.

With parameters bigger than 4 bytes, the response to a read request is output through the network variable `nvoLongRes`.

### 6.1.2.1 Structure of network variables

**Tip!**

User data are displayed in left-justified Motorola format.

High byte or high word are shown before low byte or low word (see

Calculation examples (📖 6-6) ).

The structure of `nviParamReq` and `nvoParamRes` is defined as follows:

| 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6th byte | 7th byte | 8th byte | 9th byte |
|---|---|---|---|---|---|---|---|---|
| **Length** | **code** High byte | **code** Low byte | **Subcode** | **Service** | **Data 4** | **Data 3** | **Data2** | **Data1** |
| | | | | | | | | **Error code** |

The structure of `nvoLongRes` is defined as follows:

| 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6th byte | .... | 28th byte |
|---|---|---|---|---|---|---|---|
| **Length** | **code** High byte | **code** Low byte | **Subcode** | **Service** | **Data 1** | .... | **Data23** |

| Details about the parameter data telegram: | 1st byte |
|---|---|
| | **Length** |

Indicates the length of the following byte for write orders. Value range: 1, 2 or 4.

| Details about the parameter data telegram: | 2nd byte | 3rd byte |
|---|---|---|
| | **code** High byte | **code** Low byte |

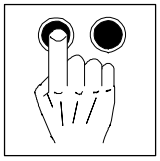
Byte 2 and byte 3 contain the code to be used.

| Details about the parameter data telegram: | 4th byte |
|---|---|
| | **Subcode** |

Byte 4 contains the subcode to be used.

- The series 82XX do not have codes with subindex, the value is always 0.
- Many codes of 93XX and 8200 vector controllers require additional addressing via a subindex.

Example for 9300 servo inverters:
Code C0039 / subcode 3 addresses ” NSET JOG” (50% = default setting)



# Parameter setting

| Details about the parameter data telegram: | 5th byte |
|---|---|
| | **Service** |

**Order type**

1 = Read

2 = Write

**Data contents**

- Response ok
  – Read: 1
  – Write: 2
- Faulty response
  – Read: $81_{hex}$ ($80_{hex}$ + 1)
  – Write: $82_{hex}$ ($80_{hex}$ + 2)

| Details about the parameter data telegram: | 6th byte | 7th byte | 8th byte | 9th byte |
|---|---|---|---|---|
| | **Data 4** | **Data 3** | **Data 2** | **Data 1** |

Bytes 6 to 9 contain the data to be written.

Bytes 6 to 9 are not relevant for reading orders.

The parameters of the corresponding code are changed by multiplying the parameter value with 10000 (the result is an integer value). Lenze parameters are mainly in the fixed-point format ,data type Integer32, with four decimal codes.

**Tip!**

Please see the Operating Instructions for your controller for the value range of Lenze codes.

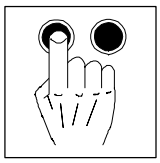
| **Lenze parameters** | | **dec** | **hex** |
|---|---|---|---|
| Example: | Set C0039 (JOG) = 150.4 Hz | 150.4 x 10000 = 1504000 | = 0016F300 |

The parameter value needs 1 to 4 bytes depending on the data format (see “Attribute table” in the Operating Instructions for your controller)

| 6th byte | 7th byte | 8th byte | 9th byte |
|---|---|---|---|
| Data 4 | Data 3 | Data 2 | Data 1 |
| **High byte** | **Low byte** | **High byte** | **Low byte** |
| **High word** | | **Low word** | |
| **Double word** | | | |

Assignment of bytes 6 to 9 with parameter value of different lengths

| 6th byte | 7th byte | 8th byte | 9th byte |
|---|---|---|---|
| 00 | 00 | 00 | Parameter value (length 1) |
| 00 | 00 | Parameter value (length 2) | |
| Parameter value (length 4) | | | |



| Details about the parameter data telegram: | 6th byte | 7th byte | 8th byte | 9th byte |
|---|---|---|---|---|
| | | | | **Error code** |

If processing the order is completed correctly, the variable `nvoParamRes` contains the read data or a copy of the written data.

If an error occurs in the network variable `nvoParamRes` the error code will be entered in byte 9. The error code can be read from byte 9 if

- the value of byte 1 = 1 and
- bit 7 is set in byte 5 (i.e. the value of byte 5 = $80_{hex}$ , see example below).

**Tip!**

Strings and data blocks cannot be written.

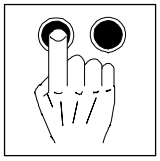
The following table explains the error codes.

| **Error code** | | |
|---|---|---|
| dec | hex | **Meaning of error message** |
| 01 | 01 | Service-ID invalid |
| 02 | 02 | Caller-ID invalid |
| 03 | 03 | Data type invalid |
| 04 | 04 | Subcode-No. invalid |
| 05 | 05 | Code-No. invalid |
| 06 | 06 | Parameter invalid, general |
| 07 | 07 | No access authorisation because of operating status |
| 08 | 08 | No access authorisation because of operating mode |
| 09 | 09 | No access authorisation because parameter can only be read |
| 10 | A | No access authorisation |
| 11 | B | Data block too long |
| 12 | C | Collision with other parameter values |
| 14 | D | Quit value range |
| 15 | I | General limit value exceeding |
| 32 | 20 | General |
| 33 | 21 | Time limit exceeded |
| 34 | 22 | Frame error |
| 35 | 23 | Parity error |
| 36 | 24 | Overflow |

Example:

The brake opening time, code C0196 ($C4_{hex}$), is wrong.
Error code 14, 'Parameter value exceeds the max. permissible value'.

| 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6th byte | 7th byte | 8th byte | 9th byte |
|---|---|---|---|---|---|---|---|---|
| **01** | **00** | **C4** | **00** | **80** | **00** | **00** | **00** | **0E** |



# Parameter setting

### Example 1: Read parameter

*The heat sink temperature* (Here: $\vartheta$ = 43° C) *of the controller is to be read* (C0061).

- Byte 1: Length
  – 4 (data length 4 bytes)
- Byte 2 and byte 3: Code
  – C0061: 61 = $3D_{hex}$
- Byte 4: Subcode
  – Subindex = 0, as there is not subindex under code C0061.
- Byte 5: Service
  – Service = 1 (Read)
- Byte 6 ... 9: Data (part of the response of `nvoParamRes`)
  – Data 1 to data 4 = 43[° C] x 10000 = 430000 = 00 06 8F $B0_{hex}$

*Result:*

| | Byte1 | Byte 2 + byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 | Byte 9 |
|---|---|---|---|---|---|---|---|---|
| | Length | code | Lenze subcode | Service | Parameter value | | | |
| **Request telegram from master to drive** | | | | | | | | |
| Hex: | **04** | **003D** | **00** | **01** | 00 | 00 | 00 | 00 |
| Wait for response identification with code = 003D and subcode 0 | | | | | | | | |
| **Response telegram from drive to master (for faulty execution)** | | | | | | | | |
| Hex: | 04 | 003D | 00 | 01 | **00** | **06** | **8F** | **B0** |

Tab. 6-2 Telegram exchange in LON parameter data channel

### Example 2: Write parameter

*The acceleration time* (C0012) *of the controller is to be set to* $T_{ir}$ *= 20 s.*

- Byte 1: Length
  – 4 (data length 4 bytes)
- Byte 2 and byte 3: Code
  – C0012: 12 = $0C_{hex}$
- Byte 4: Subcode
  – Subindex = 0, as there is not subindex under code C0061.
- Byte 5: Service
  – Service = 2 (Write)
- Byte 6 ... 9: Data (part of the response of `nvoParamRes`)
  – Data 1 to data 4 = 20 [s] x 10000 = 200000 = 00 03 0D $40_{hex}$

*Result:*

| | Byte1 | Byte 2 + byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 | Byte 9 |
|---|---|---|---|---|---|---|---|---|
| | Length | code | Lenze subcode | Service | Parameter value | | | |
| **Request telegram from master to drive** | | | | | | | | |
| hex: | **04** | **000C** | **00** | **02** | 00 | 03 | 0D | 40 |
| Wait for response identification with code = 00C and subcode 0 | | | | | | | | |
| **Response telegram from drive to master (for faulty execution)** | | | | | | | | |
| hex: | 04 | 000C | 00 | 02 | **00** | **00** | **00** | **00** |

Tab. 6-3 Telegram exchange in LON parameter data channel

## 6.2 Overview: Network variable and configuration variable

Communication with 2141 fieldbus module via network variable.

The control **Variable Speed Motor Drive (VSD)** is the part of the network variable defined in the LonMark® Functional Profile.

The Lenze control **AIF-CTRL** is the manufacturer-related part of the network variable. Lenze drives are parameterised and controlled by AIF-CTRL.

The changeover between the two controls is automatic.

- VSD is active if one of the variables `nviDriveSpeedStpt` or `nviDriveSpeedScale` gets a new value.
- AIF-CTRL is active if one of the variables `nviAIFIn1` to `nviAIFIn3` or `nviAIFStw` gets a new value.

The network variable **Node Object** accesses the entire node.

### 6.2.1 Overview: Node Object

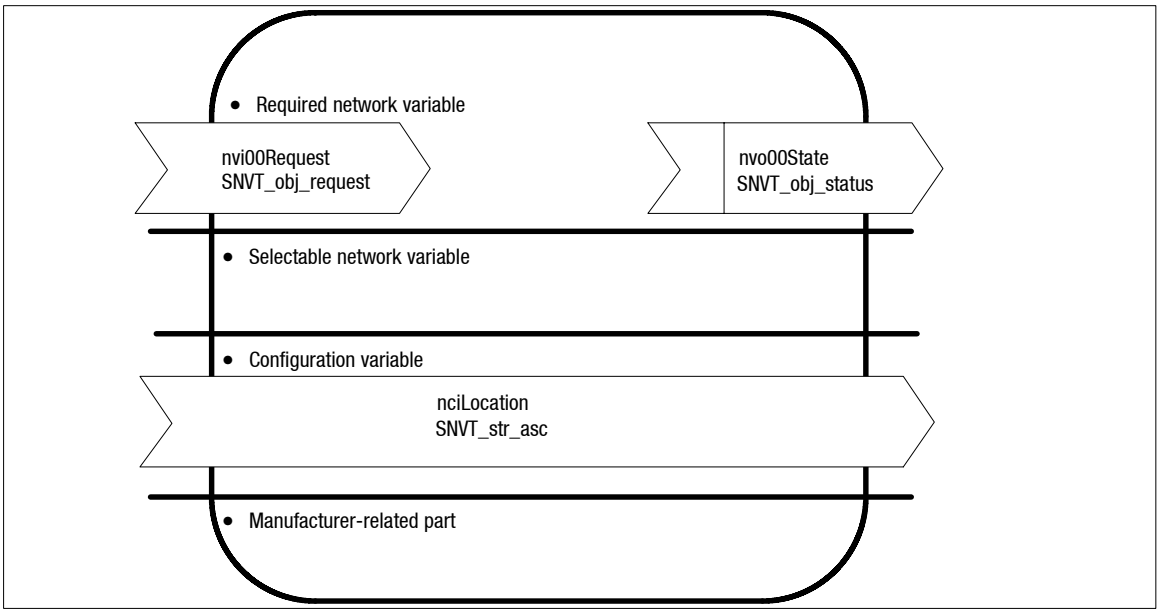
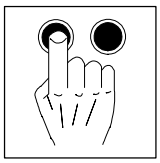



