Controller-based Automation

CANopen®
## Contents

1. **About this documentation**  
  ➢  Document history  
  ➢  Conventions used  
  ➢  Terminology used  
  ➢  Definition of the notes used  
2. **Safety instructions**  
3. **Controller-based Automation: Central motion control**  
4. **System bus (CAN) / CANopen**  
  ➢  CANopen (Logic) / CANopen (Motion)  
  ➢  Field devices  
  ➢  CANopen hardware for Lenze Controllers  
  ➢  CAN wiring  
  ➢  Lenze Engineering tools  
5. **Technical data**  
  ➢  General data  
  ➢  Technical data of the MC-CAN2 communication card  
  ➢  CAN interface of the c300/p300 controller  
5.5.1 **Total cable length**  
5.5.2 **Segment cable length**  
5.5.3 **Use of repeaters**  
6. **Planning the CANopen network**  
  ➢  COB-IDs acc. to DS301  
  ➢  Example of an overview screen  
  ➢  Device specifications of the field devices  
  6.3.1 **Special features of the Servo Drives 9400**  
  6.3.2 **Special features of the Inverter Drives 8400**  
  6.3.3 **Special features of the I/O-System 1000 (EPM-Sxxx)**  
  ➢  Special case: Delayed switch-on of one or more slaves  
6.5 **Preparing the field devices**  
7. **Installing field devices**  
7.2 **Setting node addresses and baud rate**  
7.3 **Connecting the Engineering PC to the Lenze Controller**  
8. **Preparation of the Engineering PC**  
9. **Installation of the Engineering tools**  
10. **Commissioning the Engineering tools**  
11. **Maintenance and diagnostic tools**  
12. **Commissioning the CANopen network**  
13. **Controller-based Automation: Central motion control**  
14. **Safety instructions**  
15. **Controller-based Automation: Central motion control**  
16. **System bus (CAN) / CANopen**  
17. **CANopen (Logic) / CANopen (Motion)**  
18. **Field devices**  
19. **CANopen hardware for Lenze Controllers**  
20. **CAN wiring**  
21. **Lenze Engineering tools**  
22. **Technical data**  
23. **Technical data of the MC-CAN2 communication card**  
24. **CAN interface of the c300/p300 controller**  
25. **Bus cable specification**  
26. **Bus cable length**  
27. **Planning the CANopen network**  
28. **COB-IDs acc. to DS301**  
29. **Example of an overview screen**  
30. **Device specifications of the field devices**  
31. **Special features of the Servo Drives 9400**  
32. **Special features of the Inverter Drives 8400**  
33. **Special features of the I/O-System 1000 (EPM-Sxxx)**  
34. **Special case: Delayed switch-on of one or more slaves**  
35. **Preparing the field devices**  
36. **Installing field devices**  
37. **Setting node addresses and baud rate**  
38. **Connecting the Engineering PC to the Lenze Controller**  
39. **Preparation of the Engineering PC**  
40. **Installation of the Engineering tools**  
41. **Commissioning the Engineering tools**  
42. **Maintenance and diagnostic tools**  
43. **Commissioning the CANopen network**  
44. **Controller-based Automation: Central motion control**
## Contents

### Chapter 8: Commissioning the CANopen Logic bus
- 8.1 Sample projects (Application Samples) ................................................. 38
- 8.2 Overview of the commissioning steps .................................................. 39
- 8.3 Create a project folder ........................................................................... 40
- 8.4 Commission the field devices ................................................................. 41
- 8.5 Creating a PLC program with a target system (Logic) .......................... 42
- 8.6 Configuring the communication parameters ....................................... 44
- 8.7 Importing missing devices / device description files ............................ 46
- 8.8 Creating a control configuration (adding field devices) ....................... 47
- 8.9 Setting of CAN parameters and PDO mapping .................................... 52
  - 8.9.1 Cross communication between the slaves ....................................... 55
  - 8.9.2 Special features of the I/O-System 1000 (EPM-Sxxx) ....................... 56
- 8.10 Creating the program code for controlling the Logic field device .......... 59
- 8.11 Preparing the restart ............................................................................ 61
  - 8.11.1 Special features of the 9400 servo drives HighLine ....................... 62
  - 8.11.2 Special features of the Inverter Drives 8400 .................................. 62
  - 8.11.3 Special features of the I/O-System 1000 (EPM-Sxxx) ....................... 63
- 8.12 Compiling the PLC program code ....................................................... 66
- 8.13 Logging in on the Lenze Controller with the »PLC Designer« ............ 66
- 8.14 Starting the PLC program ................................................................. 66
- 8.15 Start parameters of the Servo Drives 9400 HighLine CiA 402 .......... 66

### Chapter 9: Commissioning the CANopen Motion bus
- 9.1 Sample projects (Application Samples) ................................................ 68
- 9.2 Overview of the commissioning steps .................................................. 69
- 9.3 Create a project folder ........................................................................... 70
- 9.4 Commission the field devices ................................................................. 71
- 9.5 Creating a PLC program with target system (Motion) .......................... 72
- 9.6 Configuring the communication parameters ....................................... 74
- 9.7 Creating a Motion task ......................................................................... 76
- 9.8 Creating a control configuration ............................................................. 80
- 9.9 Parallel operation of two synchronised CAN buses ............................. 84
- 9.10 Setting SoftMotion parameters ............................................................. 86
- 9.11 Setting of CAN parameters and PDO mapping .................................... 89
- 9.12 Creating the program code for controlling the Motion field device ...... 92
- 9.13 Preparing the restart ............................................................................ 92
- 9.14 Compiling the PLC program code ....................................................... 93
- 9.15 Logging in on the Lenze Controller with the »PLC Designer« ............ 93
- 9.16 Starting the PLC program ................................................................. 93
- 9.17 Start parameters of the Servo Drives 9400 HighLine CiA 402 .......... 93

### Chapter 10: Mixed operation of CANopen and EtherCAT
- 10.1 Sample projects (Application Samples) ............................................... 94
- 10.2 Overview of the commissioning steps .................................................. 95
- 10.3 Creating a project folder ...................................................................... 96
- 10.4 Commission the field devices ............................................................... 97
- 10.5 Creating a PLC program with target system (Logic) ........................... 97
- 10.6 Configuring the communication parameters ....................................... 99
- 10.7 Creating the program code for controlling the Logic field device ........ 100
- 10.8 Preparing the restart .......................................................................... 101
- 10.9 Compiling the PLC program code ....................................................... 102
- 10.10 Logging in on the Lenze Controller with the »PLC Designer« .......... 102
- 10.11 Starting the PLC program ................................................................. 102
- 10.12 Start parameters of the Servo Drives 9400 HighLine CiA 402 .......... 102

### Chapter 11: SM3_Drive_Lenze.lib function library
- 11.1 Sample projects (Application Samples) ............................................... 95
- 11.2 Overview of the commissioning steps .................................................. 96
- 11.3 Creating a project folder ...................................................................... 97
- 11.4 Commission the field devices ............................................................... 98
- 11.5 Creating a PLC program with target system (Logic) ........................... 98
- 11.6 Configuring the communication parameters ....................................... 100
- 11.7 Creating the program code for controlling the Logic field device ........ 101
- 11.8 Preparing the restart .......................................................................... 101
- 11.9 Compiling the PLC program code ....................................................... 102
- 11.10 Logging in on the Lenze Controller with the »PLC Designer« .......... 102
- 11.11 Starting the PLC program ................................................................. 102
- 11.12 Start parameters of the Servo Drives 9400 HighLine CiA 402 .......... 102

### Chapter 12: Restarting the CAN bus
- 12.1 Sample projects (Application Samples) ............................................... 95
- 12.2 Overview of the commissioning steps .................................................. 96
- 12.3 Creating a project folder ...................................................................... 97
- 12.4 Commission the field devices ............................................................... 98
- 12.5 Creating a PLC program with target system (Logic) ........................... 98
- 12.6 Configuring the communication parameters ....................................... 100
- 12.7 Creating the program code for controlling the Logic field device ........ 101
- 12.8 Preparing the restart .......................................................................... 101
- 12.9 Compiling the PLC program code ....................................................... 102
- 12.10 Logging in on the Lenze Controller with the »PLC Designer« .......... 102
- 12.11 Starting the PLC program ................................................................. 102
- 12.12 Start parameters of the Servo Drives 9400 HighLine CiA 402 .......... 102