Fig. 6-1 Node Object, described in chapter 6.3





## 6.2.2 Overview: Network variable for VSD control

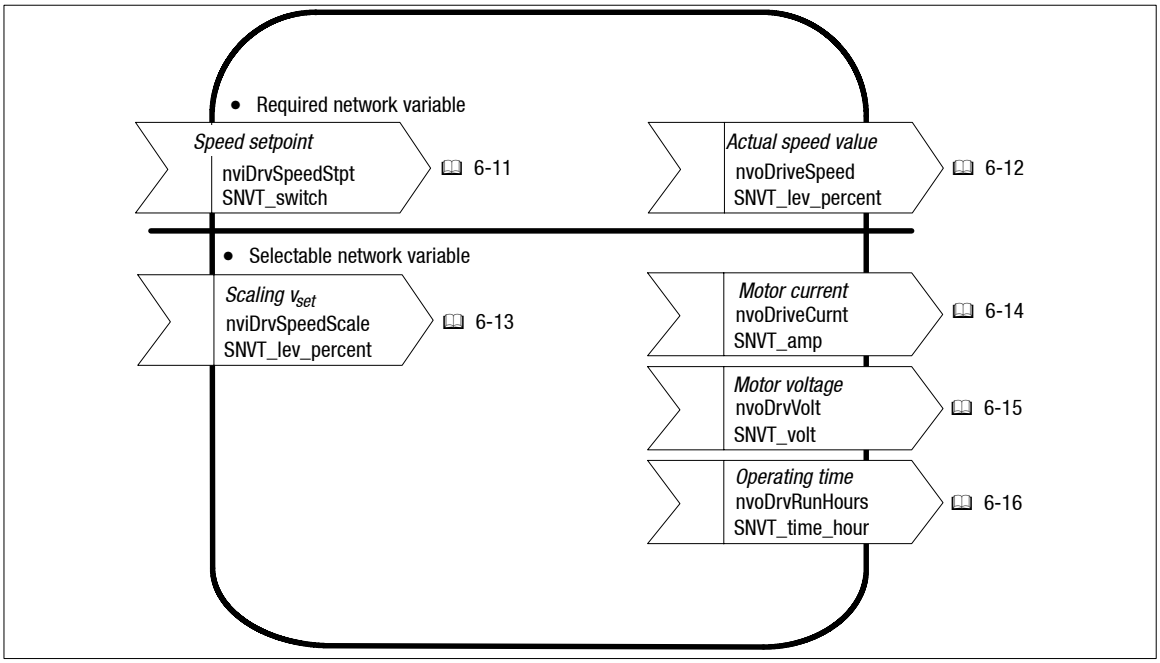



Fig. 6-2 Network variable VSD, described in chapter 6.4

## 6.2.3 Overview: Network variable for AIF-CTRL control

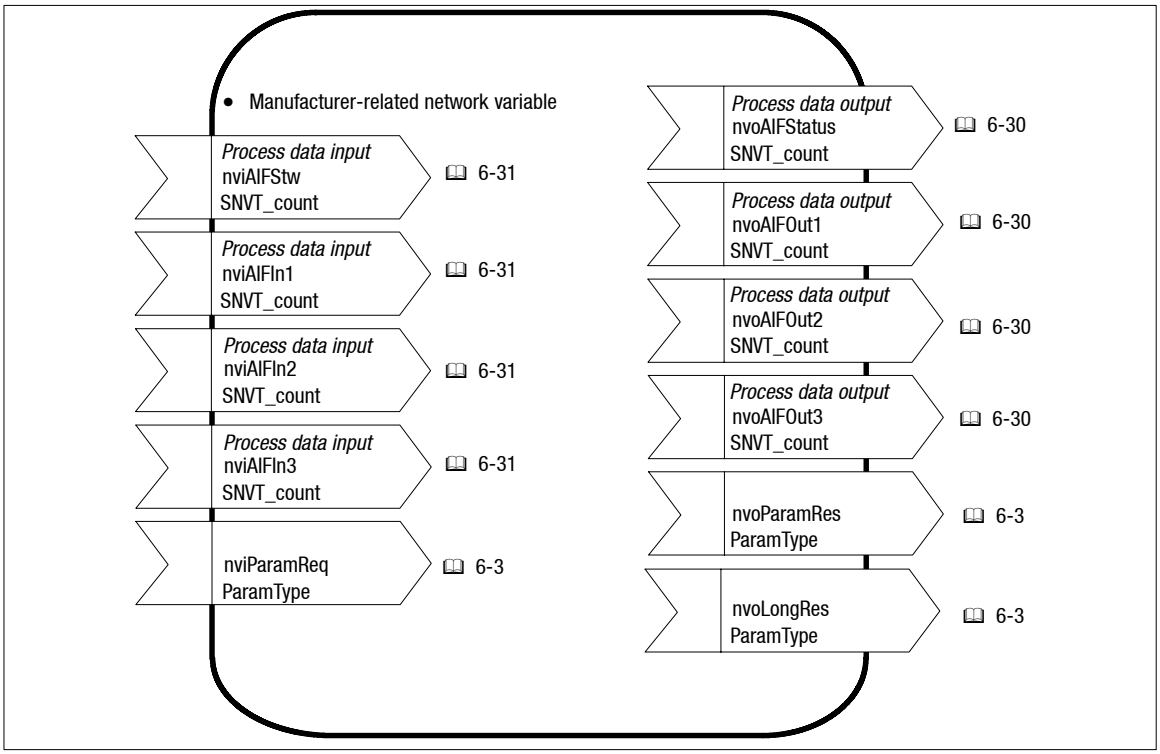



Fig. 6-3 Network variable AIF-CTRL, described in chapter 6.5

## 6.2.4 Overview: Configuration variable for VSD control

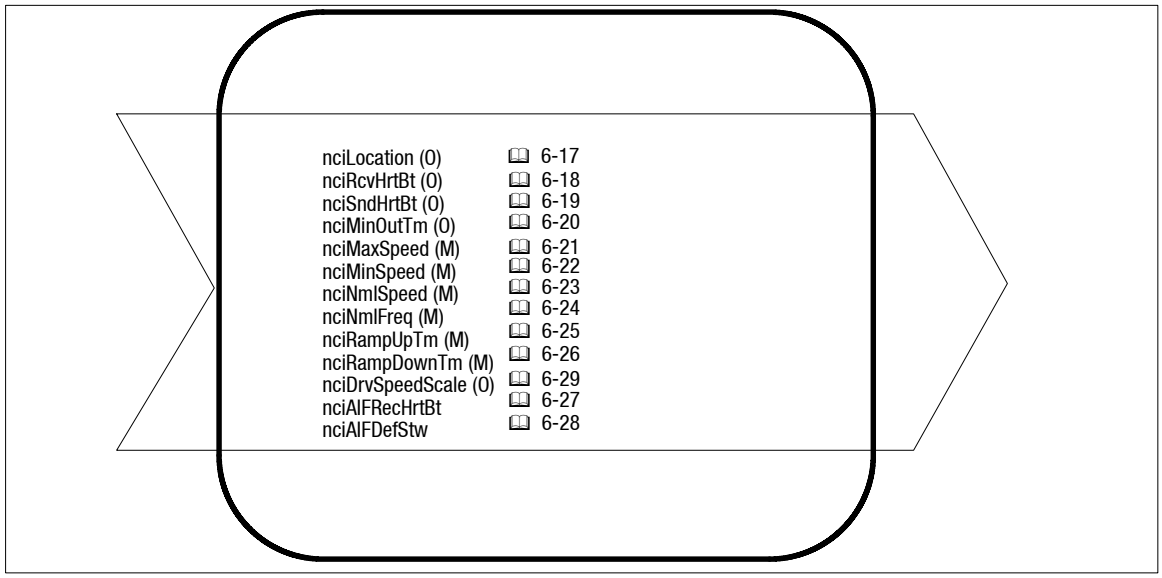
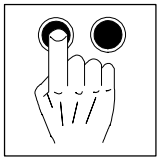



Fig. 6-4 Configuration variable VSD, described in chapter 6.4.2





## 6.3 Node Object

### 6.3.1 Inhibit / Enable all objects

All objects can be inhibited or enabled via the network variable `nvi00Request`:

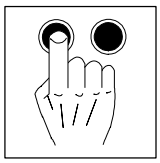
- Inhibit: `nvi00Request.object_id` = Object no.
  `nvi00Request.object_request` = 1 (RQ_DISABLED)
- Enable: `nvi00Request.object_id` = Object no.
  `nvi00Request.object_request` = 7 (RQ_ENABLED)

### 6.3.2 Status messages

In the event of a communication error between the LON interface module and the basic unit the bit comm_failure will be set under `nvo00State`.

Structure Type Definition:

```
typedef struct {
        unsigned long object_id;
        unsigned invalid_id : 1;
        unsigned invalid_request : 1;
        unsigned disabled : 1;
        unsigned out_of_limits : 1;
        unsigned open_circuits : 1;
        unsigned out_of_service : 1;
        unsigned mechanical fault : 1;
        unsigned feedback_failure : 1;
        unsigned over_range : 1;
        unsigned under_range : 1;
        unsigned electrical_fault : 1;
        unsigned unable_to_measure : 1;
        unsigned comm_failure : 1;
        unsigned fail_self_test : 1;
        unsigned self_test_in_progress : 1;
        unsigned locked_out : 1;
        unsigned manual_control: 1;
        unsigned in_alarm: 1;
        unsigned in_override: 1;
        unsigned report_mask: 1;
        unsigned programming_mode: 1;
        unsigned programming_fail: 1;
        unsigned alarm_notify_disabled: 1;
        unsigned reserved1: 1;
        unsigned reserved1: 8;
} SNVT_obj_status;
```



# 6.4 VSD control VSD

## 6.4.1 Network variable (VSD)

### 6.4.1.1 Speed setpoint

```
network input SNVT_switch nviDrvSpeedStpt;
```

The input network variable offers the following:

- Motor start/stop option
- Speed selection in percent
  (resolution = 0.5% ref. to value selected under `nciMaxSpeed`)

**Input range**

| Status | Value | Command | Explanation |
|---|---|---|---|
| 0 (FALSE) | NA | Stop | |
| 1 (TRUE) | 0 | 0% | |
| 1 (TRUE) | 1 to 200 | 0.5% to 100.0% | |
| 1 (TRUE) | 201 to 255 | 100.0% | |
| $jFF_{hex}$ | NA | AUTO (invalid) | Standard value.<br>This value will be set after a reset or if the time set under `nciRcvHrtBt` expires with an update of the variables.<br>AUTO has the same function as stop, but is also used to check whether the “Receive Heartbeat Time” has expired. |

**Function**

The speed setpoint for the controller is calculated as follows:

$$n_{soll} = \frac{\texttt{nviDrvSpeedStpt.value} \times \texttt{nviDrvSpeedScale}}{100}$$

The result is used as setpoint preselection in per cent. The setpoint selection is mapped to the AIF input data word 1 of the process data channel. Since 82XX and 93XX controllers expect different values here, the setpoints are converted in two different ways.

82XX controllers use a setpoint in Hz and a factor of 50 (24000 = 480Hz).

The value for AIF data word 1 is calculated as follows:

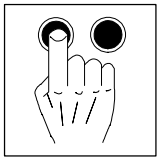
$$\text{AIFIn.W1} = \frac{\texttt{nviDrvSpeedStPt.value}}{100[\%] \times 2\ [\mathit{NVfactor}]} \times \frac{\texttt{nviDrvSpeedScale}}{100[\%] \times 200\ [\mathit{NVfactor}]} \times \frac{\texttt{nciNmlFreq}}{10\ [\mathit{NVfactor}]} \times 50\ [\mathit{AIFfactor}]$$

93XX servos use a setpoint in % and a factor of 163.83 (16383 = 100%).

The value for AIF input data word 1 results from the following:

$$\text{AIFIn.W1} = \frac{\texttt{nviDrvSpeedStPt.value}}{100[\%] \times 2\ [\mathit{NVfactor}]} \times \frac{\texttt{nviDrvSpeedScale}}{100[\%] \times 200\ [\mathit{NVfactor}]} \times 16383\ [\mathit{AIFfactor}]$$

If the result of the calculation is a negative value, the direction of rotation will be reversed. This is done by setting bit 2 in the AIF control word.





### 6.4.1.2 Actual speed value

```
network output SNVT_lev_percent nvoDrvSpeed;
```

The output network variable indicates the current speed as proportion of the rated value. Display value in [%].

#### Output range

The possible bandwidth for the output variables is between –163.840% and 163.830%. Resolution: 0.005%.

The value +163.835% ($7FFF_{hex}$) is assumed to be wrong.

#### Update rate

The variable is sent if the value changes or if the time set under *Maximum Send Time* (`nciSndHrtBt`) has expired.

The variable value will not be sent quicker than set in *Minimum Send Time* (`nciNinOutTm`).

#### Standard Service Type

The Standard Service Type is acknowledged.

#### Function

The value of `nvoDrvSpeed` results from the AIF output data word 1 of the process data channel. Since 82XX and 93XX send different value, the actual speed values are converted in different ways.

82XX controllers send actual values in [Hz] and a factor of 50 (24000 = 480Hz).

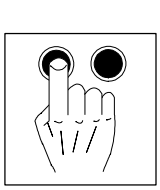
The value for `nvoDrvSpeed` results from:

$$\texttt{nvoDrvSpeed} = \frac{\text{AIFOut.W1}}{\frac{\texttt{nciNmlFreq}}{10\ [NVfactor]} \times 50[AIFfactor]} \times 100[\%] \times 200\ [NVfactor]$$

93XX controllers send actual values in [%] and a factor of 163.83 (16383 = 100%).

The value for `nvoDrvSpeed` results from:

$$\texttt{nvoDrvSpeed} = \frac{\text{AIFOut.W1}}{16383\ [NVfactor]} \times 100[\%] \times 200\ [NVfactor]$$



### 6.4.1.3 Scaling of the speed setpoint

```
network input SNVT_lev_percent nviDrvSpeedScale;
```

The setpoint can be scaled through this input network variable (`nviDrvSpeedStPt`).

Negative values result in the opposite direction of rotation.

Actual speed setpoint = `nviDrvSpeedStpt` × `nviDrvSpeedScale`

Example:

- `nviDrvSpeedStpt.value` = 50%
- `nviDrvSpeedScale` = –150%

The actual speed setpoint is –75% of the rated value.

#### Input range

The possible bandwidth for the input variables is between –163.840% and 163.830%. Resolution 0.005%.

The value +163.835% ($7FFF_{hex}$) is assumed to be wrong.

#### Standard value

The standard value is determined by the “Configuration Property” `nciDrvSpeedScale`.

This value will be set after a reset or if the time set under `nciRcvHrtBt` passes without an update of the variable.

#### Function

The value of `nviDrvSpeedStPt.value` is multiplied with the value of `nviDrvSpeedScale` and divided by 100.

The result is used as setpoint preselection in per cent. The setpoint selection is mapped to the AIF input data word 1 of the process data channel. Since 82XX and 93XX controllers expect different values here, the setpoints are converted in two different ways.

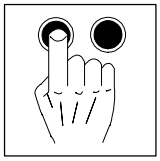
82XX controller expect the setpoint in [Hz] and a factor of 50 (24000 = 480Hz). The value for AIF data word 1 is calculated as follows:

$$\text{AIFIn.W1} = \frac{\texttt{nviDrvSpeedStPt.value}}{100[\%] \times 2\ [\mathit{NVfactor}]} \times \frac{\texttt{nviDrvSpeedScale}}{100[\%] \times 200\ [\mathit{NVfactor}]} \times \frac{\texttt{nciNmlFreq}}{10\ [\mathit{NVfactor}]} \times 50\ [\mathit{AIFfactor}]$$

93XX controllers expect the setpoint in [%] and a factor of 163.83 (16383 = 100%). The value for AIF input data word 1 results from the following:

$$\text{AIFIn.W1} = \frac{\texttt{nviDrvSpeedStPt.value}}{100[\%] \times 2\ [\mathit{NVfactor}]} \times \frac{\texttt{nviDrvSpeedScale}}{100[\%] \times 200\ [\mathit{NVfactor}]} \times 16383\ [\mathit{AIFfactor}]$$

If the result of the calculation is a negative value, the direction of rotation will be reversed. This is done by setting bit 2 in the AIF control word.





### 6.4.1.4 Motor current

```
network output SNVT_amp nvoDriveCurnt;
```

This network variable outputs the actual output current in Ampere.

#### Output range

The possible bandwidth for the output variable is between –3276.8 A and +3276.6 A. Resolution 0.1 A.

The value +3276.7 A ($7FFF_{hex}$) is assumed to be wrong.

#### Update rate

The variable is sent if the value has changed or if the time set under *Configuration Property Maximum Send Time* (`nciSndHrtBt`) has expired.

The variable value will not be sent quicker than set in *Configuration Property Minimum Send Time* (`nciMinOutTm`).

#### Standard Service Type

The Standard Service Type is acknowledged.

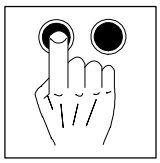
#### Function

The output current is cyclically read from the drive

(code 0054, subcode 00). The value is calculated as follows:

$$\texttt{nvoDriveCurnt} = \frac{\text{SDF.LW}}{10000\ [\textit{AIFfactor}]} \times 10\ [\textit{NVfactor}]$$

If an error occurs during the query, *nvoDriveCurnt* will be set to 7FFFhex.



### 6.4.1.5 Motor voltage

```
network output SNVT_volt nvoDrvVolt;
```

The output network variable indicates the current output voltage in [Volt].

#### Output range

The possible bandwidth for the output variable is between –3276.8 V and +3276.6 V. Resolution 0.1 V.

The value +3276.7 V ($7FFF_{hex}$) is assumed to be wrong.

The typical output range is between 0 V and 700 V.

#### Update rate

The variable is sent if the value has changed.

The variable is sent as heart beat if the time set under *Configuration Property Maximum Send Time* (`nciSndHrtBt`) has expired.

The variable value will not be sent quicker than set in *Configuration Property Minimum Send Time* (`nciMinOutTm`).

#### Function

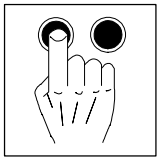
The current output voltage is cyclically read by the drive (code 0052, subcode 00). The value is calculated as follows:

$$\texttt{nvoDriveVolt} = \frac{\text{SDF.LW}}{10000\ [\textit{AIFfactor}]} \times 10\ [\textit{NVfactor}]$$

If an error occurs during the query, `nvoDrvVolt` is set to $7FFF_{hex}$.

#### Standard Service Type

The Standard Service Type is acknowledged.





### 6.4.1.6 Operating time

```
network output SNVT_time_hour nvoDrvRunHours;
```

This network variable output the entire operating time in hours.

**Output range**

The possible bandwidth for the output variable is between 0 and 65534 hours. Resolution: 1 hour.

The value of 65535 hours ($FFFF_{hex}$) is assumed to be wrong.

**Update rate**

The variable is sent if the value has changed.

The variable value will not be sent quicker than set im *Configuration Property Minimum Send Time* (`nciMinOutTm`).

**Standard Service Type**

The Standard Service Type is acknowledged.

**Function**

The operating time is cyclically read by the drive (code 0174, subcode 00).

82XX controllers send the operating time in hours with the AIF factor 10000 (30000 = 3 hours).

The value is calculated as follows:

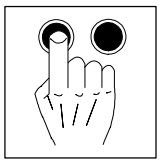
$$\texttt{nvoDriveRunHours} = \frac{\text{SDF.LW}}{10000\ [\textit{AIFfactor}]} \times 1\ [\textit{NVfactor}]$$

93XX controllers send the operating time in seconds without AIF factor.

The value is calculated as follows:

$$\texttt{nvoDriveRunHours} = \frac{\text{SDF.LW}}{3600\ [\frac{\textit{seconds}}{\text{hour}}]} \times 1\ [\textit{NVfactor}]$$

If an error occurs during the query, `nvoDrvRunHours` is set to $FFFF_{hex}$.



## 6.4.2 Configuration variable (VSD)

### 6.4.2.1 Location label

```
network input config SNVT_str_asc nciLocation;
```

This configuration variable can be used to store a 31-byte string.

**Input range**

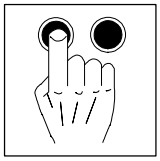
A ZERO terminated string of max. 31 byte

**Standard value**

ZERO

**SCPT Reference**

SCPT_location (17)





### 6.4.2.2 Receive Heartbeat Time

```
network input config SNVT_time_sec nciRcvHrtBt;
```

This configuration variable is used to store the time expected for an update, if this time expires without an update, the values for the following network variables will be reset to their default settings:

`nviDrvSpeedStpt` (📖 6-11)

`nviDrvSpeedScale` (📖 6-13)

It is also used to monitor the variables configured in `nciAIFRecHrtBt` (📖 6-27). The variables are: `nviAIFStw, nviAIFIn1, nviAIFIn2, nviAIFIn3`.