### Chapter 13: Defining the minimum cycle time of the PLC project
- 13.1 Sample projects (Application Samples) ............................................... 95
- 13.2 Overview of the commissioning steps .................................................. 96
- 13.3 Creating a project folder ...................................................................... 97
- 13.4 Commission the field devices ............................................................... 98
- 13.5 Creating a PLC program with target system (Logic) ........................... 98
- 13.6 Configuring the communication parameters ....................................... 100
- 13.7 Creating the program code for controlling the Logic field device ........ 101
- 13.8 Preparing the restart .......................................................................... 101
- 13.9 Compiling the PLC program code ....................................................... 102
- 13.10 Logging in on the Lenze Controller with the »PLC Designer« .......... 102
- 13.11 Starting the PLC program ................................................................. 102
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>14 Diagnostics</td>
<td>100</td>
</tr>
<tr>
<td>14.1 Logbook of the Lenze Controller</td>
<td>100</td>
</tr>
<tr>
<td>14.2 &quot;Status&quot; tab of the connected field devices</td>
<td>101</td>
</tr>
<tr>
<td>14.3 Diagnostic codes</td>
<td>101</td>
</tr>
<tr>
<td>14.4 System bus configurator of the »Engineer«</td>
<td>102</td>
</tr>
<tr>
<td>14.5 »PCAN-View« for diagnostic purposes</td>
<td>103</td>
</tr>
<tr>
<td>14.5.1 Monitor telegram traffic on the CANopen bus</td>
<td>103</td>
</tr>
<tr>
<td>14.5.2 Setting all CANopen nodes to the &quot;Operational&quot; status</td>
<td>105</td>
</tr>
<tr>
<td>14.6 Notes regarding the visualisation using »VisiWinNET«</td>
<td>106</td>
</tr>
<tr>
<td>15 Parameter reference</td>
<td>107</td>
</tr>
<tr>
<td>Index</td>
<td>109</td>
</tr>
<tr>
<td>Your opinion is important to us</td>
<td>111</td>
</tr>
</tbody>
</table>
1 About this documentation

This documentation ...

• contains detailed information about the commissioning, configuration, and diagnostics of the CANopen® bus system as part of the Lenze automation system Controller-based Automation.

• is part of the "Controller-based Automation" manual collection. It consists of the following sets of documentation:

<table>
<thead>
<tr>
<th>Documentation type</th>
<th>Subject</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product catalogue</td>
<td>Controller-based Automation (system overview, sample topologies)</td>
</tr>
<tr>
<td></td>
<td>Lenze controllers (product information, technical data)</td>
</tr>
<tr>
<td>System manuals</td>
<td>Visualisation (system overview/sample topologies)</td>
</tr>
<tr>
<td>Communication manuals</td>
<td>Bus systems</td>
</tr>
<tr>
<td>Online helps</td>
<td>• Controller-based Automation EtherCAT®</td>
</tr>
<tr>
<td></td>
<td>• Controller-based Automation CANopen®</td>
</tr>
<tr>
<td></td>
<td>• Controller-based Automation PROFIBUS®</td>
</tr>
<tr>
<td></td>
<td>• Controller-based Automation PROFINET®</td>
</tr>
<tr>
<td>Reference manuals</td>
<td>Lenze Controllers:</td>
</tr>
<tr>
<td>Online helps</td>
<td>• Controller 3200 C</td>
</tr>
<tr>
<td></td>
<td>• Controller c300</td>
</tr>
<tr>
<td></td>
<td>• Controller p300</td>
</tr>
<tr>
<td></td>
<td>• Controller p500</td>
</tr>
<tr>
<td>Software manuals</td>
<td>Lenze Engineering Tools:</td>
</tr>
<tr>
<td>Online helps</td>
<td>• »PLC Designer« (programming)</td>
</tr>
<tr>
<td></td>
<td>• »Engineer« (parameter setting, configuration, diagnostics)</td>
</tr>
<tr>
<td></td>
<td>• »VisiWinNET® Smart« (visualisation)</td>
</tr>
<tr>
<td></td>
<td>• »Backup &amp; Restore« (backup, restore, update)</td>
</tr>
</tbody>
</table>
1 About this documentation

More technical documentation for Lenze components

Further information on Lenze products which can be used in conjunction with Controller-based Automation can be found in the following sets of documentation:

<table>
<thead>
<tr>
<th>Design / configuration / technical data</th>
<th>Symbols:</th>
</tr>
</thead>
<tbody>
<tr>
<td>☐ Product catalogues</td>
<td>□ Printed documentation</td>
</tr>
<tr>
<td>• Controller-based Automation</td>
<td>□ PDF file / online help in the Lenze engineering tool</td>
</tr>
<tr>
<td>• Controllers</td>
<td></td>
</tr>
<tr>
<td>• Inverter Drives/Servo Drives</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Installation and wiring</td>
</tr>
<tr>
<td>☐ Mounting instructions</td>
<td></td>
</tr>
<tr>
<td>• Controllers</td>
<td></td>
</tr>
<tr>
<td>• Communication cards (MC-xxx)</td>
<td></td>
</tr>
<tr>
<td>• I/O system 1000 (EPM-Sxxx)</td>
<td></td>
</tr>
<tr>
<td>• Inverter Drives/Servo Drives</td>
<td></td>
</tr>
<tr>
<td>• Communication modules</td>
<td></td>
</tr>
<tr>
<td>☐ Hardware manuals</td>
<td></td>
</tr>
<tr>
<td>• Inverter Drives/Servo Drives</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Parameterisation / configuration / commissioning</td>
</tr>
<tr>
<td>☐ Online help/reference manuals</td>
<td></td>
</tr>
<tr>
<td>• Controllers</td>
<td></td>
</tr>
<tr>
<td>• Inverter Drives/Servo Drives</td>
<td></td>
</tr>
<tr>
<td>• I/O system 1000 (EPM-Sxxx)</td>
<td></td>
</tr>
<tr>
<td>☐ Online help/communication manuals</td>
<td></td>
</tr>
<tr>
<td>• Bus systems</td>
<td></td>
</tr>
<tr>
<td>• Communication modules</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sample applications and templates</td>
</tr>
<tr>
<td>☐ Online help / software manuals and reference manuals</td>
<td></td>
</tr>
<tr>
<td>• i700 application sample</td>
<td></td>
</tr>
<tr>
<td>• Application Samples 8400/9400</td>
<td></td>
</tr>
<tr>
<td>• FAST application template</td>
<td></td>
</tr>
<tr>
<td>• FAST technology modules</td>
<td></td>
</tr>
</tbody>
</table>

💡 Tip!
Current documentation and software updates with regard to Lenze products can be found in the download area at:

www.lenze.com

Target group
This documentation is intended for persons who commission and maintain a Controller-based automation system by means of a Lenze Controller and the »PLC Designer« engineering tool.

Information regarding the validity
The information provided in this documentation is valid for the Lenze "Controller-based Automation" system from release 3.0 onwards.

Screenshots/application examples
All screenshots in this documentation are application examples. Depending on the firmware version of the field devices and the software version of the Engineering tools installed (e.g. »PLC Designer« ), screenshots in this documentation may differ from the representation on the screen.
## 1.1 Document history

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.7</td>
<td>11/2016</td>
<td>TD17 Chapter updated:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Setting of CAN parameters and PDO mapping (p 52)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Setting of CAN parameters and PDO mapping (p 89)</td>
</tr>
<tr>
<td>6.6</td>
<td>08/2016</td>
<td>TD17 Chapter updated:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CAN wiring (p 20)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CAN interface of the c300/p300 controller (p 23)</td>
</tr>
<tr>
<td>6.5</td>
<td>10/2015</td>
<td>TD17 • Update for the Lenze “Controller-based Automation” system 3.12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• New: Chapter “CAN wiring” (p 20)</td>
</tr>
<tr>
<td>6.4</td>
<td>01/2015</td>
<td>TD17 Update for the Lenze “Controller-based Automation” system 3.9</td>
</tr>
<tr>
<td>6.3</td>
<td>04/2014</td>
<td>TD17 Update for the Lenze “Controller-based Automation” system 3.8</td>
</tr>
<tr>
<td>6.2</td>
<td>11/2013</td>
<td>TD17 Update for the Lenze “Controller-based Automation” system 3.6</td>
</tr>
<tr>
<td>6.1</td>
<td>03/2013</td>
<td>TD17 Update for the Lenze “Controller-based Automation” system 3.5</td>
</tr>
<tr>
<td>6.0</td>
<td>11/2012</td>
<td>TD17 • General corrections</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• New layout</td>
</tr>
<tr>
<td>5.3</td>
<td>07/2012</td>
<td>TD17 • Update for the Lenze “Controller-based Automation” system 3.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Information with regard to the ECS servo system and »GDC« removed.</td>
</tr>
<tr>
<td>5.2</td>
<td>12/2011</td>
<td>TD17 Update for the Lenze “Controller-based Automation” system 3.2</td>
</tr>
<tr>
<td>5.1</td>
<td>03/2011</td>
<td>TD17 • Chapter “Parallel operation of two synchronised CAN buses” (p 84)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• supplemented.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• SoftMotion settings for Servo Drives 9400 and ECSxM supplemented.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• References to Lenze sample projects for CANopen Logic field devices (device</td>
</tr>
<tr>
<td></td>
<td></td>
<td>application + PLC program) added.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Commissioning the CANopen Logic bus (p 38)</td>
</tr>
<tr>
<td>5.0</td>
<td>10/2010</td>
<td>TD17 Commissioning and configuration with the Lenze »PLC Designer« V3.x</td>
</tr>
<tr>
<td>4.0</td>
<td>10/2009</td>
<td>TD17 General update</td>
</tr>
<tr>
<td>3.0</td>
<td>06/2009</td>
<td>TD17 General update</td>
</tr>
<tr>
<td>2.0</td>
<td>09/2008</td>
<td>TD17 Chapter “Mixed operation of CANopen and EtherCAT” (p 94) added.</td>
</tr>
<tr>
<td>1.0</td>
<td>06/2008</td>
<td>TD17 First edition</td>
</tr>
</tbody>
</table>
1 About this documentation