**Input range**

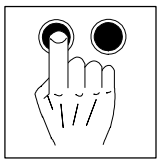
The possible bandwidth for the configuration variable is between 0.0 and 6553.4 seconds (resolution 0.1 seconds)

**Standard value**

0.0 (no monitoring)

**SCPT Reference**

SCPTmaxRcvTime (48)



### 6.4.2.3 Maximum Send Time

```
network input config SNVT_time_sec nciSndHrtBt;
```

This configuration variable stores the time needed before the following network variables will be updated:

```
nvoDrvSpeed

nvoDrvCurnt

nvoDrvVolt

nvoAIFStw

nvoAIFOut1

nvoAIFOut2

nvoAIFOut3
```

**Input range**

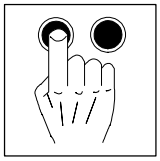
The possible bandwidth for the configuration variable is between 0.0 and 6553.4 seconds (resolution 0.1 seconds)

**Standard value**

0.0 (no automatic update)

**SCPT Reference**

SCPTmaxSendTime(49)





### 6.4.2.4 Minimum Send Time

```
network input config SNVT_time_sec nciMinOutTm;
```

Here you store the time needed before an output network variable can be updated.

**Input range**

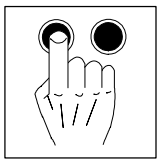
The possible bandwidth for the configuration variable is between 0.1 and 6553.4 seconds (resolution 0.1 seconds)

**Standard value**

0.5 seconds

**SCPT Reference**

SCPTminSendTime (52)



### 6.4.2.5 Maximum speed

```
network input config SNVT_lev_percent nciMaxSpeed;
```

The configuration variable contains the maximum motor speed. The value is indicated as percentage of the rated value (`NciNmlSpeed`).

#### Input range

The possible bandwidth for the configuration variable is between –163.840% and 163.830% (resolution 0.005%). The value +163.835% ($7FFF_{hex}$) is assumed to be wrong.

The entered value will be checked by means of the following formula:

$$-163.840\% \leq \text{Minimum speed} \leq \text{Maximum speed} \leq 163.830\%$$

If the result is wrong, the lower value is taken as maximum speed.

#### Standard value

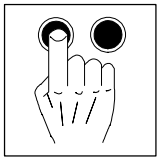
100.000 %

#### Function

After a change of the maximum speed, the value will be sent to the drive via the parameter channel (code 0011, subcode 00).

#### SCPT Reference

SCPTmaxSetpoint (50)





### 6.4.2.6 Minimum speed

```
network input config SNVT_lev_percent nciMinSpeed;
```

The configuration variable contains the minimum motor speed. The value is indicated as percentage of the rated value (`nciNmlSpeed`).

**Input range**

The possible bandwidth for the configuration variable is between –163.840% and 163.830% (resolution 0.005%). The value +163.835% ($7FFF_{hex}$) is assumed to be wrong.

The entered value will be checked by means of the following formula:

$$-163.840\% \leq \text{Minimum speed} \leq \text{Maximum speed} \leq 163.830\%$$

If the result is wrong, the lower value is taken as minimum speed.

**Standard value**

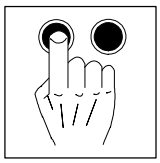
0.000 %

**Function**

After a change of the minimum speed, the value will be sent to the drive via the parameter channel (code 0010, subcode 00).

**SCPT Reference**

SCPTminSetpoint (53)



### 6.4.2.7 Rated speed value

```
network input config SNVT_rpm nciNmlSpeed;
```

This configuration variable contains the rated motor speed in rpm.

**Input range**

The possible bandwidth for the configuration variable is between 0 and 65534 rpm (resolution 1 rpm)

**Standard value**

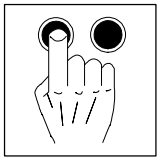
3000 rpm

**Function**

After a change of the rated speed, the value will be sent to the drive via the parameter channel (code 0087, subcode 00).

**SCPT Reference**

SCPTnomRPM (158)





### 6.4.2.8 Rated frequency

```
network input config SNVT_freq_hz nciNmlFreq;
```

This configuration variable contains the rated motor frequency in Hz.

#### Input range

The possible bandwidth for the configuration variable is between 0 and 6553,4 Hz (resolution 0,1 Hz)

**Standard value**

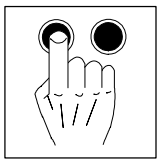
50 Hz

**Function**

After a change of the rated frequency, the value will be sent to the drive via the parameter channel (code 0089, subcode 00).

**SCPT Reference**

SCPTnomFreq (159)



### 6.4.2.9 Acceleration time

```
network input config SNVT_time_sec nciRampUpTm;
```

This configuration variable indicates the acceleration time of the motor.

**Input range**

The possible bandwidth for the configuration variable is between 0 and 6553.4 seconds (resolution 0,1 Sekunden)

**Standard value**

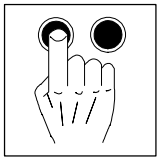
5 seconds

**Function**

After a change of the acceleration time, the value will be sent to the drive via the parameter channel (code 0012, subcode 00).

**SCPT Reference**

SCPTrampUpTm (160)





### 6.4.2.10 Deceleration time

```
network input config SNVT_time_sec nciRampDownTm;
```

This configuration variable indicates the deceleration time of the motor.

**Input range**

The possible bandwidth for the configuration variable is between 0 and 6553.4 seconds (resolution 0,1 Sekunden)

**Standard value**

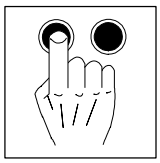
5 seconds

**Function**

After a change of the deceleration time, the value will be sent to the drive via the parameter channel (code 0013, subcode 00).

**SCPT Reference**

SCPTrampDownTm (161)



### 6.4.2.11 Receive heartbeat for AIF-CTRL

```
network input config SNVT_state nciAIFRecHrtBt;
```

Here you can select the manufacturer-specific network variables (nviAIFStw, nviAIFIn1, nviAIFIn2, nviAIFIn3) which are to be monitored with a *Receive Timeout*.

nciAIFRecHrtBt.bit0 = 0 – No timeout monitoring for nviAIFStw

nciAIFRecHrtBt.bit0 = 1 – Timeout monitoring for nviAIFStw

nciAIFRecHrtBt.bit1 = 0 – No timeout monitoring for nviAIFIn1

nciAIFRecHrtBt.bit1 = 1 – Timeout monitoring for nviAIFIn1

nciAIFRecHrtBt.bit2 = 0 – No timeout monitoring for nviAIFIn2

nciAIFRecHrtBt.bit2 = 1 – Timeout monitoring for nviAIFIn2

nciAIFRecHrtBt.bit3 = 0 – No timeout monitoring for nviAIFIn3

nciAIFRecHrtBt.bit3 = 1 – Timeout monitoring for nviAIFIn3

If one of the monitored variables gets a *Receive Heartbeat Timeout*, the function selected in nciAIFDefStw will be carried out.

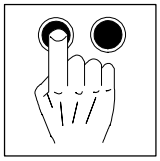
**Input range**

The possible range for the input variable is 0 or 1 for every bit.

Only bits 0 and 3 will be evaluated.

**Standard value**

0





### 6.4.2.12 Monitoring reaction for AIF-CTRL

```
network input config SNVT_state nciAIFDefStw;
```

If one of the monitored variables of the AIF-CTRL gets a Receive Heartbeat Timeout (📖 6-30), the function selected in `nciAIFDefStw` will be carried out.

The quick stop functions sets bit 3 in the AIF control word, the controller inhibit function sets bit 9 in the AIF control word.

**Tip!**

The functions quick stop and controller inhibit will only be reset after all monitored variables have been updated within the Receive Heartbeat Timeout.

**Input range**

The possible bandwidth for the configuration variables is 0, 1 oooder 2

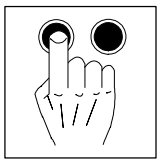
`nciAIFDefStw` = 0 – No reaction

`nciAIFDefStw` = 1 – Quick stop

`nciAIFDefStw` = 2 – Controller inhibit

**Standard value**

0



### 6.4.2.13 Default value for nviDrvSpeedScale

```
network input config_lev_percent nciDrvSpeedScale;
```

This value determines the default setting for `nciDrvSpeedScale`. The current configuration variable `nciDrvSpeedScale` is overwritten with the default value

- after a reset or
- if the time from `nciRcvHrtBt` passes without an update of the variable `nciDrvSpeedScale`.

**Input range**

The possible bandwidth for the input variable is between -163.840 % and 163.830 % (resolution 0.005 %).

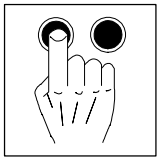
The value +163.835 % ($7FFF_{hex}$) is assumed to be wrong.

**Standard value**

0

**SCPT Reference**

SCPTdefScale (162)





# 6.5 Device control AIF-CTRL

## 6.5.1 General information

### 6.5.1.1 Output variable

```
network output SNVT_count nvoAIFStatus;

network output SNVT_count nvoAIFOut1;

network output SNVT_count nvoAIFOut2;

network output SNVT_count nvoAIFOut3;
```

The output network variable `nvoAIFStatus` is directly taken from the AIF status word.

The output network variables `nvoAIFOut1` to `nvoAIFOut3` are directly taken from the corresponding AIF data words.

**Output range**

The possible bandwidth for the output variable is between 0 and 65535.

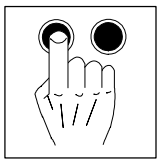
**Update range**

The variables are sent if the value changes or if the time set under *Configuration Property Maximum Send Time* (`nciSndHrtBt`) has expired.

The variable value will not be sent quicker than set in *Configuration Property Minimum Send Time* (`nciMinOutTm`).

**Standard Service Type**

The Standard Service Type is acknowledged.



### 6.5.1.2 Input variable

```
network input SNVT_count nviAIFStw;

network input SNVT_count nviAIFIn1;

network input SNVT_count nviAIFIn2;

network input SNVT_count nviAIFIn3;
```

The input network variable `nviAIFStw` is directly accepted by the AIF control word.

The input network variables `nviAIFIn1` to `nviAIFIn3` are directly accepted by the corresponding AIF input data words.

**1Input range**

The possible bandwidth for the output variable is between 0 and 65535.

**Standard value**

The standard value after a reset is 0 for all four variables.

The standard value for `nviAIFStw` is determined by the configuration properties `nciAIFDefStw` and `nciAIFRecHrtBt`. `nciAIFDefStw` indicates the function to be executed in the event of a *Receive Heartbeat Timeout* and `nciAIFRecHrtBt` indicates the variables monitored by a *Receive Heartbeat*.

`nciAIFRecHrtBt.bit0` = 1 – Timeout monitoring for `nviAIFStw`

`nciAIFRecHrtBt.bit1` = 1 – Timeout monitoring for `nviAIFIn1`

`nciAIFRecHrtBt.bit2` = 1 – Timeout monitoring for `nviAIFIn2`

`nciAIFRecHrtBt.bit3` = 1 – Timeout monitoring for `nviAIFIn3`

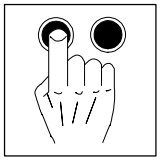
`nciAIFStw` = 0 – No reaction

`nciAIFStw` = 1 – Quick stop

`nciAIFStw` = 2 – Controller inhibit

If one of the monitored variables gets a *Receive Heartbeat Timeout*, the function selected in `nciAIFDefStw` will be carried out.

The quick stop functions sets bit 3 in the AIF control word, the controller inhibit function sets bit 9 in the AIF control word. The functions quick stop and controller inhibit will only be reset after all monitored variables have been updated within the Receive Timeouts.





## 6.5.2 Network variable for 82XX controllers

### Setpoint source selection

The setpoint source for these controllers is selected under code C0001. An evaluation of process data is only possible if code C0001 is set to ”3” when the controller is operated together with the fieldbus module.

Therefore the process data channel serves as setpoint source, describing the frequency setpoint (C0046) and the control word (parameter channel, C0135) (see Operating Instructions 82XX).

**Tip!**

Please observe that code C0001 is available in all parameter sets.

### Network variable <u>from</u> drive

- `nvoAIFStatus` reads the status word (📖 6-38)

| Status word | |
|---|---|
| LOW byte | HIGH byte |
| Bits 0 to 7 of the status word (code C0150) are entered here. | Bits 8 to 15 of the status word (code C0150) are entered here. |

- `nvoAifOut1` reads the actual frequency value

| Actual frequency value | |
|---|---|
| LOW byte | HIGH byte |
| The actual frequency value is read from code C0050. The actual frequency value with the signed normalisation 24000 = 480 Hz is provided here. | |

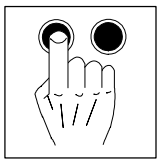
### Network variable <u>to</u> drive

- `nviAIFStw` writes the control word (📖 6-39)

| Control word | |
|---|---|
| LOW byte | HIGH byte |
| Bits 0 to 7 of the control word (code C0135) are entered here. | Bits 8 to 15 of the control word (code C0135) are entered here. |

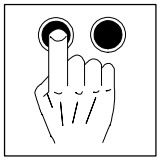
- `nviAIFIn1` writes the frequency setpoint

| Frequency setpoint | |
|---|---|
| LOW byte | HIGH byte |
| The frequency setpoint (code C0046 via parameter data channel) is selected as process data word. The normalization differs from the setting under C0046. It is a signed value, 24000 = 480 Hz. | |



### 6.5.2.1 Status word for 82XX

| Bit | 820X | 821x,822x |
|---|---|---|
| **00** | 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 |
| **01** | 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 |
| **02** | $I_{max}$ (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 |
| **03** | not assigned | $f_d = f_{dset}$<br>$0 = f_d \neq f_{dset}$<br>$1 = f_d = f_{dset}$ |
| **04** | $f_d = f_{dset}$<br>0: $f_d \neq f_{dset}$<br>1: $f_d = f_{dset}$ | RFG on = RFG off<br>0 = RFG on ≠ RFG off<br>1 = RFG on = RFG out |
| **05** | Qmin ($f_d \leq f_{dQmin}$)<br>0 = Qmin not active<br>1 = Qmin active | Qmin ($f_d \leq f_{dQmin}$)<br>0 = Qmin not active<br>1 = Qmin active |
| **06** | $f_d = 0$ (act. frequency = 0)<br>0: $f_d \neq 0$<br>1: $f_d > 0$ | $f_d > 0$ (actual frequency value = 0)<br>$0 = f_d \neq 0$<br>$1 = f_d > 0$ |
| **07** | 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 |
| **08...11** | Controller status<br>0 = Controller initialization<br>8 = Error active | Controller status<br>0 = Controller initialization<br>1 = 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 |
| **13** | $U_{Gmax}$ (DC-bus overvoltage)<br>0 = No overvoltage<br>1 = overvoltage | $U_{Gmax}$ (DC-bus overvoltage)<br>0 = No overvoltage<br>1 = overvoltage |
| **14** | Direction of rotation<br>0 = CW rotation<br>1 = CCW rotation | Direction of rotation<br>0 = CW rotation<br>1 = CCW rotation |
| **15** | Ready<br>0 = not ready for operation<br>1 = ready for operation | Ready<br>0 = not ready for operation<br>1 = ready for operation |