1.2 Conventions used

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

<table>
<thead>
<tr>
<th>Type of information</th>
<th>Highlighting</th>
<th>Examples/notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spelling of numbers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decimal</td>
<td>Normal spelling</td>
<td>Example: 1234</td>
</tr>
<tr>
<td>Decimal separator</td>
<td>Point</td>
<td>The decimal point is always used. For example: 1234.56</td>
</tr>
<tr>
<td>Hexadecimal</td>
<td>0x[0 ... 9, A ... F]</td>
<td>Example: 0x60F4</td>
</tr>
</tbody>
</table>
| Binary                      | 0b[0, 1]      | Example: ‘0b0110’
|                             |               | Example: ‘0b0110.0100’                             |
| Text                        |               |                                                     |
| Program name                | « »           | PC software                                         |
|                             |               | Example: Lenze »Engineer«                           |
| Window                      | italics       | The *message window* / The *Options* dialog box ...  |
| Variable names              |               | Setting *bEnable* to TRUE...                         |
| Control element             | bold          | The *OK* button ... / The *Copy* command ... / The *Name* input field ... |
| Sequence of menu commands   |               | If several successive commands are required for executing a function, the individual commands are separated from each other by an arrow: Select the command *File* ➤ *Open* to... |
| Shortcut                    | <bold>        | Use <F1> to open the online help.                   |
|                             |               | If a key combination is required for a command, a "+" is placed between the key identifiers: With <Shift>+<ESC>... |
| Program code                | Courier       | IF var1 < var2 THEN  |
|                             |               |   a = a + 1                                           |
|                             |               | END IF                                               |
| Keyword                     | Courier bold  |                                                     |
| Hyperlink                   | underlined    | Optically highlighted reference to another topic. Can be activated with a mouse-click in this documentation. |
| Icons                       |               |                                                     |
| Page reference              | (8)           | Optically highlighted reference to another page. Can be activated with a mouse-click in this documentation. |
| Step-by-step instructions   |               | Step-by-step instructions are indicated by a pictograph. |
### Terminology used

<table>
<thead>
<tr>
<th>Term</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>Parameter for parameterising or monitoring the field device. The term is also referred to as “index” in common usage.</td>
</tr>
<tr>
<td>Controllers</td>
<td>The Controller is the central component of the Lenze automation system which control the motion sequences by means of the application software. The Controller communicates with the field devices (inverters) via the fieldbus.</td>
</tr>
<tr>
<td>Engineering PC</td>
<td>The Engineering PC and the Engineering tools installed serve to configure and parameterise the system “Controller-based Automation”. The Engineering PC communicates with the controller via Ethernet.</td>
</tr>
<tr>
<td>Engineering tools</td>
<td>Software solutions for easy engineering in all phases which serve to commission, configure, parameterise and diagnose the Lenze automation system.</td>
</tr>
<tr>
<td>FAST</td>
<td>By default, the Lenze FAST application software is installed on the Lenze Controller in the &quot;FAST runtime&quot; version with &quot;FAST Motion&quot; for the central control of PLC applications.</td>
</tr>
<tr>
<td>Fieldbus node</td>
<td>Devices integrated in the bus system as, for instance, Controller and inverter</td>
</tr>
<tr>
<td>Field device</td>
<td>Programmable Logic Controller</td>
</tr>
<tr>
<td>Subcode</td>
<td>If a code contains several parameters, they are stored in &quot;subcodes&quot;. In the documentation, the slash &quot;/&quot; is used as a separator between the code and the subcode (e.g. &quot;C00118/3&quot;).</td>
</tr>
<tr>
<td>Bus systems</td>
<td></td>
</tr>
<tr>
<td>CAN</td>
<td>CAN (Controller Area Network) is an asynchronous, serial fieldbus system.</td>
</tr>
<tr>
<td>CANopen</td>
<td>CANopen® is a communication protocol based on CAN. The Lenze system bus (CAN on board) operates with a subset of this communication protocol. CANopen® is a registered community trademark of the CAN user organisation CIA® (CAN in Automation e. V.).</td>
</tr>
<tr>
<td>ETHERNET</td>
<td>Ethernet specifies the software (protocols) and hardware (cables, plugs, etc.) for wired data networks. In the form of “Industrial Ethernet”, the Ethernet standard is used in industrial production systems. On the basis of IEEE 802.3, standard Ethernet is specified by the Institute of Electrical and Electronics Engineers (IEEE), USA.</td>
</tr>
<tr>
<td>EtherCAT®</td>
<td>EtherCAT® (Ethernet for Controller and Automation Technology) is an Ethernet-based fieldbus system which fulfils the application profile for industrial real-time systems. EtherCAT® is a registered trademark and patented technology, licenced by Beckhoff Automation GmbH, Germany.</td>
</tr>
<tr>
<td>EtherNet/IP™</td>
<td>EtherNet/IP™ (EtherNet Industrial Protocol) is an Ethernet-based fieldbus system that uses Common Industrial Protocol™ (CIP™) to exchange data. EtherNet/IP™ and Common Industrial Protocol™ (CIP™) are brand labels and patented technologies, licensed by the ODVA user organisation (Open DeviceNet Vendor Association), USA.</td>
</tr>
</tbody>
</table>
1 About this documentation

1.4 Definition of the notes used

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

Safety instructions
Layout of the safety instructions:

⚠️ Pictograph and signal word!
(characterise the type and severity of danger)

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

<table>
<thead>
<tr>
<th>Pictograph</th>
<th>Signal word</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>⚠️ Danger!</td>
<td>Danger of personal injury through dangerous electrical voltage Reference to an imminent danger that may result in death or serious personal injury if the corresponding measures are not taken.</td>
<td></td>
</tr>
<tr>
<td>⚠️ Danger!</td>
<td>Danger of personal injury through a general source of danger Reference to an imminent danger that may result in death or serious personal injury if the corresponding measures are not taken.</td>
<td></td>
</tr>
<tr>
<td>STOP</td>
<td>Danger of property damage Reference to a possible danger that may result in property damage if the corresponding measures are not taken.</td>
<td></td>
</tr>
</tbody>
</table>

Application notes

<table>
<thead>
<tr>
<th>Pictograph</th>
<th>Signal word</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Note!</td>
<td>Important note to ensure trouble-free operation</td>
<td></td>
</tr>
<tr>
<td>Tip!</td>
<td>Useful tip for easy handling</td>
<td></td>
</tr>
<tr>
<td>✗</td>
<td>Reference to another document</td>
<td></td>
</tr>
</tbody>
</table>
2 Safety instructions

Please observe the safety instructions in this documentation when you want to commission an automation system or a plant with a Lenze Controller.

The device documentation contains safety instructions which must be observed!
Read the documentation supplied with the components of the automation system carefully before you start commissioning the Controller and the connected devices.

Danger!

High electrical voltage
Injury to persons caused by dangerous electrical voltage
Possible consequences
Death or severe injuries
Protective measures
Switch off the voltage supply before working on the components of the automation system.
After switching off the voltage supply, do not touch live device parts and power terminals immediately because capacitors may be charged.
Observe the corresponding information plates on the device.

Danger!

Injury to persons
Risk of injury is caused by ...
• unpredictable motor movements (e.g. unintended direction of rotation, too high velocities or jerky movement);
• impermissible operating states during the parameterisation while there is an active online connection to the device.
Possible consequences
Death or severe injuries
Protective measures
• If required, provide systems with installed inverters with additional monitoring and protective devices according to the safety regulations valid in each case (e.g. law on technical equipment, regulations for the prevention of accidents).
• During commissioning, maintain an adequate safety distance to the motor or the machine parts driven by the motor.
Stop!

Damage or destruction of machine parts
Damage or destruction of machine parts can be caused by …
- Short circuit or static discharges (ESD);
- unpredictable motor movements (e.g. unintended direction of rotation, too high velocities or jerky movement);
- impermissible operating states during the parameterisation while there is an active online connection to the device.

Protective measures
- Always switch off the voltage supply before working on the components of the automation system.
- Do not touch electronic components and contacts unless ESD measures were taken beforehand.
- If required, provide systems with installed inverters with additional monitoring and protective devices according to the safety regulations valid in each case (e.g. law on technical equipment, regulations for the prevention of accidents).
3 Controller-based Automation: Central motion control

The Lenze "Controller-based Automation" system serves to create complex automation solutions with central motion control. Here, the Controller is the control centre of the system.

System structure of the Controller-based Automation

[Example: CANopen with the 3231 C Lenze Controller (I/O system 1000 and Servo Drive 9400 as slaves)]
Lenze provides especially coordinated system components:

- **Engineering software**
  The [Lenze Engineering tools](http://www.lenze.com) on your Engineering PC (Windows® operating system ) serve to parameterise, configure and diagnose the system. The Engineering PC communicates with the Controller via Ethernet.
  The Lenze engineering tools are available for download at: [www.lenze.com](http://www.lenze.com) → Download → Software Downloads

- **Controllers**
  The Lenze Controller is available as Panel Controller with integrated touch display and as Cabinet Controller in control cabinet design.
  Cabinet Controllers provide a direct coupling of the I/O system 1000 via the integrated backplane bus.

- **Bus systems**
  EtherCAT is the standard "on-board" bus system of the Controller-based Automation. EtherCAT enables the control of all nodes on one common fieldbus.
  Optionally, CANopen, PROFIBUS and PROFINET can be used as extended topologies.
  With Controllers 3200 C and p500 it is also possible to use EtherNet/IP via the Ethernet interfaces.
  Controllers c300 and p300 are provided with an "on board" CANopen interface (in addition to EtherCAT).

- **Inverter (e.g. Servo-Inverter i700)**

"Application software" of the Lenze Controllers

The "application software" of the Lenze Controllers enables the control and/or visualisation of motion sequences.

**FAST technology modules** provide for an easy development of a modular machine control in the »PLC Designer«.

The following "Application Software" versions are available:

- **"FAST Runtime"**
  The sequence control takes place (by logically combined control signals) in the Controller.
  The motion control takes place in the inverter.

- **"FAST Motion"**
  The sequence control and the motion control take place in the controller.
  The inverter merely serves as actuating drive.
  Motion applications make special demands on the cycle time and real-time capability of the bus system between the Controller and the subordinate fieldbus nodes. This is the case, for instance, if the nodes are to be traversed in a synchronised way or position setpoints are to be transferred.

- **"Visualisation"**
  The optional visualisation of the automation system can be used separately or additionally to "FAST Runtime" or "FAST Motion".
  For this purpose, an external monitor panel/display can be connected to the Cabinet Controller 3231 C/3241 C/3251 C.
Fieldbus communication

The Lenze controllers have different interfaces for fieldbus communication:

<table>
<thead>
<tr>
<th>Range</th>
<th>Cabinet Controller</th>
<th>Panel Controller</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>c300</td>
<td>3200 C series</td>
</tr>
<tr>
<td>Interfaces (on board)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ethernet</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>EtherNet/IP</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>EtherCAT</td>
<td>1 (^1)</td>
<td>1</td>
</tr>
<tr>
<td>CANopen</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>Optional interfaces (communication cards)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CANopen</td>
<td>-</td>
<td>●</td>
</tr>
<tr>
<td>MC-CAN2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROFIBUS master</td>
<td>-</td>
<td>●</td>
</tr>
<tr>
<td>MC-PBM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROFIBUS slave</td>
<td>-</td>
<td>●</td>
</tr>
<tr>
<td>MC-PBS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROFINET device</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>MC-PND</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ethernet</td>
<td>-</td>
<td>●</td>
</tr>
<tr>
<td>MC-ETH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Serial interfaces</td>
<td>-</td>
<td>●</td>
</tr>
<tr>
<td>MC-ISI</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1) Only the master functionality is supported.
2) Up to release 3.9: "EL 100 CAN" driver / from release 3.10: "Lenze CAN driver"

Ethernet interface

The Ethernet interface serves to connect the Engineering PC or to create line topologies (no integrated switch for Controller c300/p300).

With Controllers 3200 C and p500, the Ethernet interfaces also provide for EtherNet/IP communication.
The control technology based on CANopen allows for the integration of all Lenze device series provided with the Lenze system bus (CAN on board).

In order to extend the existing limits of the CAN bus, several CAN lines synchronised with each other can be used. The number of CAN lines available depends on the equipment of the Lenze Controller in each case.

The maximum possible number of nodes on a CAN line depends on the baud rate and the cycle time set.

**Example:** In the case of a cycle time of 1 ms and a baud rate of 1 Mbps, three nodes with a setpoint PDO and an actual value PDO, respectively, can be actuated on the CAN bus.

**Tip!**

Detailed information on CAN/CANopen can be found on the website of the CAN User Organization CiA (CAN in Automation):

[www.can-cia.org](http://www.can-cia.org)
4 System bus (CAN) / CANopen
4.1 CANopen (Logic) / CANopen (Motion)

Example: CANopen (Logic/Motion) with the 3231 C controller (I/O system 1000 and Servo Drive 9400 as slaves)

Due to the requirements regarding the real time behaviour of the fieldbus system and due to its limited transfer capacity, it is useful to operate Logic and Motion devices on separate CAN phases if CANopen is used – on a logic bus and a motion bus.

The Lenze Controllers ... 
• with the MC-CAN2 communication card (19) have two CAN interfaces for CANopen (Logic) and CANopen (Motion);
• can also be used as CAN slaves.

Depending on the required number of Motion nodes and bus cycle time, up to 2 Motion bus lines can be created.

Tip!
A sample project for operation of a 3200 C controller as CAN slave can be found in the "Download" area at www.Lenze.com:
"Application Knowledge Base": All articles ➔ Application Ideas Pool ➔ Controller 3200 C
4 System bus (CAN) / CANopen

4.2 Field devices

The Lenze automation system supports the following Logic/Motion components:

<table>
<thead>
<tr>
<th>Field devices</th>
<th>System bus (CAN/CANopen)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Logic</td>
</tr>
<tr>
<td><strong>Controllers</strong></td>
<td></td>
</tr>
<tr>
<td>Controller 32xx C</td>
<td>●</td>
</tr>
<tr>
<td>Controller c300</td>
<td>●</td>
</tr>
<tr>
<td>Controller p300</td>
<td>●</td>
</tr>
<tr>
<td>Controller p500</td>
<td>●</td>
</tr>
<tr>
<td><strong>Servo Drives 9400</strong></td>
<td></td>
</tr>
<tr>
<td>HighLine 1)</td>
<td>●</td>
</tr>
<tr>
<td>HighLine with CiA402</td>
<td>●</td>
</tr>
<tr>
<td>PLC</td>
<td>●</td>
</tr>
<tr>
<td>regenerative power supply module commissioning guidelines</td>
<td>●</td>
</tr>
<tr>
<td><strong>Inverter Drives 8400</strong></td>
<td></td>
</tr>
<tr>
<td>Baseline</td>
<td>●</td>
</tr>
<tr>
<td>StateLine</td>
<td>●</td>
</tr>
<tr>
<td>HighLine</td>
<td>●</td>
</tr>
<tr>
<td>TopLine</td>
<td>●</td>
</tr>
<tr>
<td>protec</td>
<td>●</td>
</tr>
<tr>
<td>motec</td>
<td>●</td>
</tr>
<tr>
<td><strong>I/O system 1000</strong></td>
<td></td>
</tr>
<tr>
<td>EPM-Sxxx</td>
<td>●</td>
</tr>
</tbody>
</table>

1) with technology application (TA)
4.3 CANopen hardware for Lenze Controllers

MC-CAN2 communication card

The MC-CAN2 communication card serves to connect a Lenze Controller to the CAN bus system. The card provides two independent bus lines.

![MC-CAN2 communication card](image)

Technical data of the MC-CAN2 communication card (23)

Application

The MC-CAN2 communication card is installed in the corresponding slot of the Lenze Controller.

Example: Lenze Controller 3231 C with MC-CAN2 communication card

<table>
<thead>
<tr>
<th>MC-CAN2</th>
<th>MC-CAN2 communication card</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAN1</td>
<td>Connections for two bus lines</td>
</tr>
<tr>
<td>CAN2</td>
<td>• CAN1: CANopen (Logic and/or Motion)</td>
</tr>
<tr>
<td></td>
<td>• CAN2: CANopen (Logic and/or Motion)</td>
</tr>
</tbody>
</table>
4.4 CAN wiring

**Note!**
Connect a 120 Ω terminating resistor to the first and last node.

![Connection plan for the CAN bus](image1)

- **A1**: Node 1
- **A2**: Node 2
- **An**: Node n
- **CG**: CAN-Ground
- **LO**: CAN-Low
- **HI**: CAN-High
- **R**: Terminating resistor 120 Ω

- **Bus cable specification**
- **Connection: 9-pole SUB-D plug**
- **Connection: 5-pole Phoenix Combicon socket**
- **Bus cable length**

**Shielding**

![CAN cable shield connection over cable clamp in the control cabinet](image2)
4.5 Lenze Engineering tools

The Lenze Engineering tools enable the configuration and operation of controller-based Lenze automation systems according to individual requirements.

Use the corresponding Engineering tool applicable to the field device.