# Parameter setting

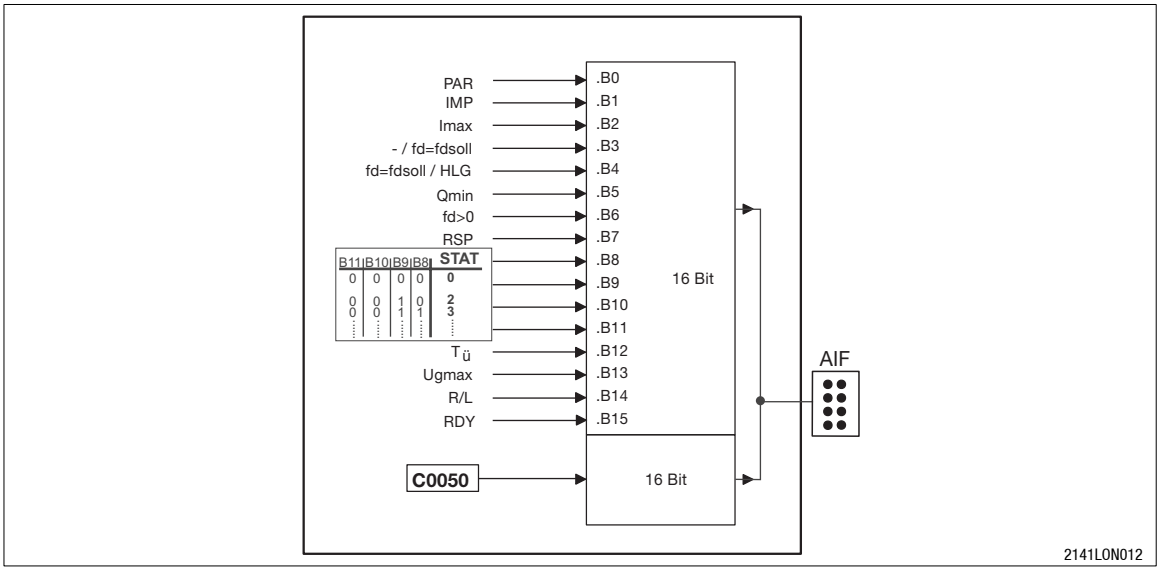
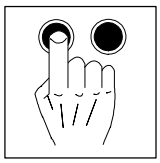



Fig. 6-5 Read access to status word and actual frequency in 82XX (fixed assignment)



### 6.5.2.2 Control word for 82XX

| Bit | 820X | 821x,822x |
|---|---|---|
| **00, 01** | 00 = C0046 active<br>01 = JOG1 active in C0037<br>10 = JOG2 active in C0038<br>11 = JOG3 active in C0039 | 00 = C0046 active<br>01 = JOG1 active in C0037<br>10 = JOG2 active in C0038<br>11 = JOG3 active in C0039 |
| **02** | 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 |
| **03** | QSP (quick stop)<br>0 = QSP not active<br>1 = QSP active | QSP (quick stop)<br>0 = QSP not active<br>1 = QSP active |
| **04** | Reserved | RFG stop (stop of the ramp function generator)<br>0 = RFG stop not active<br>1 = RFG stop active |
| **05** | Reserved | RFG zero (deceleration along the $T_{if}$ ramp C0013)<br>0 = RFG zero not active<br>1 = RFG zero active |
| **06** | Reserved | UP function for motor potentiometer<br>0 = UP not active<br>1 = UP active |
| **07** | Reserved | DOWN function for motor potentiometer<br>0 = DOWN not active<br>1 = DOWN active |
| **08** | Reserved | Reserved |
| **09** | 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 |
| **10** | Reserved | Reserved |
| **11** | Reserved | 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 |
| **13** | Reserved | Reserved |
| **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 |
| **15** | Reserved | Reserved |

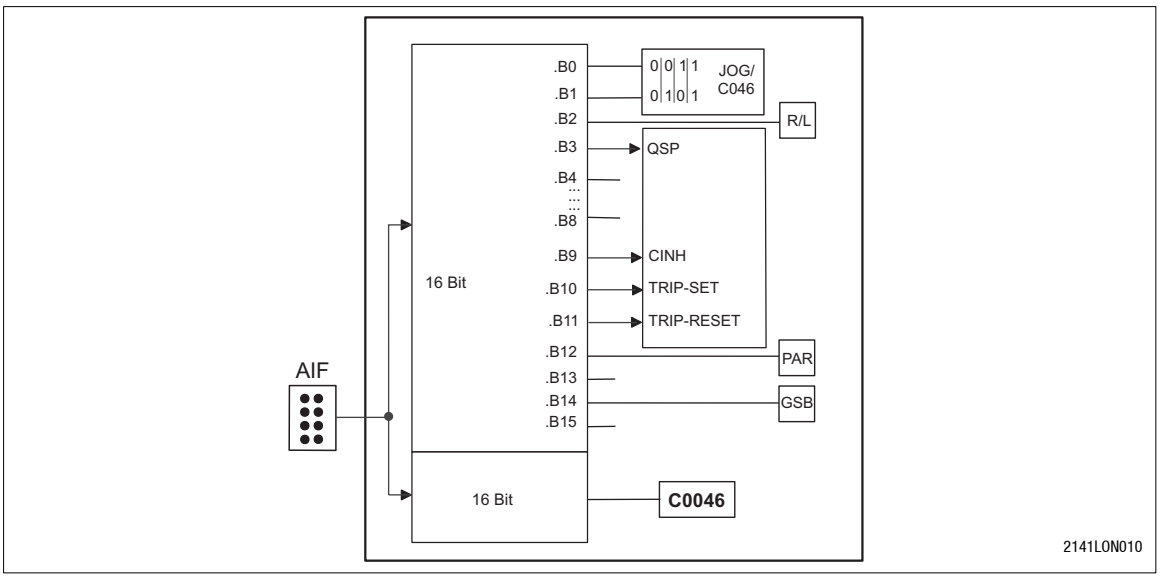
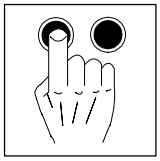



Fig. 6-6 Access to control word and actual frequency in 82XX (fixed assignment)





## 6.5.3 Network variable for 8200 vector controllers

**Setpoint source selection**

The setpoint source for these controllers is selected under code C0001. An evaluation of process data is only possible if code C0001 is set to ”3” when the controller is operated together with the fieldbus module. (Selection: process data channel of a fieldbus module AIF-IN.W1 or AIF-IN.W2).

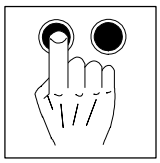
The process data channel which describes the frequency setpoint (C0046) and the control word (parameter channel, C0135) is the setpoint source (see 8200 vector Operating Instructions).

Check under C0412/x whether the setpoint source is assigned to the signal wanted.

**Tip!**

Please ensure that code C0001 or C0410 is the same for all parameter sets.

Digital and analog input and output signals can be freely configured (see Operating Instructions for 8200 vector; codes C0410, C0412, C0417 and C0421) The change of the code C0001 to 3 starts the preconfiguration of the process data words in the controller.



**Network variable <u>from</u> drive** (see function block AIF-OUT (📖 6-38) )

- `nvoAIFStatus` reads bits B0 ... B15 of the status word AIF-STAT (📖 6-38)

| Status word AIF-STAT | |
|---|---|
| LOW byte | HIGH byte |
| Bits 0 to 7 of the status word will be entered here. | Bits 8 to 15 of the status words will be entered here. |

- `nvoAifOut1` reads the output word AIF-OUT.W1

| Actual value | |
|---|---|
| LOW byte | HIGH byte |
| The actual frequency value with the signed normalisation 24000 = 480 Hz is provided here.<br>Torque values are normalised with 16384 = 100%. AIF-OUT.W1 see C0421/1, (Operating Instructions for 8200 vector) | |

- `nvoAifOut2` reads the output word AIF-OUT.W2

| Actual value | |
|---|---|
| LOW byte | HIGH byte |
| The actual frequency value with the signed normalisation 24000 = 480 Hz is provided here.<br>Torque values are normalised with 16384 = 100%. AIF-OUT.W2 see C0421/2, (Operating Instructions for 8200 vector) | |

**Network variable <u>to</u> drive** (see function block AIF-IN (📖 6-40))

- `nviAIFStw` writes bits B0 ... B15 in the control word AIF-CTRL (📖 6-39)

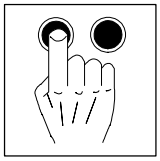
| Control word AIF-CTRL | |
|---|---|
| LOW byte | HIGH byte |
| Bits 0 to 7 of the control word will be entered here. | Bits 8 to 15 of the control word will be entered here. |

- `nviAIFIn1` writes the input word AIF-IN.W1

| AIF-IN.W1 | |
|---|---|
| LOW byte | HIGH byte |
| The frequency setpoint is preselected as process data word. The normalisation differs from the setting under C0046. It is a signed value, 24000 = 480 Hz. AIF-IN.W1 see C0412/x, Wert “10” (Operating Instructions for 8200 vector) | |

- `nviAIFIn2` writes the input word AIF-IN.W2

| AIF-IN.W2 | |
|---|---|
| LOW byte | HIGH byte |
| The frequency setpoint is preselected as process data word.<br>The normalisation differs from the setting under C0046. It is a signed value, 24000 = 480 Hz.<br>AIF-IN.W2 see C0412/x, Wert “11” (Operating Instructions for 8200 vector) | |





### 6.5.3.1 Status word for 8200 vector

| Bit | 8200vector default setting |
|---|---|
| **00** | DCTRL-PAR-B0 |
| **01** | DCTRL1-IMP |
| **02** | MCTRL1-IMAX |
| **03** | MCTRL1-RFG1=NOUT |
| **04** | NSET1-RFG1-I=0 |
| **05** | PCTRL1-QMIN |
| **06** | DCTRL1-NOUT=0 |
| **07** | DCTRL1-CINH |
| **08...11** | Controller status<br>0 = Controller initialization<br>1 = 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** | DCTRL1-OH-WARN |
| **13** | DCTRL1-OV |
| **14** | DCTRL1-CCW |
| **15** | DCTRL1-RDY |

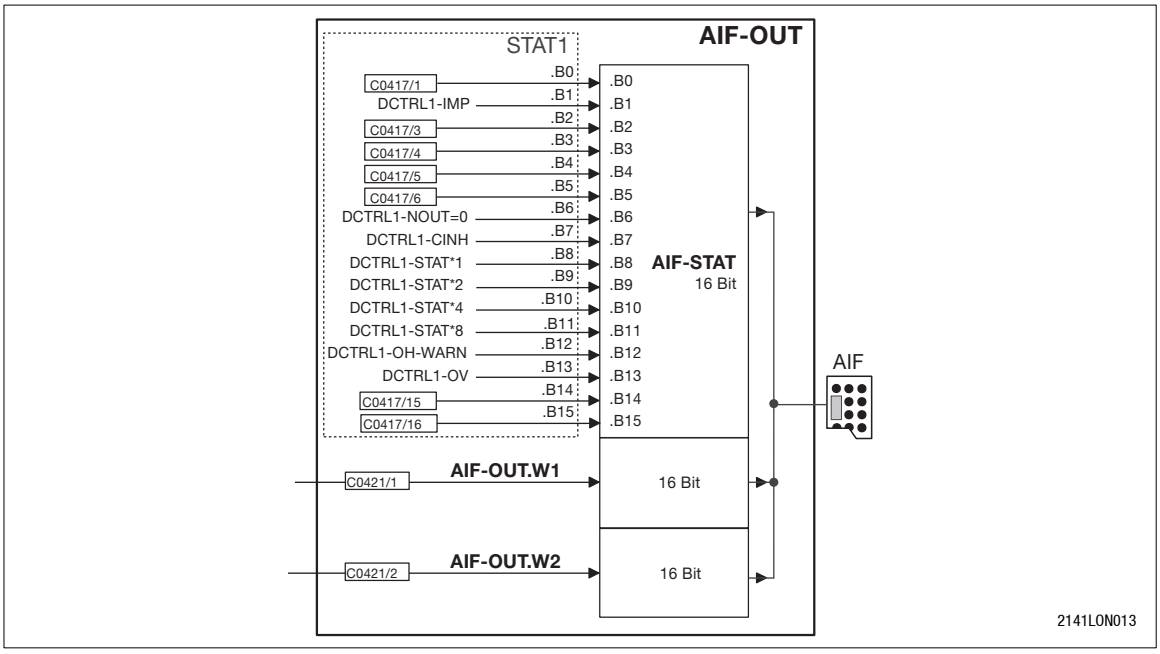
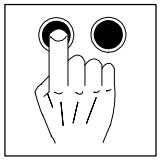
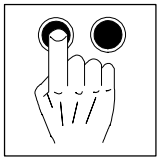



Fig. 6-7 Function block AIF-OUT in 8200 vector



## 6.5.3.2 Control word for 8200 vector

| Bit | 8200vector | |
|---|---|---|
| | **Default setting: C0001=3 if C0007 < 52** | **Default setting: C0001=3 if C0007 > 51** |
| **00, 01** | 00 = C0046 active<br>01 = NSET1-JOG1 (C0037) active<br>10 = NSET1-JOG2 (C0038) active<br>11 = NSET1-JOG3 (C0039) active | Freely configurable by user |
| **02** | DCTRL1-CW/CCW<br>0 = not active<br>1 = active | |
| **03** | AIF-CTRL-QSP<br>0 = not active<br>1 = active | |
| **04** | NSET1-RFG1-STOP<br>0 = not active<br>1 = active | Freely configurable by user |
| **05** | NSET1-RFG1-0<br>0 = not active<br>1 = active | |
| **06** | MPOT1-UP<br>0 = not active<br>1 = active | |
| **07** | MPOT1-DOWN<br>0 = not active<br>1 = active | |
| **08** | Freely configurable by user | |
| **09** | AIF-CTRL-CINH<br>0 = not active<br>1 = active | |
| **10** | AIF-CTRL-TRIP-SET<br>0 = not active<br>1 = active | |
| **11** | AIF-CTRL-TRIP-RESET<br>0 -> 1 = Edge from 0 to 1 | |
| **12** | DCTRL1-PAR2/4<br>0 = not active<br>1 = active | Freely configurable by user |
| **13** | DCTRL1-PAR3/4<br>0 = not active<br>1 = active | |
| **14** | MCTRL1-DCB<br>0 = not active<br>1 = active | |
| **15** | Freely configurable by user | |



# Parameter setting

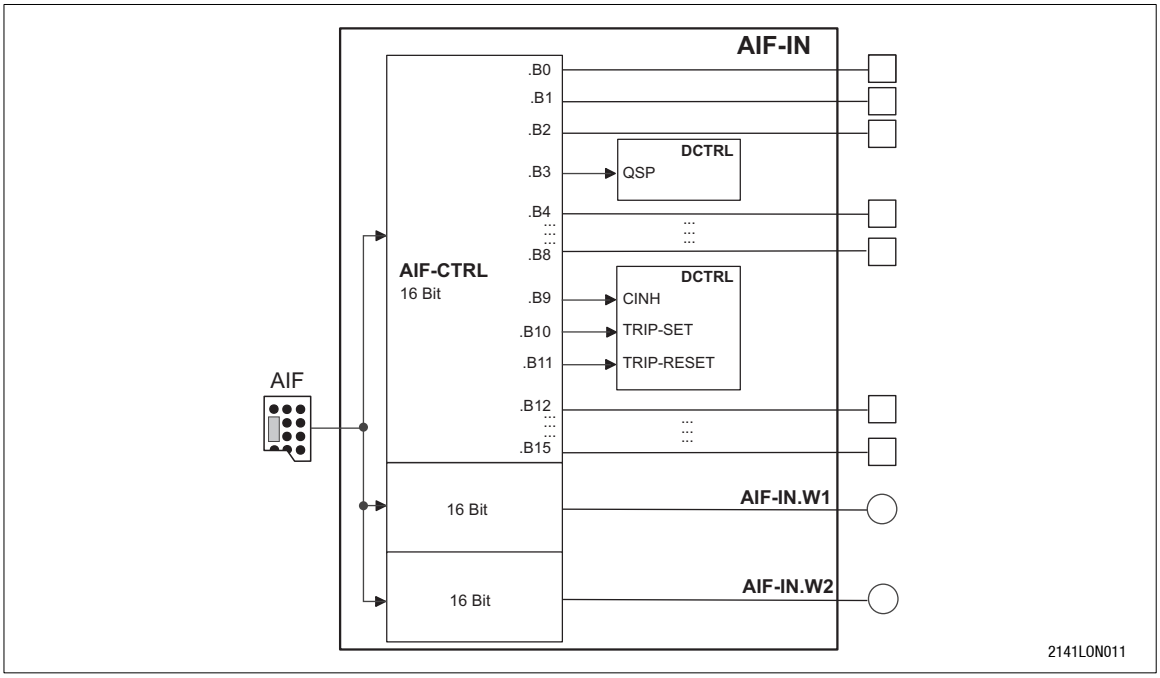
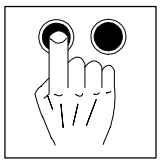



Fig. 6-8 Function block AIF-IN in 8200 vector



## 6.5.4 Network variable for 93XX controllers

### Setpoint source selection

With 93XX controllers it is not possible to select a setpoint source by means of one code only. It is however easy to adapt the controller to the drive task because it is working with preconfigured function blocks which only have to be connected. For this it is not necessary to have special programming skills.