»EASY Navigator«: Starting the suitable Engineering tool

The Lenze Engineering software consists of the Engineering tools optimised for the respective Engineering stage.

The »EASY Navigator« represents the Lenze Engineering tools installed on the Engineering PC. Start the desired Engineering tool via the corresponding button:

- simplifies orientation for selecting the suitable Engineering tool;
- allows for the simple start of the required Engineering tool (depending on the application):

<table>
<thead>
<tr>
<th>What would you like to do?</th>
<th>Button</th>
<th>Engineering tool</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Programming</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Program the controller</td>
<td>![PLC]</td>
<td>»PLC Designer«</td>
</tr>
<tr>
<td>• Parameterise/commission the Servo-Inverter i700</td>
<td>![PLC]</td>
<td></td>
</tr>
<tr>
<td>• Parameterise the I/O system 1000</td>
<td>![PLC]</td>
<td></td>
</tr>
<tr>
<td><strong>Parameterising/configuring the inverter</strong></td>
<td>![arm]</td>
<td>»Engineer«</td>
</tr>
<tr>
<td>• Parameterising and configuring the automation/drive system</td>
<td>![arm]</td>
<td></td>
</tr>
<tr>
<td>• Parameterising Inverter Drives 8400/Servo Drives 9400</td>
<td>![arm]</td>
<td></td>
</tr>
<tr>
<td><strong>Visualisation</strong></td>
<td>![monitor]</td>
<td>»VisiWinNET«</td>
</tr>
<tr>
<td>• Visualising the applications of the automation system</td>
<td>![monitor]</td>
<td></td>
</tr>
<tr>
<td>• Creating the visualisation/user interfaces</td>
<td>![monitor]</td>
<td></td>
</tr>
<tr>
<td><strong>Online diagnostics</strong></td>
<td>![EASY Starter]</td>
<td>»EASY Starter«</td>
</tr>
<tr>
<td>• Easy online diagnostics of the controllers (from »EASY Starter« V1.2) and other Lenze devices</td>
<td>![EASY Starter]</td>
<td>(reading parameters)</td>
</tr>
<tr>
<td><strong>Online parameterisation</strong></td>
<td>![EASY Starter]</td>
<td>»EASY Starter«</td>
</tr>
<tr>
<td>• Online parameterisation/commissioning of Lenze devices</td>
<td>![EASY Starter]</td>
<td>(reading/writing parameters)</td>
</tr>
<tr>
<td>• Direct online parameterisation when the online connection to the Lenze devices is active.</td>
<td>![EASY Starter]</td>
<td></td>
</tr>
</tbody>
</table>

Further Engineering tools that are not called via the »EASY Navigator« are:
- »WebConfig« (web-based parameterisation, configuration, and online diagnostics)
- »Backup & Restore« (data backup/recovery, software update).
## 5 Technical data

### 5.1 General data

<table>
<thead>
<tr>
<th>Range</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication profile</td>
<td>CANopen (DS301, V4.02)</td>
</tr>
<tr>
<td>Standards</td>
<td>CAN, ISO 11898 / EN 50325-4</td>
</tr>
<tr>
<td>Network topology</td>
<td>Line, terminated at both ends with 120 Ω</td>
</tr>
<tr>
<td></td>
<td>(e.g. terminated with Sub-D plug of type EW20046)</td>
</tr>
<tr>
<td>Max. number of nodes</td>
<td>127</td>
</tr>
<tr>
<td>Adjustable node addresses</td>
<td>1 ... 127</td>
</tr>
<tr>
<td></td>
<td>(adjustable for Lenze devices via DIP switches)</td>
</tr>
<tr>
<td>Baud rates [kbps]</td>
<td>• 10</td>
</tr>
<tr>
<td></td>
<td>• 20</td>
</tr>
<tr>
<td></td>
<td>• 50</td>
</tr>
<tr>
<td></td>
<td>• 125</td>
</tr>
<tr>
<td></td>
<td>• 250</td>
</tr>
<tr>
<td></td>
<td>• 500</td>
</tr>
<tr>
<td></td>
<td>• 1000</td>
</tr>
<tr>
<td>Parameter data</td>
<td>Max. 10 client and server SDO channels with 1 ... 8 bytes</td>
</tr>
<tr>
<td>Cycle time - Motion/CNC task</td>
<td>1 ... 16 ms</td>
</tr>
<tr>
<td>Number of drives/ms on the Motion bus</td>
<td>Max. 3 drives/ms</td>
</tr>
<tr>
<td>Signal propagation delay drive ➔ controller ➔ drive</td>
<td>4 cycles</td>
</tr>
<tr>
<td>Cross communication</td>
<td>Only possible with CANopen (Logic)</td>
</tr>
<tr>
<td></td>
<td>In the case of CANopen (Motion), communication is executed centrally via the Lenze Controller.</td>
</tr>
<tr>
<td>Number of DI + DO (bits/ms)</td>
<td>384 (max. 6 PDOs/ms on the Logic bus)</td>
</tr>
<tr>
<td>Cycle synchronisation with locked PLL (jitter)</td>
<td>+/-10 μs</td>
</tr>
</tbody>
</table>
5.2 Technical data of the MC-CAN2 communication card

<table>
<thead>
<tr>
<th>Range</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type within the network</td>
<td>Master or slave</td>
</tr>
<tr>
<td>Max. number of nodes</td>
<td>63</td>
</tr>
<tr>
<td>Max. baud rate</td>
<td>1000 kbps</td>
</tr>
<tr>
<td>Connection</td>
<td>SUB-D, 9-pole plug</td>
</tr>
</tbody>
</table>

**Connection: 9-pole SUB-D plug**

<table>
<thead>
<tr>
<th>View</th>
<th>Pin</th>
<th>Assignment</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>Free</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>LO</td>
<td>CAN-LOW</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>CG</td>
<td>CAN-Ground</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>Free</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>Free</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>CG</td>
<td>CAN-Ground</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>HI</td>
<td>CAN-HIGH</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>Free</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>Free</td>
<td>-</td>
</tr>
</tbody>
</table>

5.3 CAN interface of the c300/p300 controller

**Connection: 5-pole Phoenix Combicon socket**

<table>
<thead>
<tr>
<th>View</th>
<th>Pin</th>
<th>Assignment</th>
<th>Description</th>
<th>Plug</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>CG</td>
<td>CAN-Ground</td>
<td>MSTB 2.5/5-STF-5.8 Phoenix Combicon plug</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>LO</td>
<td>CAN-LOW</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Free</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>HI</td>
<td>CAN-HIGH</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>Free</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

5.4 Bus cable specification

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

<table>
<thead>
<tr>
<th>CAN cables according to ISO 11898-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cable type</td>
</tr>
<tr>
<td>Twisted in pairs with shield</td>
</tr>
<tr>
<td>Impedance</td>
</tr>
<tr>
<td>120 Ω (95 ... 140 Ω)</td>
</tr>
<tr>
<td>Cable resistance/cross-section</td>
</tr>
<tr>
<td>Cable length ≤ 300 m:</td>
</tr>
<tr>
<td>≤ 70 mΩ/m / 0.25 ... 0.34 mm² (AWG22)</td>
</tr>
<tr>
<td>Cable length 301 ... 1000 m:</td>
</tr>
<tr>
<td>≤ 40 mΩ/m / 0.5 mm² (AWG20)</td>
</tr>
<tr>
<td>Signal propagation delay</td>
</tr>
<tr>
<td>≤ 5 ns/m</td>
</tr>
</tbody>
</table>
5.5 Bus cable length

**Note!**

- It is absolutely necessary to comply with the permissible cable lengths.
- Observe the reduction of the total cable length due to the signal delay of the repeater.
- Use of repeaters (Lu 25)
- If the total cable lengths of the nodes are different at the same baud rate, the smaller value must be used to determine the maximum cable length.

5.5.1 Total cable length

The total cable length is also specified by the baud rate.

<table>
<thead>
<tr>
<th>Baud rate [kbps]</th>
<th>Max. bus length [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Servo Drives 9400</td>
</tr>
<tr>
<td>10</td>
<td>8000</td>
</tr>
<tr>
<td>20</td>
<td>4000</td>
</tr>
<tr>
<td>50</td>
<td>1500</td>
</tr>
<tr>
<td>125</td>
<td>600</td>
</tr>
<tr>
<td>250</td>
<td>275</td>
</tr>
<tr>
<td>500</td>
<td>110</td>
</tr>
<tr>
<td>1000</td>
<td>13</td>
</tr>
</tbody>
</table>

5.5.2 Segment cable length

Repeaters divide the total cable length into segments. The segment cable length is defined by the cable cross-section and the number of nodes per segment. Without a repeater, the segment cable length corresponds to the total cable length.