The user himself can link the function blocks. It is however easier to use Lenze preconfigurations which are already stored in the controller ROM. The Lenze preconfigurations (see code C0005) determine the source (terminal, keypad, fieldbus module) which describes the frequency setpoint and the control word.

The value in code C0005 must be set to "xxx3" for LON bus operation (x = selected preconfiguration).

Fore more information please see the corresponding Manual and Operating Instructions for the controller.

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

### Network variable <u>from</u> drive

- `nvoAIFStatus` = STAT.B0 ... STAT.B15 (status word)

| Status word | |
|---|---|
| LOW byte | HIGH byte |
| Bits 0 to 7 of the status word will be entered here. | Bits 8 to 15 of the status word will be entered here (see chapter 6.5.4.1). The description of the bits can be obtained from the Code Table. |

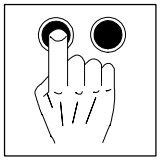
- `nvoAifOut1` = AIF-OUT.W1
- `nvoAifOut2` = AIF-OUT.W2
- `nvoAifOut3` = AIF-OUT.W3

| AIF-OUT.Wx | |
|---|---|
| LOW byte | HIGH byte |
| The actual frequency value with the signed normalisation 24000 = 480 Hz is provided here. | |

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.

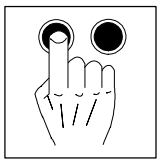
The function block AIF-OUT determines the controller output data as data interface for the 2141 fieldbus module.



# Parameter setting

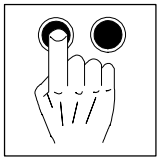
| Signal configuration (L-C0005) | | AIF-OUT.W1 | AIF-OUT.W2 | AIF-OUT.W3 | AIF-OUT.D1 |
|---|---|---|---|---|---|
| Speed control | 1003/1013/1113 | MCTRL-NACT<br>Actual speed<br>100 % = 16383 | MCTRL-MSET2<br>Torque display<br>100 % = 16383 | MCTRL-NSET2<br>Speed controller input<br>100 % = 16383 |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



### 6.5.4.1 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 braking 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 |



# Parameter setting

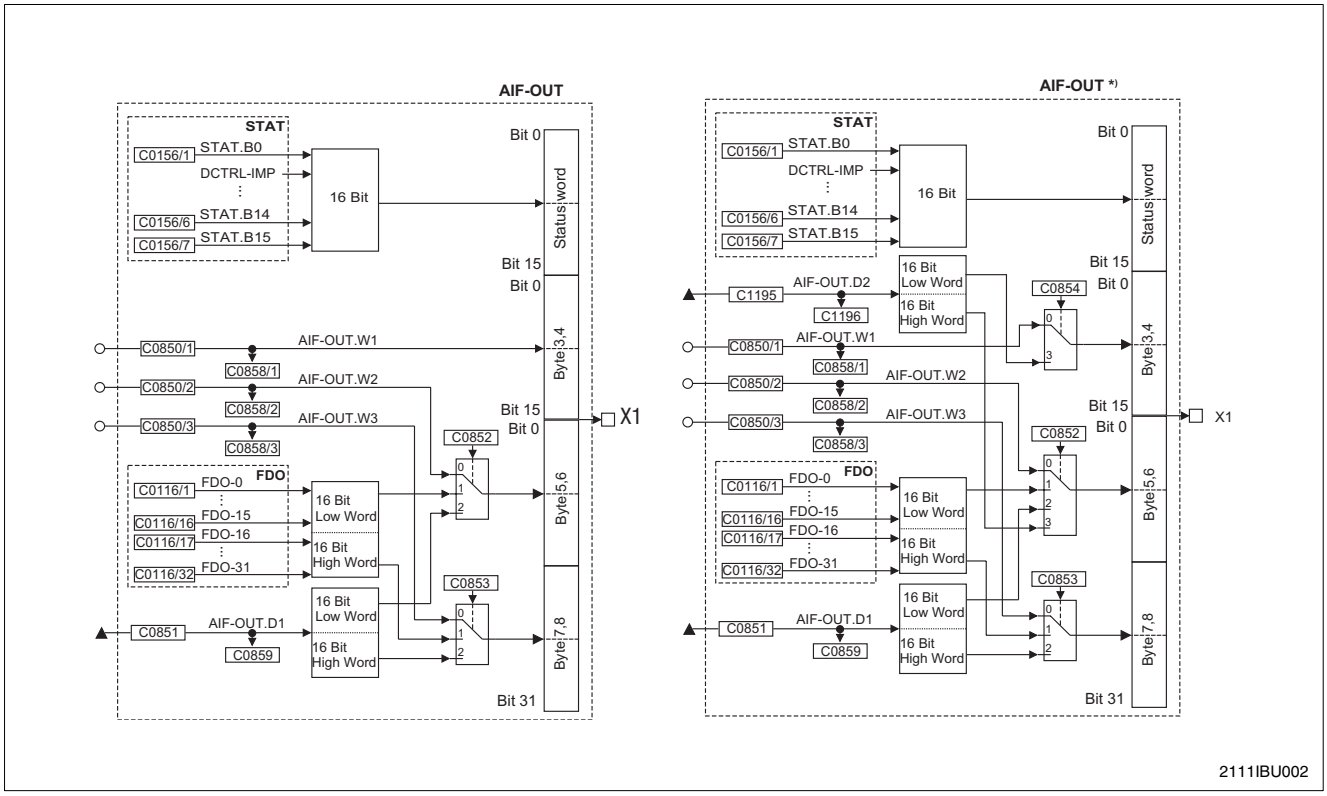
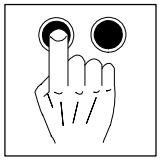



Fig. 6-9 Function block AIF-OUT and AIF-OUT*)
AIF-OUT*) is available for the 9300 technology variants: servo, positioning controller and cam profiler as of software version 2.0 . AIF-OUT.D2 is new.



### Network variable to drive

The function block AIF-IN determines the input data of the controller as data interface for the 2141 fieldbus module. For more detailed information about the function block AIF-IN, see the Manual 93XX.

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

Speed setpoint: 100 %= 16383

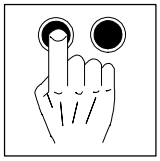
- `nviAIFStw` = AIF-CTRL.B0 ... AIF-CTRL.B15 (control word)

| Control word | |
|---|---|
| LOW byte | HIGH byte |
| Bits 0 to 7 of the control word will be entered here. | Bits 8 to 15 of the control word will be entered here (see chapter 6.5.4.2)<br>The description of the bits can be obtained from the Code Table. |

- `nviAIFIn1` = AIF-IN.W1
- `nviAIFIn2` = AIF-IN.W2
- `nviAIFIn3` = AIF-IN.W3

| AIF-IN.Wx | |
|---|---|
| LOW byte | HIGH byte |

| Signal configuration (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 | 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 | DFSET-N-TRIM<br>Speed trimming | | |
| DF-slave cascade | 7003 / 7013 / 7113 | DFSET-VP-DIV<br>DF factor | DFSET-A-TRIM<br>Phase trimming | | |
| Cam profiler | 1xxx3 | NSET1-FACT | not assigned | | |
| Positioning | 2xxx3 | not assigned | | | |
| vector control | 1xx3 / 2xx3 / 3xx3 / 5xx3 / 100x3 | NLIM-IN1 | | | |
| vector control | 4xx3 | NCTRL-MADD | | | |
| vector control | 6xx3 | DFSET-A-TRIM | DFSET-N-TRIM | | |
| vector control | 7xx3 / 8xx3 / 9xx3 | DFSET-VP-DIV | DFSET-A-TRIM | | |
| vector control | 100x3 | NLIM-IN1 | not assigned | | |
| vector control | 110x3 | not assigned | | | |





### 6.5.4.2 Control word for 93XX

| 9300 | 9300 Servo | | | | 9300 Positioning controller | 9300 Cam profiler | 9300 Vector | | |
|---|---|---|---|---|---|---|---|---|---|
| C0005 | 1xx3 | 4xx3 | 5xx3 | 6xx3,7xx3 | 2xxx3 | xxx3 | 1xxx, 2xxx, 3xxx, 5xxx, 10xxx, 11xxx | 4xx3 | 6xx3,7xx3 |
| 0 | NSET-JOG*1 | not assigned | NSET-JOG*1 | not assigned | not assigned | CSEL1-CAM*1 | NSET-JOG*1 | not assigned | not assigned |
| 1 | NSET-JOG*2 | not assigned | NSET-JOG*2 | not assigned | not assigned | CSEL1-CAM*2 | NSET-JOG*2 | not assigned | not assigned |
| 2 | NSET-N-INV | NSET-N-INV | NSET-N-INV | NSET-N-INV | not assigned | CSEL1-CAM*4 | NSET-N-INV | not assigned | not assigned |
| 3 | AIF-CTRL.QSP | AIF-CTRL.QSP | AIF-CTRL.QSP | AIF-CTRL.QSP | AIF-CTRL.QSP | AIF-CTRL.QSP | AIF-CTRL.QSP | AIF-CTRL.QSP | AIF-CTRL.QSP |
| 4 | NSET-RFG-STOP | NSET-RFG-STOP | NSET-RFG-STOP | NSET-RFG-STOP | POS-PRG-START | CSEL1-EVENT | NSET-RFG-STOP | NSET-RFG-STOP | not assigned |
| 5 | NSET-RFG-0 | NSET-RFG-0 | NSET-RFG-0 | NSET-RFG-0 | POS-PRG-STOP | CDATA-CYCLE | NSET-RFG-0 | NSET-RFG-0 | not assigned |
| 6 | not assigned | not assigned | not assigned | not assigned | not assigned | CSEL1-LOAD | not assigned | not assigned | not assigned |
| 7 | not assigned | not assigned | not assigned | not assigned | POS-PRG-RESET | CSEL1-LOAD | not assigned | not assigned | not assigned |
| 8 | not assigned | not assigned | not assigned | not assigned | not assigned | not assigned | not assigned | not assigned | not assigned |
| 9 | AIF-CTRL.CINH | AIF-CTRL.CINH | AIF-CTRL.CINH | AIF-CTRL.CINH | AIF-CTRL.CINH | AIF-CTRL.CINH | AIF-CTRL.CINH | AIF-CTRL.CINH | AIF-CTRL.CINH |
| 10 | AIF-CTRL.TRIP-SET | AIF-CTRL.TRIP-SET | AIF-CTRL.TRIP-SET | AIF-CTRL.TRIP-SET | AIF-CTRL.TRIP-SET | AIF-CTRL.TRIP-SET | AIF-CTRL.TRIP-SET | AIF-CTRL.TRIP-SET | AIF-CTRL.TRIP-SET |
| 11 | AIF-CTRL.TRIP-RESET | AIF-CTRL.TRIP-RESET | AIF-CTRL.TRIP-RESET | AIF-CTRL.TRIP-RESET | AIF-CTRL.TRIP-RESET | AIF-CTRL.TRIP-RESET | AIF-CTRL.TRIP-RESET | AIF-CTRL.TRIP-RESET | AIF-CTRL.TRIP-RESET |
| 12 | DCTRL-PAR*1 | DCTRL-PAR*1 | DCTRL-PAR*1 | DCTRL-PAR*1 | POS-PS-CANCEL | not assigned | DCTRL-PAR*1 | DCTRL-PAR*1 | DCTRL-PAR*1 |
| 13 | DCTLR-PAR-LOAD | DCTLR-PAR-LOAD | DCTLR-PAR-LOAD | DCTLR-PAR-LOAD | POS-PARAM-RD | not assigned | DCTLR-PAR-LOAD | DCTLR-PAR-LOAD | DCTLR-PAR-LOAD |
| 14 | NSET-Ti*1 | NSET-JOG*1 | REF-ON | REF-ON | POS-LOOP-ONH | not assigned | NSET-Ti*1 | NSET-JOG*1 | not assigned |
| 15 | NSET-Ti*2 | NSET-JOG*2 | NSET-Ti*1 | not assigned | POS-STBY-STP | not assigned | NSET-Ti*2 | NSET-JOG*2 | not assigned |

**Tip!**

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

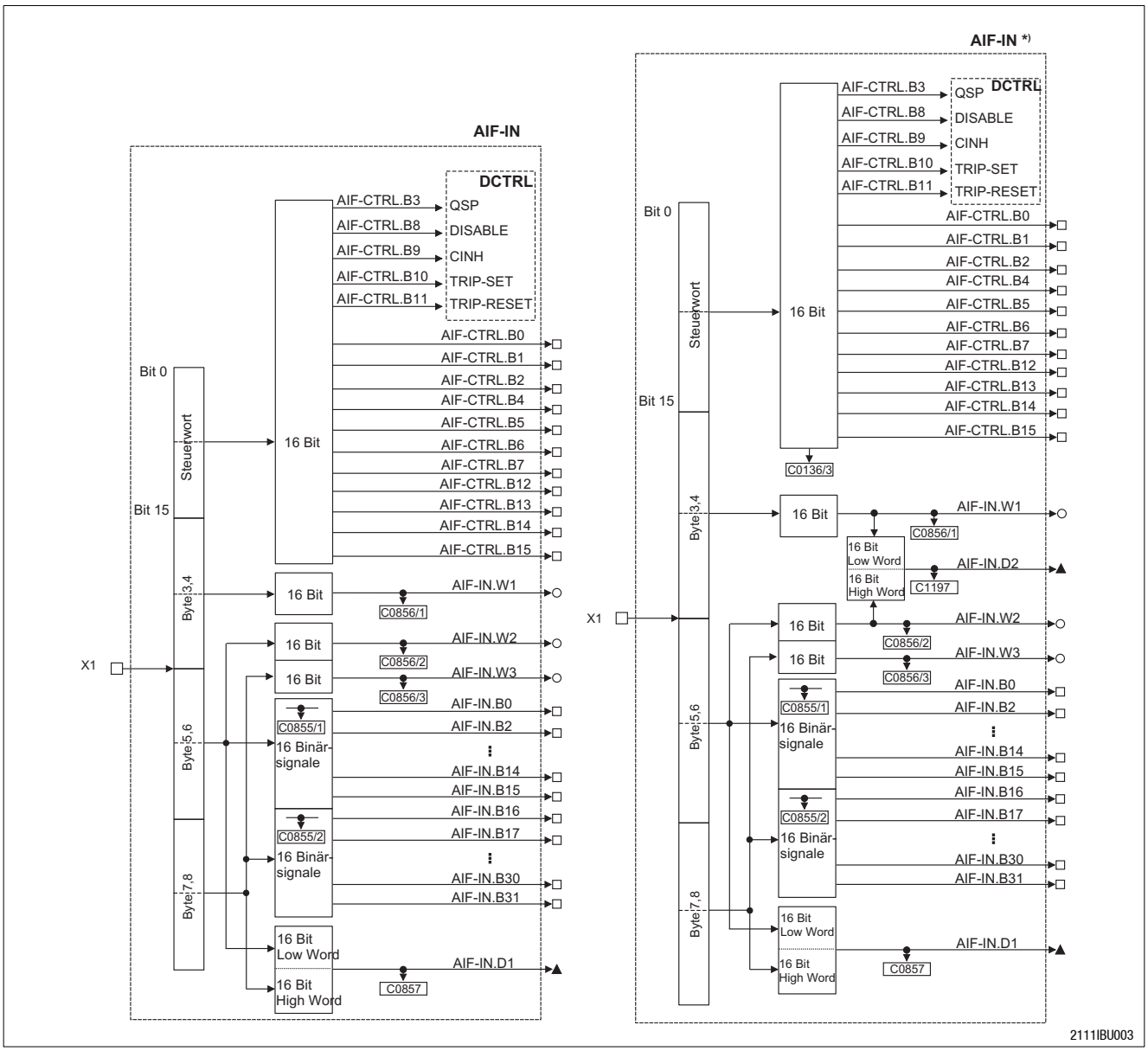
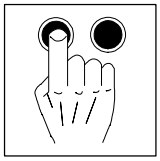



Fig. 6-10 Function block AIF-IN and AIF-IN*)
AIF-IN*) is available for the 9300 technology variants: servo, positioning controller and cam profiler as of software version 2.0 . AIF-IN.D2 is new.