<table>
<thead>
<tr>
<th>Max. number of nodes per segment</th>
<th>Cable cross-section</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.25 mm²</td>
</tr>
<tr>
<td>2</td>
<td>240 m</td>
</tr>
<tr>
<td>5</td>
<td>230 m</td>
</tr>
<tr>
<td>10</td>
<td>230 m</td>
</tr>
<tr>
<td>20</td>
<td>210 m</td>
</tr>
<tr>
<td>32</td>
<td>200 m</td>
</tr>
<tr>
<td>63</td>
<td>170 m</td>
</tr>
<tr>
<td>100</td>
<td>150 m</td>
</tr>
</tbody>
</table>
5.5.3 Use of repeaters

Compare the values from the tables Total cable length (24) and Segment cable length (24). If the detected segment cable length is smaller than the total cable length to be achieved, repeaters must be used.

Example: Detecting cable lengths / number of repeaters

<table>
<thead>
<tr>
<th>Given</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cable cross-section</td>
<td>0.5 mm², according to Bus cable specification (23)</td>
</tr>
<tr>
<td>Number of nodes</td>
<td>127</td>
</tr>
<tr>
<td>Repeater</td>
<td>Lenze repeater, type 2176 (cable reduction: 30 m)</td>
</tr>
</tbody>
</table>

If using the maximum number of nodes (127), the following cable lengths/number of repeaters from the specifications have to be observed:

<table>
<thead>
<tr>
<th>Baud rate [kbps]</th>
<th>10</th>
<th>20</th>
<th>50</th>
<th>125</th>
<th>250</th>
<th>500</th>
<th>800</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. cable length [m]</td>
<td>8000</td>
<td>3900</td>
<td>1500</td>
<td>630</td>
<td>290</td>
<td>110</td>
<td>40</td>
<td>17</td>
</tr>
<tr>
<td>Segment cable length [m]</td>
<td>270</td>
<td>270</td>
<td>270</td>
<td>270</td>
<td>270</td>
<td>110</td>
<td>40</td>
<td>17</td>
</tr>
<tr>
<td>Number of repeaters</td>
<td>33</td>
<td>16</td>
<td>6</td>
<td>2</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
5  Technical data
5.5  Bus cable length

Example: Check use of repeater

<table>
<thead>
<tr>
<th>Given</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Baud rate</td>
<td>125 kbps</td>
</tr>
<tr>
<td>Cable cross-section</td>
<td>0.5 mm²</td>
</tr>
<tr>
<td>Number of nodes</td>
<td>28</td>
</tr>
<tr>
<td>Cable length</td>
<td>450 m</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test step</th>
<th>Cable length</th>
<th>See table ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Total cable length at 125 kbps: 630 m</td>
<td>Total cable length (24)</td>
</tr>
<tr>
<td>2</td>
<td>Segment cable length for 28 nodes and a cable cross-section of 0.5 mm²: 360 m</td>
<td>Segment cable length (24)</td>
</tr>
<tr>
<td>3</td>
<td>Comparison: The detected segment cable length is smaller than the total cable length of 450 m to be achieved.</td>
<td></td>
</tr>
</tbody>
</table>

Conclusion:
- It is not possible to use a cable length of 450 m without using a repeater.
- After 360 m (test step 2) a repeater has to be used.

Result:
- The Lenze repeater, type 2176 (cable reduction: 30 m), is used
- Calculation of the maximum cable length:
  - First segment: 360 m
  - Second segment: 360 m (see table Segment cable length (24)) minus 30 m (cable reduction for a repeater)
  - Max. achievable cable length with a repeater: 690 m
- Now it is possible to use the required cable length.
Create an overview screen of the planned CANopen network with all field devices to be implemented. Start with the Lenze Controller and arrange the other field devices below it (see Example of an overview screen (Lu 29)).

Provide the following data for each device:

<table>
<thead>
<tr>
<th>Device type</th>
<th>Type designation of the field device</th>
</tr>
</thead>
</table>
| **Used CAN interface of the device**             | • The functionality of the two available CAN interfaces is identical. Both Logic and Motion devices can be connected. The combination of Logic and Motion on an interface is possible as well.  
  • If possible, the Logic and Motion devices should be installed on different CAN lines:  
    • The requirements of the Motion devices regarding the synchronicity of the bus are higher.  
    • Shorter cycle times are needed.  
    • The data volume to be transferred is larger.  
  CANopen (Logic) / CANopen (Motion) (Lu 17)      |
| **Unambiguous CAN node address**                 | • If system bus (CAN) devices are used, max. 63 nodes/node addresses are possible.  
  • With CANopen-compliant devices, up to 127 nodes/node addresses are possible.  
  **Note:** Do not use the node address 1, in order to avoid unintentional mistakes and conflicts with a device containing the factory adjustment. |
| **Baud rate**                                    | • The baud rate applies to all nodes of the CANopen network.  
  • 50, 125, 250 and 500 kbps are supported by all device types of the system.  
  • Observe the connection between bus cable length and baud rate.  
  Bus cable length (Lu 24)                                                                            |
| **Master role of the device (NMT master/sync master)** | • An NMT master sets itself and then the NMT slaves to the “Operational” state. In this state, process data can be communicated. Generally, there can be an optional number of NMT masters on one CANopen bus.  
  • A Sync master cyclically sends a sync telegram providing for an exactly simultaneous processing of process data and/or a simultaneous task start in all sync receivers.  
  • Via CAN synchronisation, the Lenze Controller can influence the exact time of the following events in the field device:  
    • Acceptance and transmission of sync-controlled PDOs  
    • Starting time of the task of the application (only possible in 9400)  
  • You only need to use CAN synchronisation on the Logic bus if an exact simultaneity in the range of milliseconds is of importance. A mere operating periphery (operator button, control lamps, etc.) does not require CAN synchronisation. |
| **CAN objects and COB-IDs**                      | • Plan your COB-IDs according to the CANopen DS301 communication profile. This convention is optimised for the communication with a central master device.  
  COB-IDs acc. to DS301 (Lu 28)  
  • Up to 4 PDOs per device can be identified with this scheme. If you require more, e.g. for a modular I/O system with more than 8 modules, you can add them later.  
  • You can easily assign the node during the bus diagnostics by means of the COB-IDs.  
  • COB-ID = basic identifier + node address
6 Planning the CANopen network

6.1 COB-IDs acc. to DS301

Please observe ...
the device-specific information on the CAN configuration in the documentation for the field devices to be implemented.

6.1 COB-IDs acc. to DS301

<table>
<thead>
<tr>
<th>Object</th>
<th>Direction</th>
<th>Basic identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>from the drive</td>
<td>to the drive</td>
</tr>
<tr>
<td>NMT</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Sync</td>
<td></td>
<td>128</td>
</tr>
<tr>
<td>Time Stamp</td>
<td></td>
<td>256</td>
</tr>
<tr>
<td>Emergency</td>
<td></td>
<td>128</td>
</tr>
<tr>
<td>PDO1 (Process data channel 1)</td>
<td></td>
<td>384</td>
</tr>
<tr>
<td>PDO2 (Process data channel 2)</td>
<td></td>
<td>640</td>
</tr>
<tr>
<td>PDO3 (Process data channel 3)</td>
<td></td>
<td>896</td>
</tr>
<tr>
<td>PDO4 (Process data channel 4)</td>
<td></td>
<td>1152</td>
</tr>
<tr>
<td>SDO (Parameter data channel 1)</td>
<td></td>
<td>1408</td>
</tr>
<tr>
<td>NMT Error Control</td>
<td></td>
<td>1792</td>
</tr>
</tbody>
</table>

Note!
In Lenze system bus (CAN) devices, two SDO channels are permanently active, in CANopen devices, only one by default.
When using CANopen devices, activate a second SDO channel for access of the »Engineer«. Otherwise communication with the device will be interfered if you go online with the »Engineer« while the Lenze Controller has access as well.

The COB-IDs for your CANopen network can be calculated according to the following formula:

\[
\text{COB-ID} = \text{basic identifier} + \text{node address}
\]

- Basic identifier - 9400 Servo Drives ([31]
- Basic identifier - 8400 Inverter Drives ([32]
- Basic identifier - I/O system 1000 (EPM-Sxxx) ([33)]
6 Planning the CANopen network

6.2 Example of an overview screen

The illustration shows you an example of an overview screen for planning a CANopen network:

![Diagram](image.png)

### 10 Controller - CAN1
- NMT Master
- Sync Master

### 11 9400 HighLine - CAN on board

<table>
<thead>
<tr>
<th></th>
<th>Rx</th>
<th>hex</th>
<th>Tx</th>
<th>hex</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDO 1</td>
<td>523</td>
<td>0x20D</td>
<td>395</td>
<td>0x1B7</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>773</td>
<td>0x3CB</td>
<td>851</td>
<td>0x53B</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1035</td>
<td>0x40B</td>
<td>907</td>
<td>0x58B</td>
<td></td>
</tr>
<tr>
<td>SDO 1</td>
<td>1547</td>
<td>0x60B</td>
<td>1415</td>
<td>0x89B</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1811</td>
<td>0x64B</td>
<td>1483</td>
<td>0x5CB</td>
<td>to activate</td>
</tr>
</tbody>
</table>

### 12 ECSxP supply module - CAN on board

<table>
<thead>
<tr>
<th></th>
<th>Rx</th>
<th>hex</th>
<th>Tx</th>
<th>hex</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDO 1</td>
<td>624</td>
<td>0x23C</td>
<td>396</td>
<td>0x1B6</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>like PDO 1</td>
</tr>
<tr>
<td>SDO 1</td>
<td>1648</td>
<td>0x64C</td>
<td>1420</td>
<td>0x88C</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1812</td>
<td>0x64C</td>
<td>1484</td>
<td>0x5CC</td>
<td></td>
</tr>
</tbody>
</table>
6. Planning the CANopen network

6.3 Device specifications of the field devices

When planning your CANopen network, consider the device specifications of the implemented field devices.