## 6.5.5 Network variable for 9300 Servo PLC and Drive PLC

### Setpoint source selection

Communication via an AIF fieldbus module (e.g. 2141 LON) requires that AIF-IN 1 ... 3 or AIF-OUT ... 3 and if available the AIF management are part of the control configuration of the IEC1131 project.

### Process data telegram from drive

Following data can be assigned to 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 |

### Tip!

**9300 Servo PLC**

Link the following in the PLC program of the controller:

AIF1_wDctrlCtrl → DCTRL_wAIF1Ctrl

DCTRL_wStat → AIF1_wDctrlStat

**Drive PLC**

It is absolutely necessary to use the device control for the Drive PLC.

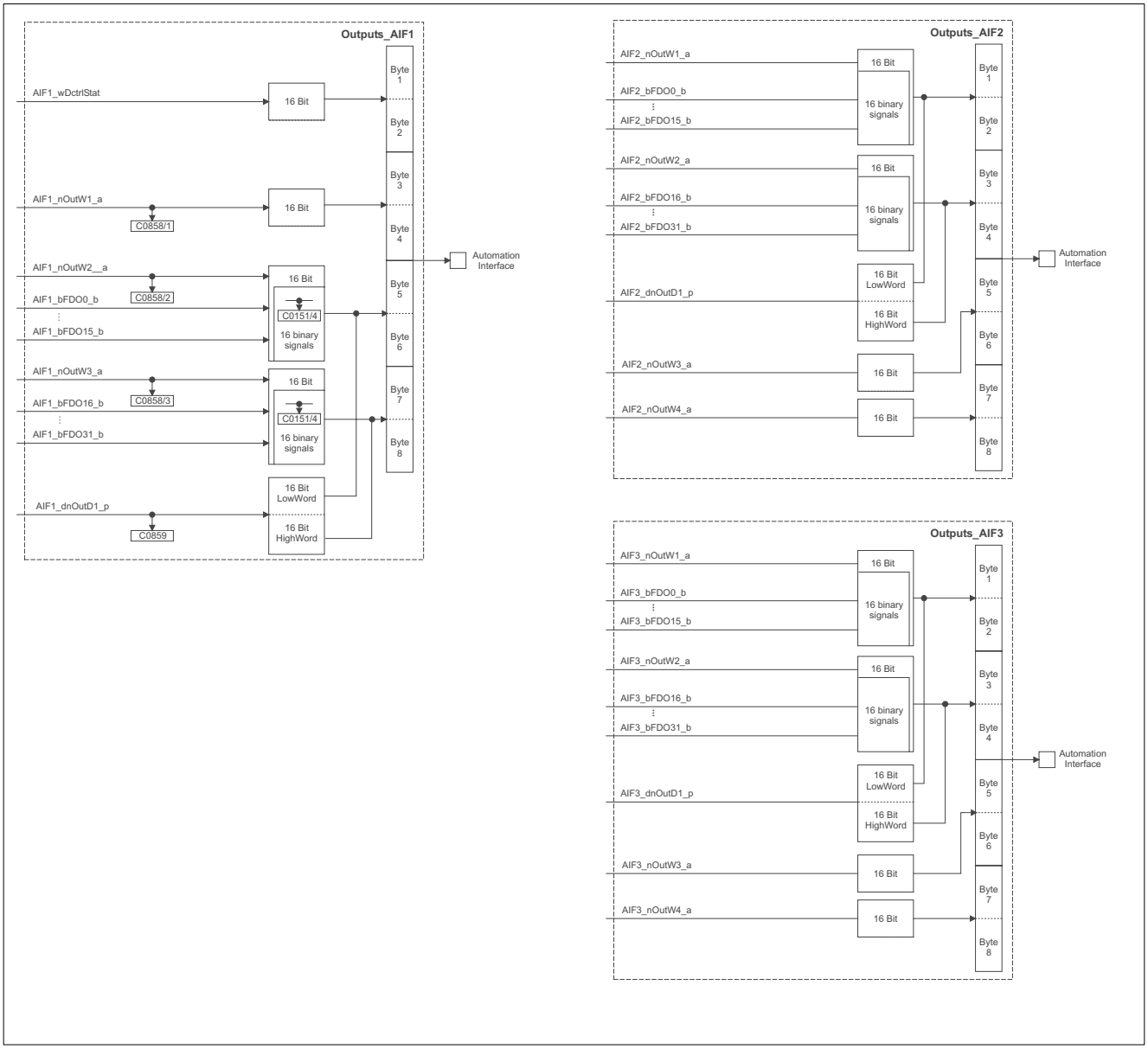
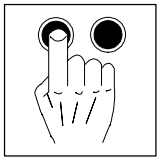



Fig. 6-11 Function blocks AIF-OUT1, AIF-OUT2 and AIF-OUT3



# Parameter setting

**Process data telegram to drive**

Following data can be assigned to PO data:

| Name/variable name | Explanation |
|---|---|
| Control word (AIF1_wDctrlCtrl) | |
| AIF1_nInW1_a | AIF word 1 |
| AIF1_nInW2_a | AIF word 2 |
| AIF1_nInW3_a | AIF word 3 |
| AIF2_nInW1_a | AIF word 4 |
| AIF2_nInW2_a | AIF word 5 |
| AIF2_nInW3_a | AIF word 6 |
| AIF2_nInW4_a | AIF word 7 |
| AIF3_nInW1_a | AIF word 8 |
| AIF3_nInW2_a | AIF word 9 |
| AIF3_nInW3_a | AIF word 10 |
| AIF3_nInW4_a | AIF word 11 |
| AIF1_dnInD1_p | AIF double word 1 |

## Tip!

**9300 Servo PLC**

Link the following in the PLC program of the controller:

AIF1_wDctrlCtrl → DCTRL_wAIF1Ctrl

DCTRL_wStat → AIF1_wDctrlStat

**Drive PLC**

It is absolutely necessary to use the device control for the Drive PLC.

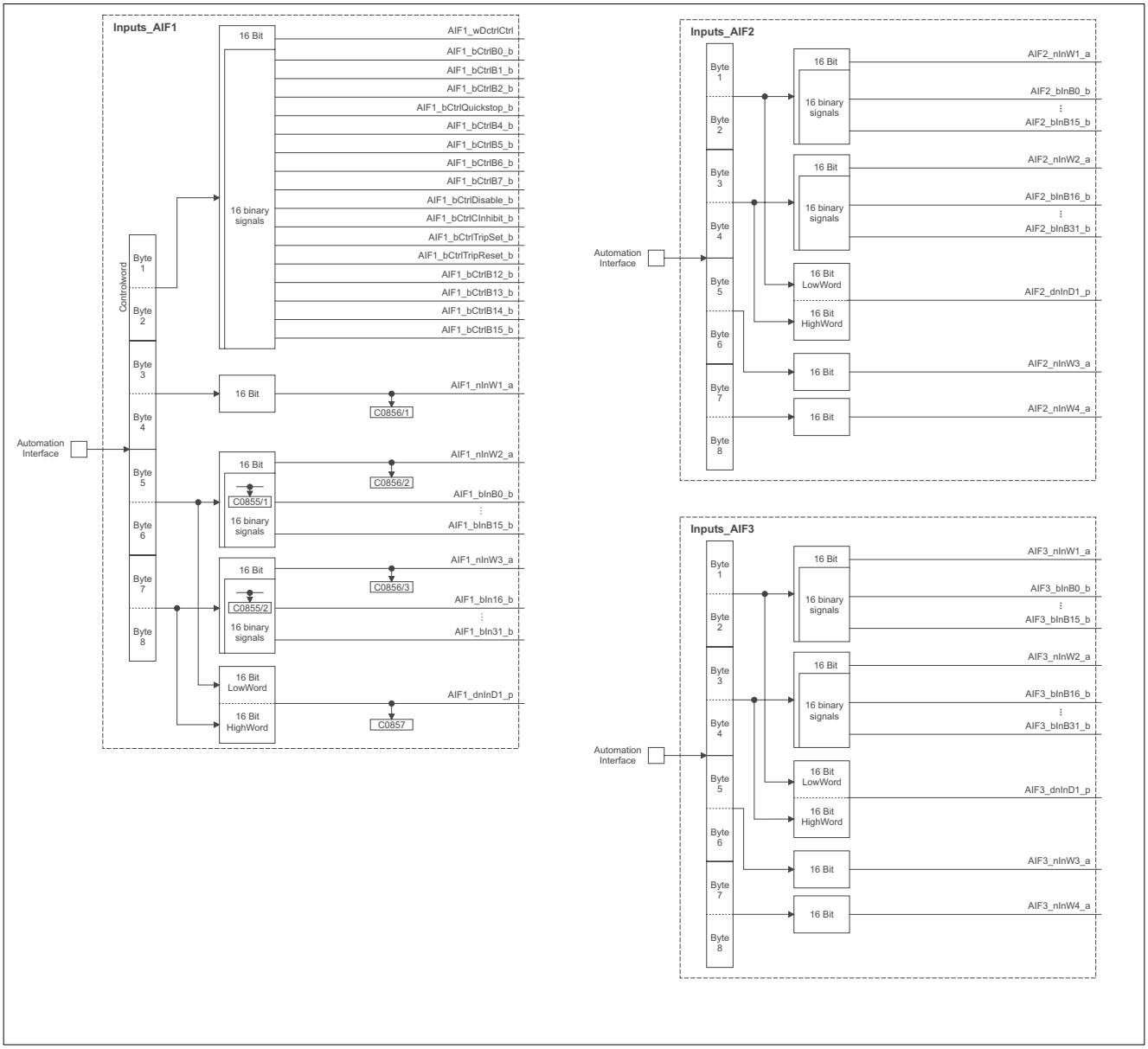
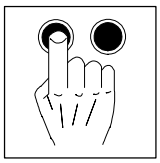



Fig. 6-12 Function blocks AIF-IN1, AIF-IN2 and AIF-IN3



# Parameter setting

# 7 Troubleshooting and fault elimination

## 7.1 No communication with the controller.

| Possible causes | Diagnostics | Remedy |
|---|---|---|
| Is the controller switched on? | One of the LED for the operating status must be on. 🕮 4-1 . | Controller supply (see Operating Instructions for controller) |
| Is the fieldbus module supplied with voltage? | The green bus LED on the fieldbus module must be 🕮 4-1 on or blinking | With internal supply, check the connection to the controller.<br>With external supply, check the corresponding terminals for a 24-V voltage: 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<br>Possibility 2: Check controller connection |
| Does the controller receive telegrams? | The yellow bus LED on the fieldbus module 🕮 4-1 must be blinking when a telegram is received correctly.<br>For testing, send telegrams cyclically from the master. | Check the wiring.<br>Check whether host sends telegrams and uses the appropriate interface. |
| | | The LON address can be set differently for controller and host. Ensure that the addresses are identical. |
| | | The addresses for all connected controllers must be unambiguous. If necessary, correct addresses.<br>Check wiring to the host. |

## 7.2 Controller does not execute write job

| Possible causes | Diagnostics | Remedy |
|---|---|---|
| Does the controller send a negative acknowledgement? | | Operating mode L-C0001: The operating mode C0001 does not match when L-C046 or L-C0135 are being addressed.<br>Set L-C0001 = 3.<br>Please see the corresponding information given in chapter "Parameter setting", 🕮 6-1 |
| | | This parameter can only be read. See the corresponding Operating Instructions. |
| Does the controller send a positive acknowledgement? | ACK response from the controller. | 82XX parameters can only be changed when the controller is inhibited.<br>Please see the corresponding information given in chapter "Parameter setting", 🕮 6-1 |
| | | The controller uses a different parameter set. The parameter change will become active after the parameter set has been changed. |

# Troubleshooting and fault elimination

# 8 Appendix

## 8.1 Starting Global Drive Control (GDC) from LONMaker™

Global Drive Control (GDC) can be directly started from LONMaker™ for Windows® by means of a plug-in.

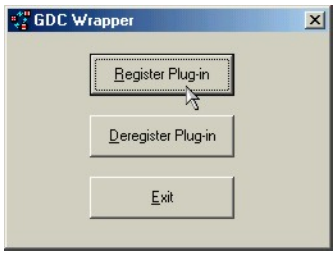
Software requirements:

- LONMaker™ for Windows® as of version 3.00.66
- “Global Drive Control easy” or “Global Drive Control, both as of version 4.4

### 8.1.1 Installation instructions for the plug-In

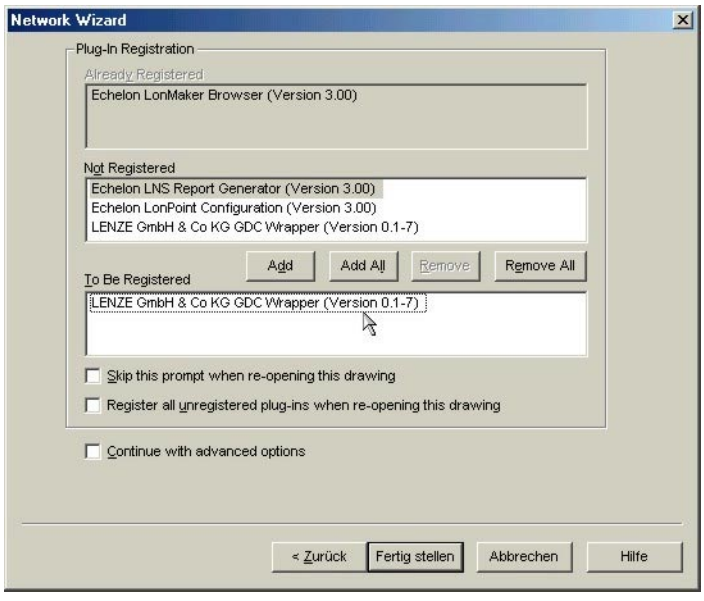
1. Copy the files “LONMaker2GDC.exe” and “GDCWrapper.exe” from the floppy (included in the delivery package) to the harddisk.
2. Register these fieles in the system

- using the following commands under DOS for Windows®
  – **LONMaker2GDC.exe /regserver**
  – **GDCWrapper.exe /regplugin**
- As alternative, the “GDCWrapper” can also be opened in Windows (e.g. double click the file ’GDCWrapper.exe’ in the Windows Explorer). The following dialog box comes up:




2141LON005

3. Copy the file “2141.XIF” to the directory “...\Lonworks\Import”.
4. Registration of the plug-In in LONMaker™ for Windows®.