Overview of the device specifications for operation with a Lenze Controller

<table>
<thead>
<tr>
<th></th>
<th>Servo Drives 9400</th>
<th>Inverter Drives 8400</th>
<th>I/O system 1000 (EPM-Sxxx)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAN interface</td>
<td>• on board</td>
<td>on board</td>
<td>on board</td>
</tr>
<tr>
<td></td>
<td>• CANopen module</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Available PDOs</td>
<td>4 Transmit (Tx) +</td>
<td>3 Transmit (Tx) +</td>
<td>10 Transmit (Tx) +</td>
</tr>
<tr>
<td></td>
<td>4 Receive (Rx)</td>
<td>3 Receive (Rx)</td>
<td>10 Receive (Rx)</td>
</tr>
<tr>
<td>Can unused PDOs be</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>deactivated?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Can PDO COB-IDs be</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>freely selected?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Can PDO transfer</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>characteristics be</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>adjusted?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Available SDO channels</td>
<td>1 ex works (fixed),</td>
<td>2 ex works (fixed),</td>
<td>1 ex works (fixed),</td>
</tr>
<tr>
<td></td>
<td>9 further can be</td>
<td>1 more can be</td>
<td>1 more can be</td>
</tr>
<tr>
<td>activated</td>
<td>activated</td>
<td>activated</td>
<td>activated</td>
</tr>
<tr>
<td>Can SDO COB-IDs be</td>
<td>only for channel 2</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>freely selected?</td>
<td>... 10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
6 Planning the CANopen network
6.3 Device specifications of the field devices

6.3.1 Special features of the Servo Drives 9400

- The parameter data channel 1 is always active.
- The optional parameter data channels 2 ... 10 can be activated via the subcodes of the codes Cxx372 and Cxx373.

<table>
<thead>
<tr>
<th>SDO identifier</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>CANopen SDO server Rx identifier</td>
<td>C00372: CAN on board</td>
</tr>
<tr>
<td></td>
<td>C13372: Module in slot 1</td>
</tr>
<tr>
<td></td>
<td>C14372: Module in slot 2</td>
</tr>
<tr>
<td>CANopen SDO server Tx identifier</td>
<td>C00373: CAN on board</td>
</tr>
<tr>
<td></td>
<td>C13373: Module in slot 1</td>
</tr>
<tr>
<td></td>
<td>C14373: Module in slot 2</td>
</tr>
</tbody>
</table>

- If bit 31 is set (0x8nnnnnnn), the corresponding SDO server is deactivated.
- In order to change the COB-ID of a currently active parameter data channel, you have to first deactivate it and then activate it with a changed COB-ID. Both processes must be rendered effective by a "Reset Node" command via C00002.

Basic identifier - 9400 Servo Drives

The default setting of the basic identifier is as follows:

<table>
<thead>
<tr>
<th>Object</th>
<th>Direction</th>
<th>Basic identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>from the drive</td>
<td>to the drive</td>
</tr>
<tr>
<td>NMT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sync 1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Emergency</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PDO1 (Process data channel 1)</td>
<td>TPDO1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RPDO1</td>
<td></td>
</tr>
<tr>
<td>PDO2 (Process data channel 2)</td>
<td>TPDO2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RPDO2</td>
<td></td>
</tr>
<tr>
<td>PDO3 (Process data channel 3)</td>
<td>TPDO3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RPDO3</td>
<td></td>
</tr>
<tr>
<td>PDO4 (Process data channel 4)</td>
<td>TPDO4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RPDO4</td>
<td></td>
</tr>
<tr>
<td>SDO1 (Parameter data channel 1)</td>
<td>TSDO1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RSDO1</td>
<td></td>
</tr>
<tr>
<td>SDO2 ... 10 (parameter data channel 2 ... 10)</td>
<td>TSDOx</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RSDOx</td>
<td></td>
</tr>
<tr>
<td>Node guarding, heartbeat</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1) When creating the sync transmit/receive identifier manually, observe the use of the emergency telegram because of the same COB-ID.
### 6.3.2 Special features of the Inverter Drives 8400

**Basic identifier - 8400 Inverter Drives**

The default setting of the basic identifier is as follows:

<table>
<thead>
<tr>
<th>Object</th>
<th>Direction</th>
<th>Basic identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>from the drive</td>
<td>to the drive</td>
</tr>
<tr>
<td>NMT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sync 1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Emergency</td>
<td>●</td>
<td></td>
</tr>
<tr>
<td>PDO1 (Process data channel 1)</td>
<td>TPDO1</td>
<td>●</td>
</tr>
<tr>
<td></td>
<td>RPDO1</td>
<td>●</td>
</tr>
<tr>
<td>PDO2 (Process data channel 2)</td>
<td>TPDO2</td>
<td>●</td>
</tr>
<tr>
<td></td>
<td>RPDO2</td>
<td>●</td>
</tr>
<tr>
<td>PDO3 (Process data channel 3)</td>
<td>TPDO3</td>
<td>●</td>
</tr>
<tr>
<td></td>
<td>RPDO3</td>
<td>●</td>
</tr>
<tr>
<td>SDO1 (Parameter data channel 1)</td>
<td>TSDO1</td>
<td>●</td>
</tr>
<tr>
<td></td>
<td>RSDO1</td>
<td>●</td>
</tr>
<tr>
<td>SDO2 (Parameter data channel 2)</td>
<td>TSDO2</td>
<td>●</td>
</tr>
<tr>
<td></td>
<td>RSDO2</td>
<td>●</td>
</tr>
<tr>
<td>Heartbeat</td>
<td>●</td>
<td></td>
</tr>
<tr>
<td>Boot-up 2)</td>
<td>●</td>
<td></td>
</tr>
</tbody>
</table>

1) When creating the sync transmit/receive identifier manually, observe the use of the emergency telegram because of the same COB-ID.

2) When the boot-up identifier is set manually, observe the use of heartbeat because of the same COB-ID.
### 6.3.3 Special features of the I/O-System 1000 (EPM-Sxxx)

**Basic identifier - I/O system 1000 (EPM-Sxxx)**

The default setting of the basic identifier is as follows:

<table>
<thead>
<tr>
<th>Object</th>
<th>Direction</th>
<th>Basic identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>from the drive</td>
<td>to the drive</td>
</tr>
<tr>
<td>NMT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sync 1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Emergency</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PDO1 (Process data channel 1)</td>
<td>TPD01</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RPD01</td>
<td></td>
</tr>
<tr>
<td>PDO2 (Process data channel 2)</td>
<td>TPD02</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RPD02</td>
<td></td>
</tr>
<tr>
<td>PDO3 (Process data channel 3)</td>
<td>TPD03</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RPD03</td>
<td></td>
</tr>
<tr>
<td>PDO4 (Process data channel 4)</td>
<td>TPD04</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RPD04</td>
<td></td>
</tr>
<tr>
<td>PDO5 (Process data channel 5)</td>
<td>TPD05</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RPD05</td>
<td></td>
</tr>
<tr>
<td>PDO6 (Process data channel 6)</td>
<td>TPD06</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RPD06</td>
<td></td>
</tr>
<tr>
<td>PDO7 (Process data channel 7)</td>
<td>TPD07</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RPD07</td>
<td></td>
</tr>
<tr>
<td>PDO8 (Process data channel 8)</td>
<td>TPD08</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RPD08</td>
<td></td>
</tr>
<tr>
<td>PDO9 (Process data channel 9)</td>
<td>TPD09</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RPD09</td>
<td></td>
</tr>
<tr>
<td>PDO10 (Process data channel 10)</td>
<td>TPD10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RPD10</td>
<td></td>
</tr>
<tr>
<td>SDO1 (Parameter data channel 1)</td>
<td>TSD01</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RSD01</td>
<td></td>
</tr>
<tr>
<td>Node guarding</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1) When creating the sync transmit/receive identifier manually, observe the use of the emergency telegram because of the same COB-ID.
6 Planning the CANopen network

6.4 Special case: Delayed switch-on of one or more slaves

When the master is started, all slave devices must be switched on.