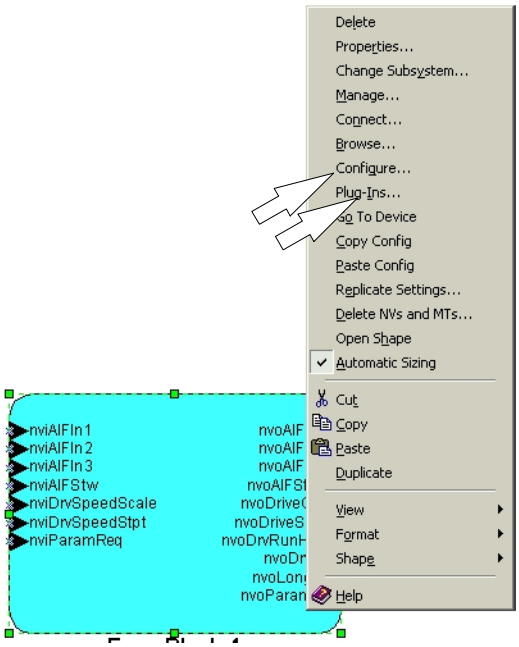



2141LON006

## 8.1.2 Open GDC from the LONMaker™

Global Drive Control (GDC) is automatically opened through the installed plug-in if you select the following from the shown context menu (right mouse key in the working area of LONMaker™):

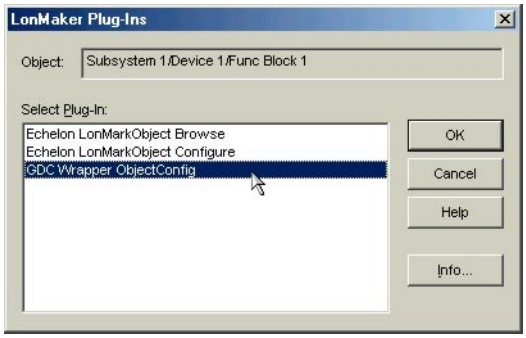
- Start GDC directly under **Configure...**




2141LON007

- or start GDC indirectly through **Plug-Ins...**.

Activate the 'GDCWrapper' in the selection box that will open.




2141LON008

Go to GDC and press the function key F2 (*Find drive*) to set the parameters for the drive. An online connection with the connected controller is necessary.

## 8.2 Configuration of the LON network

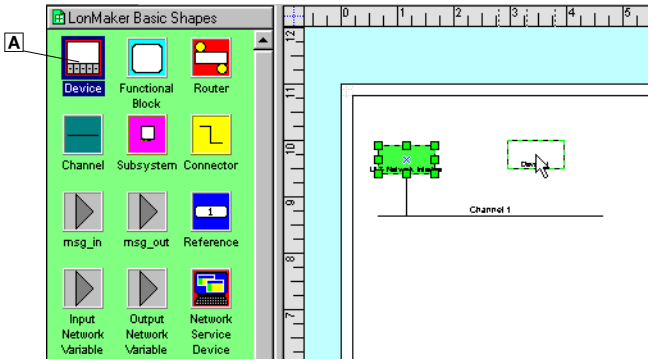
The LONMaker™ for Windows® helps to configure your drive. We use the software version 3.00.66 of LONMaker™ for Windows® which is only available in English at the time of this print.

The windows can be installed under Windows95®, Windows98® or WindowsNT®. The configuration is explained by examples but only as much as needed for the operation of Lenze controllers.

### 8.2.1 Configuration of the LON network

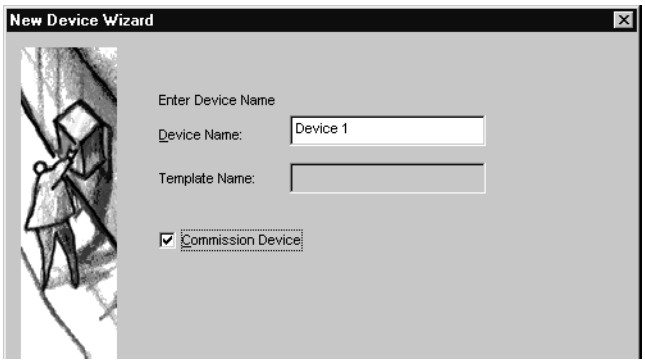
1. Insert new device connected to the network

Click the symbol “Device” and A shift it to where you need it.




2141LON105

2. Go to the dialog box *Device Name* and enter the name of the device. Highlight the function *Commission Device*.




2141LON106

3. Assignment of interface specification of the connected device

After the device name has been entered (see figure above), it must be linked with the device connected to the network.

**Tip!**

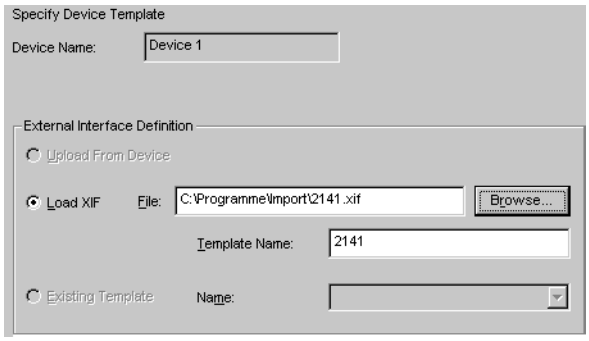
This link is necessary no matter whether you want to use a different device later or load a new application to the device.

# Appendix

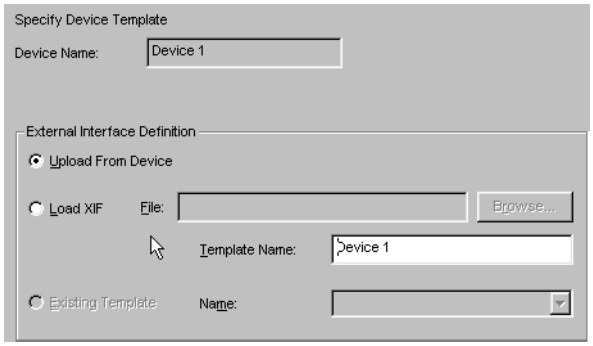
The specification can be made online and offline.

- OFFLINE (”Load XIF”, A): The specification is read from the file 2141.XIF.
- ONLINE (“Upload from Device”, B): The specification is read from the module. Please see steps 4. and 5.

A

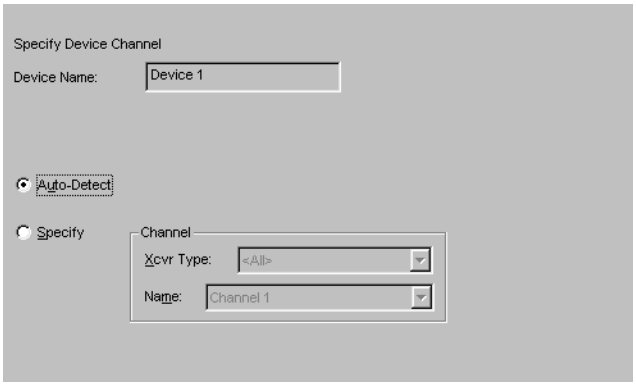


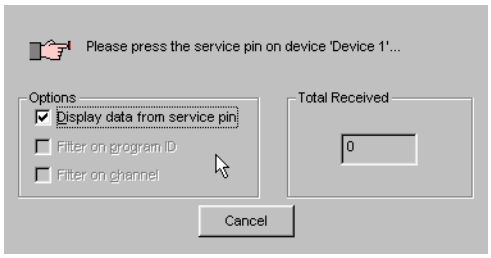

B




2141LON107

4. Select the function *Auto-Detect* (only possible with online connection)




2141LON108/111

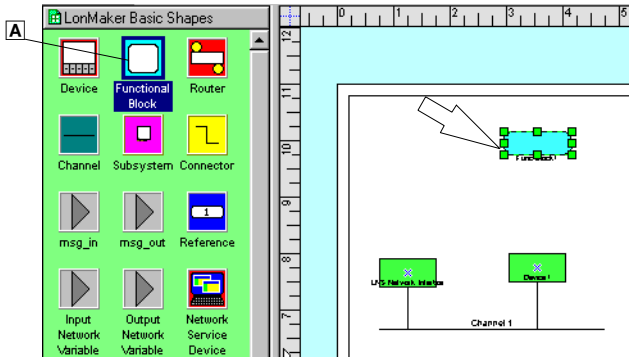
5. Press the Service key F at the front of the 2141 fieldbus module.

All configuration features of the device are automatically loaded by using the service PIN.

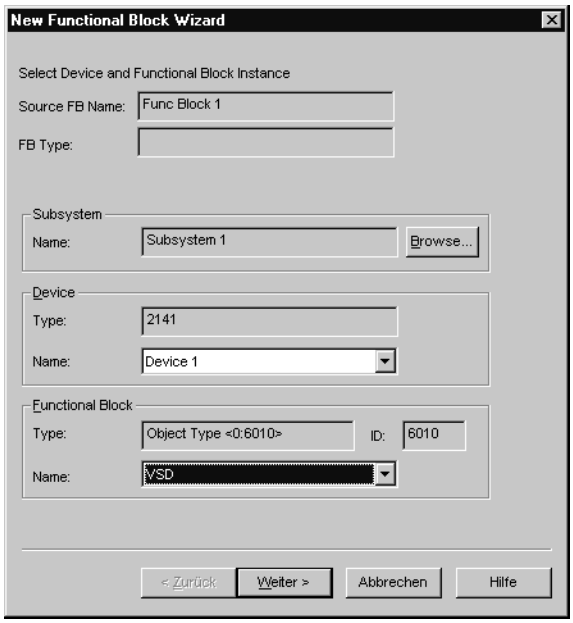
## 8.2.2 Working with network variables

### Insertion of function blocks

Click the “Functional Block” symbol and A shift it to where you need it.




2141LON114

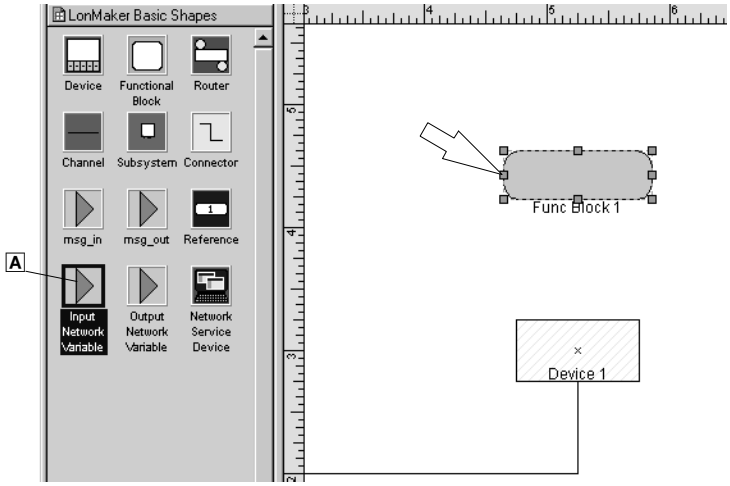



2141LON116

# Appendix

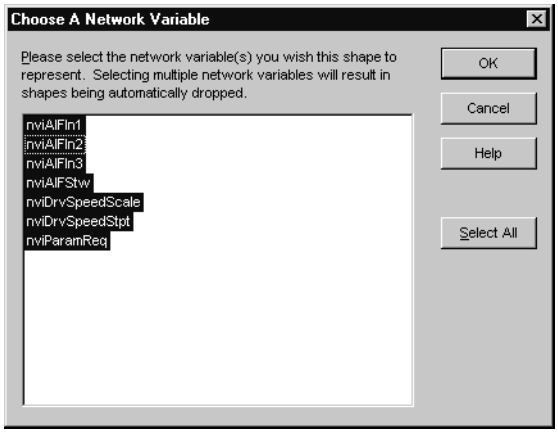
### Insertion of network variables

Click *Input Network Variable* and shift it to the function block ("Func Block 1").



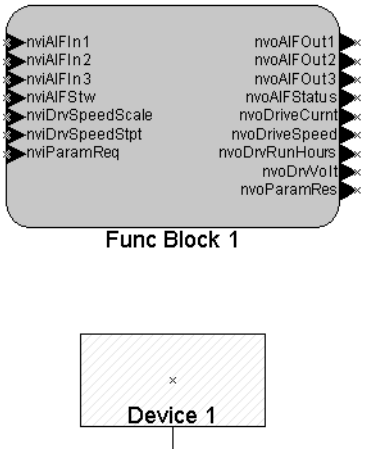

2141LON120

Highlight the network variables needed at the input.



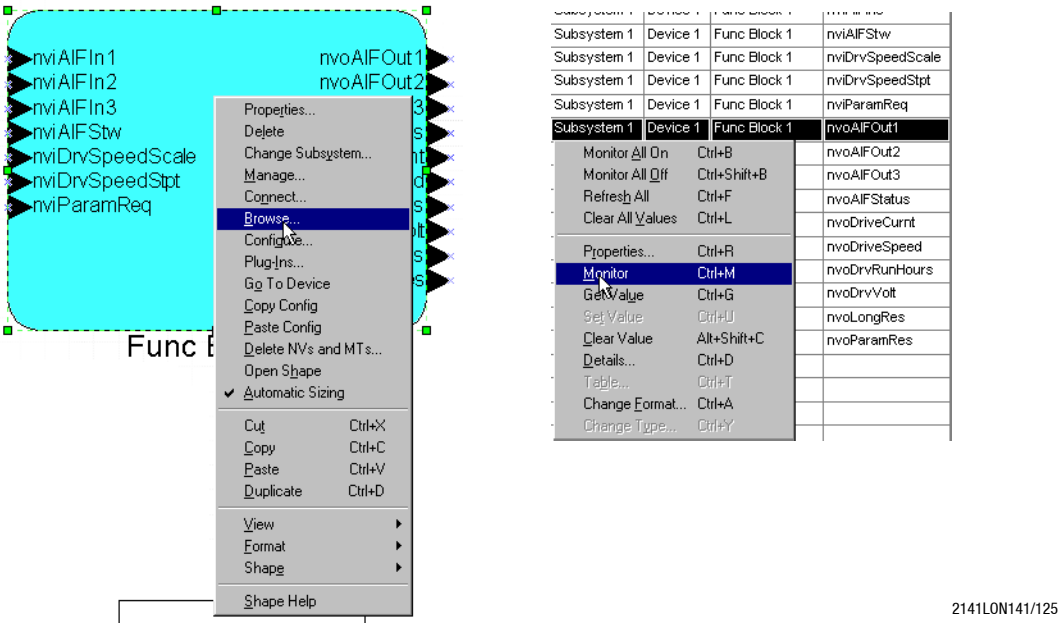

2141LON123

Repeat the same procedure for the output variables. The function block will be represented as follows:



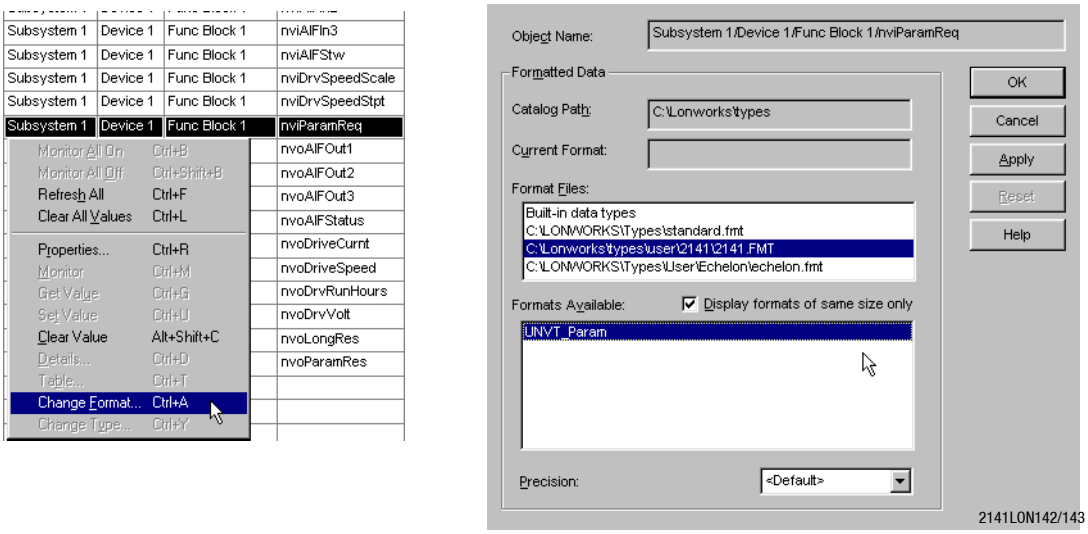

2141LON122

## Editing network variables with a browser




2141LON141/125

## Change of variable format

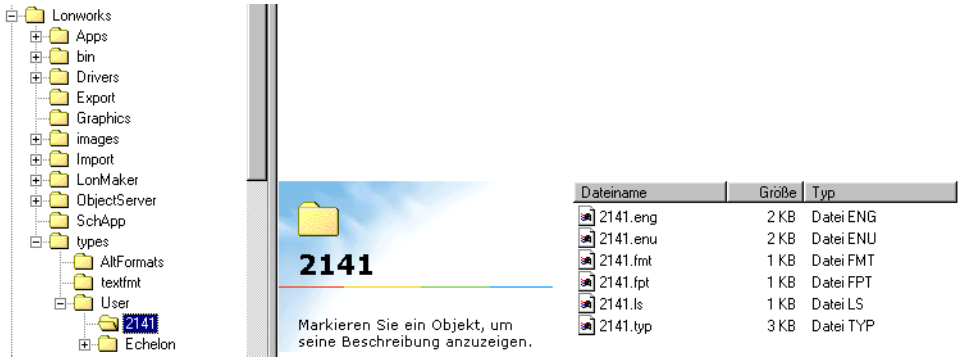



2141LON142/143

## 8.2.3 Registration of new data types

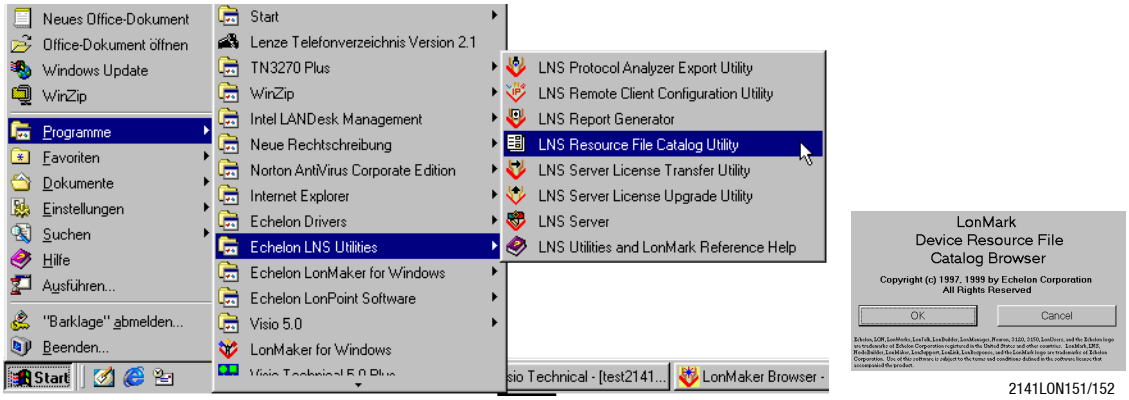
If you use these files, you will get more information and have higher configuration friendliness for the LON network with LONMaker™ for Windows®.

### Copy files to hard disk.



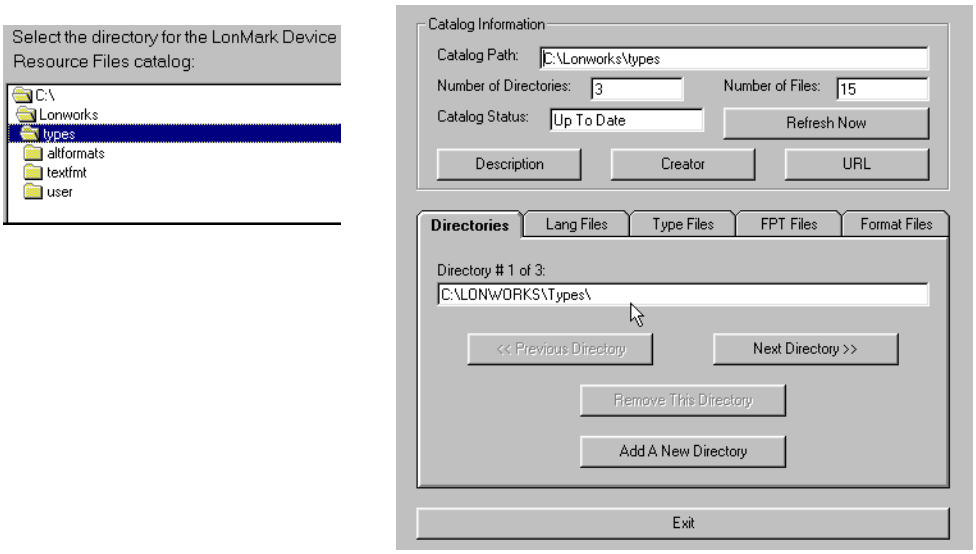

2141LON150

### Start “LNS Resource File Catalog Utility“



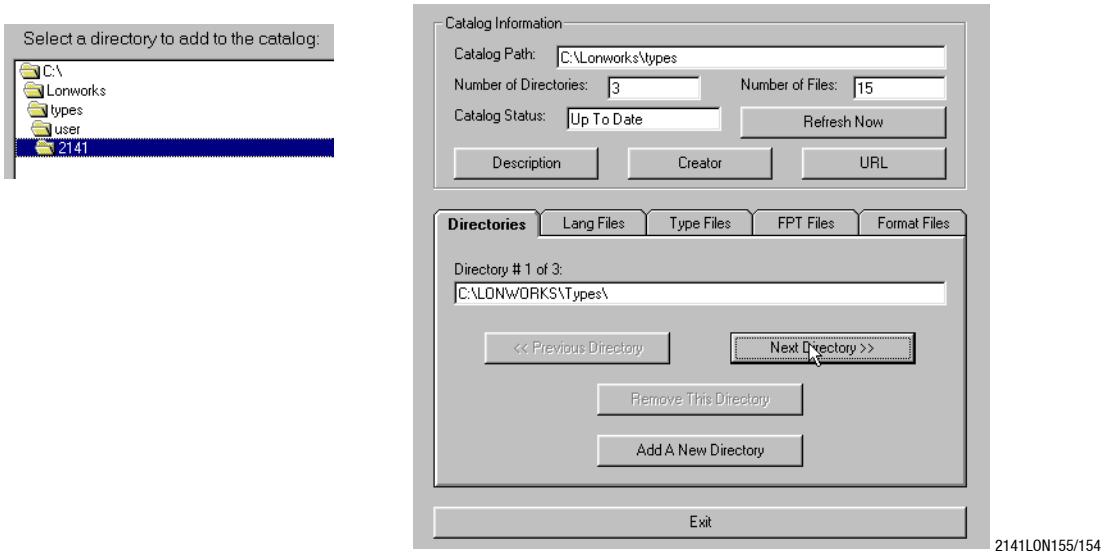

2141LON151/152

### Select Resource File Catalog directory




2141LON153/154

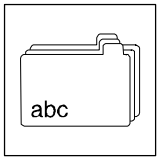
### Add new type directory




Click the buttons “Add a new directory” and “Refresh Now”.

## 8.3 List of abbreviations

| Abbreviation | Meaning |
|---|---|
| AIF | Automation Interface; interface between controller and automation or fieldbus modules. It also includes defined process data. |
| EMC | Electromagnetic Compatibility |
| DCB | DC-injection brake |
| hex | Display of values in the hexadecimal character format (0, …, 9, A, B, …, F ). |
| RFG | Ramp-function generator; setpoint integrator |
| Imax | Current limit |
| IMP | Pulse inhibit |
| JOG | Fixed speed or input for activation of the fixed speed |
| CRL | Communication reference list |
| LSB | Least Significant Bit; low-weighting bits of a binary value |
| Max-PDU | Maximum Process Data Unit |
| MSB | Most Significant Bit; high-weighting bits of a binary value |
| PO-data | Process-output data |
| PC | Personal Computer |
| PI-data | Process-input data |
| PE | Protective earth |
| PCD | Process data, |
| QSP | Quick stop |
| Ctrl. enable | Controller enable |
| RS232 | Interface standard |
| RS485 | Interface standard with difference signals |
| Ctrl. inhibit | Controller inhibit |
| TRIP | Operation fault |
| Vcc | Controlled constant voltage supply |



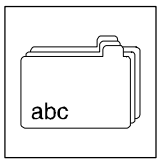

# Appendix

## 8.4 Glossary

| Technical term | Meaning |
|---|---|
| Controller | General term for servo drives, frequency inverters and DC drives |
| Baud rate | Transmission speed of data in bit/s |
| Bus terminal | Network node between long-distance and peripheral bus |
| Bus devices | Unit which communicates with the host via the bus |
| code | For input and display (access) of parameter values. |
| Code number/code | Clear labelling of a parameter, e.g. C0106. For calculation information see Lenze code addressing in chapter „Parameter setting“, |
| Fieldbus | For the exchange of data between higher-level controls (hosts) and positioning controls (e. g. controllers). |
| FTT | Free Topology Transceiver. |
| Handshake | Stipulated data-transfer method (here implemented by software). |
| Host | PC or PLC |
| Master | see host/host system |
| LON | Logical Operating Network |
| Network topology | Design and structure of a network: e.g. point-to-point network, line network, ring network) |
| Parameters | Adjustable controller variables and values addressable via codes |
| Peripheral-bus module | Bus participant in the peripheral bus |
| Peripheral-bus station | Consists of bus terminal, peripheral-bus module and long-distance bus controller. |
| Icon | Sign or symbol with an unambiguous message. |
| Process data, | Small amounts of data for fast and cyclic transmission; e.g. setpoints and actual values |
| Process-data channel (PCD channel) | Communication channel for fast and cyclic transmission of process data |
| Acknowledgement | Acknowledgement of a setting or change (e. g. of parameters)) |
| Slave | Bus participant which is only allowed to transmit data after a request by a master. For instance, controllers are slaves. (See Host/host system) |
| Application | as directed:<br>Appropriate use of the machine according to the manufacturer’s information or to common use because of its design and function.<br>inappropriate<br>The use of the machine which does not comply with the manufacturer’s declaration. |
| Cycle time | The cycle time of the communication system is the time required for the exchange of all process data between the host and the field units (e.g. controllers). |

## 8.5 More information sources

| No. | Titel | Published by |
|---|---|---|
| 1 | Engineering Bulletin 170, **LonTalk Response Time Measurements** | Echelon Corporation |
| 2 | LonMark® Functional Profile: Variable Speed Motor Drive | LONMARK Interoperability Association |
| 3 | LonWorks® FTT-10A Free Topology Transceiver User’s Guide | Echelon Corporation |
| 4 | LonMaker™ for Windows® User’s Guide | Echelon Corporation |




## 8.6 Table of keywords


### A

Acceleration time, 6-19

Actual speed value, 6-6

AIF-CTRL, Device control, 6-2, 6-24

AIF-IN, Function block, 6-41

AIF-OUT, Function block, 6-38

Appendix, 8-1

Application as directed, 1-2

Application conditions, 3-1

### B

Basic insulation, 4-4

### C

Commissioning, 5-1

Communication interference, 4-4

Communication medium, 3-2

Communication times, 3-3

Configuration variable
- Acceleration time, 6-19
- Deceleration time, 6-20
- Description , 6-11
- Location label, 6-11
- Maximum Send Time, 6-13
- Maximum speed, 6-15
- Minimum Send Time, 6-14
- Minimum speed, 6-16
- Overview, 6-3
- Rated frequency, 6-18
- Rated speed value, 6-17
- Receive heartbeat for manufacturer-specific network variables, 6-21
- Receive Heartbeat Time, 6-12

Connection
- External supply, 4-2
- Plug connector (2-pole), 4-2

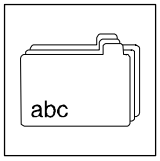
Connection terminals, 4-2

Connector plug, LON bus, 4-2

Control word
- 8200 vector, 6-30, 6-33
- 82XX, 6-26, 6-29
- 93XX, 6-37, 6-40

Controller
- Application as directed, 1-2
- Labelling, 1-2

Controller configuration , 5-2




# Appendix


## D

Data types, Registration, 8-8

Deceleration time, 6-20

Definitions, 1-1

Description , Configuration variable VSD, 6-11

Device control
- AIF-CTRL, 6-24
  - Monitoring reaction, 6-22
- VSD, 6-5

Dimensions, 3-2

Disposal, 1-2

## E

Electrical data. *Siehe* Ratings

## F

Fault elimination, 7-1

Fieldbus module, Order name, 3-1

First switch-on, 5-1

Frequency setpoint
- 8200 vector, 6-30
- 82XX, 6-26

Front view of the fieldbus module, 4-1

FTT, 3-2

Function module PROFIBUS-DP
- Lenze parameters, 6-4
- Parameters, Transmission, 6-1
- Process data,, Transmission, 6-1

## G

GDCWrapper.exe, 8-1

Glossary, 8-10
- *Siehe auch* Glossary

## I

Installation
- Electrical, 4-4
- Mechanical, 4-3
- Wiring to a host, 4-4

## L

Labelling, Controller, 1-2

Legal regulations, 1-2

Lenze parameter sets, 6-2

Lenze parameters, Function module PROFIBUS-DP, 6-4

Liability, 1-2

List
- Abbreviations, 8-9
- Technical term. *Siehe* Glossary

List of abbreviations, 8-9

Location label, 6-11

LON network, Configuration, 8-3

LONMaker, Automatic GDC start, 8-1

LONMaker2GDC.exe, 8-1

## M

Mains isolation, 4-4

Manufacturer, 1-2

Maximum Send Time, 6-13

Maximum speed, 6-15

Minimum Send Time, 6-14

Minimum speed, 6-16

Monitoring reaction, AIF-CTRL, 6-22

Motor current, 6-8

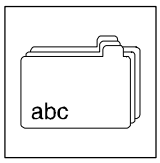
Motor voltage, 6-9

## N

Network variable, 8-5
- AIF-CTRL
  - 8200 vector, 6-30
  - 82XX, 6-26
  - 9300 Servo PLC, 6-42
  - 93XX, 6-35
  - Drive PLC, 6-42
  - Input variables, 6-25
  - Output variables, 6-24
- Overview, 6-2
- VSD
  - Actual speed value, 6-6
  - Motor current, 6-8
  - Motor voltage, 6-9
  - Operating time, 6-10
  - Scaling of the speed setpoint, 6-7
  - Speed setpoint, 6-5

Node Object, 6-1

Number of devices connected to the bus, 3-3




## O

Operating time, 6-10

Operator, 2-1

Overview
- Configuration variable, 6-1, 6-3
- Network variable, 6-1
  - AIF-CTRL, 6-2
  - VSD, 6-2
- Node Object, 6-1

## P

Parameter data channel, configuration, 6-2

Parameter sets, 6-2

Parameter setting, 6-1

Parameters
- Control word (C0135)
  - 8200 vector, 6-30
  - 82XX, 6-26
- Frequency setpoint (C0046)
  - 8200 vector, 6-30
  - 82XX, 6-26
- L-C0142, 5-2

PE shield cable, 4-4

Personnel, Qualified, 2-1

Processing time in the basic unit, 3-3

## R

Rated frequency, 6-18

Rated speed value, 6-17

Ratings, 3-2

Receive Heartbeat Time, 6-12

Registration, of data types, 8-8

## S

Safety information, 2-1
- Layout, 2-2
  - Other notes, 2-2
  - Warning of material damage, 2-2
  - Warning of personal damage, 2-2

Safetyinformation, Operating Instructions, 1-1

Scaling of the speed setpoint, 6-7

Scope ofsupply, 1-1

Service key, 4-1

Service LED, 4-1

Speed
- Maximum, 6-15
- Minimum, 6-16

Speed setpoint, 6-5
- Scaling, 6-7

Starting GDC, from LONMaker, 8-1

Status word, 82XX, 6-27

Switch-on sequence, First switch-on, 5-1

## T

Technical data, 3-1

Technicaldata
- Communication times, 3-3
- Dimensions, 3-2
- Features of the fieldbus module, 3-1
- General data/Application conditions, 3-1
- Ratings, 3-2

Troubleshooting, 7-1

## U

Use, as directed, 1-2

## V

Voltage supply, 4-5

Voltage supply, external, 4-2

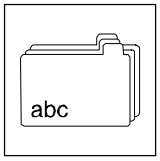
VSD, Device control, 6-2, 6-5

## W

Warranty, 1-2

Wiring, to a host, 4-4

Write parameter (example), 6-6




# Appendix