If this is not the case, a special procedure must be carried out for the following cases:

A. One or more slaves are switched on after the master starting process.
   In this case, indicate each of these devices as "optional device" on the CANopen Remote Device tab:

B. All slaves are switched on later.
   In this case, additionally deactivate the "Polling of optional slaves" option on the CANopen Manager tab:

After switching on the slaves, the Reset Communication command must be executed with the "NMT" function block which is provided in the CiA library.
7 Preparing the field devices

7.1 Installing field devices

Install the field devices according to the data given in the device-specific mounting instructions. Make sure that ...

- the CANopen installation complies with your overview screen.
- all devices are supported by the control technology system on the Logic bus and Motion bus.
- in the case of devices with several CAN interfaces, the correct interfaces are connected to the fieldbus.
- a terminating resistor is connected to the first and last node.
- the fieldbus is not unintentionally interrupted in switchable CAN connectors.

7.2 Setting node addresses and baud rate

- Set the specified node address and baud rate on the field devices via DIP switch (if available on the device), or via parameter/code.
- Mark the devices the settings of which you have changed in your overview screen.
- Attach address labels to the devices.

**Note!**

- Each node address must be unambiguous and may only be assigned once in the CANopen network.
- The baud rate must be set identically for all nodes.
- Observe the connection between bus cable length and baud rate.
  >>> Bus cable length (§ 24)

**Configuration via the »WebConfig«/»EASY Starter« for the Lenze Controller/IPC**

If the baud rate has been changed via the »WebConfig«, the Lenze Controller/IPC needs to be restarted afterwards. Then an online connection to the CAN nodes can be established with the »EASY Starter«.

Information on the DIP switch settings can be found in the documentation for the field devices.
7 Preparing the field devices
7.3 Connecting the Engineering PC to the Lenze Controller

To commission the field devices, an online connection is required between the Engineering PC and the field device. To establish an online connection between an Engineering PC and a field device (like an inverter), two ways are possible:

<table>
<thead>
<tr>
<th>Direct coupling</th>
<th>Lenze Controller as gateway</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Engineering PC</strong></td>
<td><strong>Engineering PC</strong></td>
</tr>
<tr>
<td>Ethernet</td>
<td>Ethernet</td>
</tr>
<tr>
<td>CANopen (Logic/Motion)</td>
<td>CANopen (Logic/Motion)</td>
</tr>
<tr>
<td>Lenze Controller</td>
<td>Lenze Controller</td>
</tr>
</tbody>
</table>

[7-1] Online connection between the Engineering PC and the field device

If the Lenze Controller has not been commissioned yet, directly connect the Engineering PC to the CANopen bus to commission the field devices. To activate the Engineering PC, for example use the USB system bus adapter (EMF2177IB). Then the download times are optimal and it is not necessary to commission the controller first.

As soon as the Lenze Controller has been commissioned, no direct coupling should be used anymore since it may disturb the real-time capability of the fieldbus. This especially applies to the CANopen Motion bus. Here, the transmission of the sync telegram on time can be disturbed so that an increased jitter on the fieldbus may be the result.

Moreover, each field device requires a second parameter data channel for independent bus access by two masters (controllers). For some device types, the parameter data channel must be installed separately, e.g. in the case of the Servo Drives 9400.

As an option, some controllers can operate two independent CAN interfaces. In this case, one interface can be used for the connection with the Lenze Controller, the other for the direct connection of the Engineering PC. Thus, two fieldbuses are created which are physically independent. In this case, the real-time capability of the nodes on the Motion bus is not influenced even with direct coupling. However, the wiring expense increases.
The communication speed with the field devices, when being commissioned, mainly depends on whether the Lenze Controller is currently running or is stopped. In the latter case, the total bandwidth of the fieldbus is provided to the gateway so that the speed advantage in the case of direct coupling would only be marginal. Thus, the use of the Lenze Controller as gateway as part of the control technology should be clearly preferred.

Information on the commissioning of Lenze field devices is provided in the chapter "Commission the field devices" (41).
Commissioning the CANopen Logic bus

This chapter provides information for the commissioning of the CANopen Logic field devices in the Lenze automation system.

Depending on the field devices used, the following Lenze Engineering tools are required:

- »EASY Starter«
- »Engineer«
- »PLC Designer«

8.1 Sample projects (Application Samples)

There already exist sample projects (device application + PLC program) for commissioning of Lenze Controllers.

The Lenze sample projects can be found in the MS Windows start menu under:

Start → All programs → Lenze → AppSamples → ...

The Lenze sample projects can also be opened in the »PLC Designer« via the menu command File → New project..., or using <Ctrl>+<N>.

Detailed information on the sample projects can be found in the following documentation:

- SW_ApplicationSample_i700_(PLC Designer V3)_Vx-y_DE/EN.pdf
- SW_ApplicationSamples_(Controller-based)_Vx-y_DE/EN.pdf
8 Commissioning the CANopen Logic bus

8.2 Overview of the commissioning steps

In the following sections, the individual commissioning steps are described. Follow the instructions of these sections step by step in order to commission your system.

<table>
<thead>
<tr>
<th>Step</th>
<th>Activity</th>
<th>Lenze Engineering tool to be used</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Create a project folder (§ 40)</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>Commission the field devices (§ 41)</td>
<td>»Engineer« / »EASY Starter«</td>
</tr>
<tr>
<td>3.</td>
<td>Creating a PLC program with a target system (Logic) (§ 42)</td>
<td>»PLC Designer«</td>
</tr>
<tr>
<td>4.</td>
<td>Configuring the communication parameters (§ 44)</td>
<td></td>
</tr>
<tr>
<td>5th</td>
<td>Importing missing devices / device description files (§ 46)</td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>Creating a control configuration (adding field devices) (§ 47)</td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>Setting of CAN parameters and PDO mapping (§ 52)</td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>Creating the program code for controlling the Logic field device (§ 59)</td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>Preparing the restart (§ 61)</td>
<td></td>
</tr>
<tr>
<td>10.</td>
<td>Compiling the PLC program code (§ 66)</td>
<td></td>
</tr>
<tr>
<td>11.</td>
<td>Logging in on the Lenze Controller with the »PLC Designer« (§ 66)</td>
<td>With the log-in, the fieldbus configuration and the PLC program are loaded to the controller.</td>
</tr>
<tr>
<td>12.</td>
<td>Starting the PLC program (§ 66)</td>
<td></td>
</tr>
</tbody>
</table>

More detailed information about how to work with the Lenze Engineering tools can be found in the corresponding manuals and online helps.
8 Commissioning the CANopen Logic bus

8.3 Create a project folder

Create a project folder on the Engineering PC.

Use this project folder to store the data generated in the following different project configuration steps:

- Project data created in the »Engineer« or »EASY Starter«
- The project file created in the »PLC Designer«

💡 Tip!
Create a separate project folder for every CAN configuration for storing the project files.
8 Commissioning the CANopen Logic bus

8.4 Commission the field devices

Parameterise the Lenze field devices connected to the CANopen network by means of the »Engineer« or »EASY Starter«.

CANopen is exclusively configured by means of the »PLC Designer«.

CANopen settings of the field devices which have possibly been carried out with the »Engineer«/»EASY Starter« are overwritten.

**Tip!**
We recommend to commission each field device individually and then integrate them into the PLC program.

There already exist sample projects (device application + PLC program) for commissioning of Lenze Controllers.

- **Sample projects (Application Samples)** (38)

---

**Documentation of the Lenze field devices**

Here you are provided with some detailed information relating to the commissioning of the Lenze field devices.
8 Commissioning the CANopen Logic bus

8.5 Creating a PLC program with a target system (Logic)

The »PLC Designer« serves to model the network topology in the control configuration.

Tip!
The »PLC Designer« can be used to configure CANopen nodes and nodes on other fieldbus systems.

Mixed operation of CANopen and EtherCAT (§ 94)

How to create a PLC program in the »PLC Designer«:

1. Use the menu command File → New project to create a new »PLC Designer« project.

A "Standard object" simplifies the structure of a project in the »PLC Designer«; for instance, a device tree structure with a target system, PLC logic, etc. is provided.

How to create a PLC program in the «PLC Designer»:

1. Use the menu command File → New project to create a new »PLC Designer« project.

A "Standard object" simplifies the structure of a project in the »PLC Designer«; for instance, a device tree structure with a target system, PLC logic, etc. is provided.

• Go to the 2 Name input field and enter a name for your »PLC Designer« project.
• Select the previously created project folder as storage location in the 3 Location selection field.

Create a project folder (§ 40)

3. Confirm the entries by clicking OK.
4. Go to the Standard project dialog window and select the target system in the **Controller type** selection field:

Further optional project settings

2. Selection of the Controller firmware version

3. Selection of the compiler version

4. Selection of the programming language:
   - Sequential function chart (AS)
   - Instruction list (AWL)
   - Continuous Function Chart (CFC)
   - Function block diagram (FUP)
   - Ladder diagram (KOP)
   - Structured text (ST)

5. Confirm the selection by clicking **OK**.
8.6 Configuring the communication parameters

Set the communication parameters to establish an online connection to the Lenze Controller later on.

How to configure the communication parameters

1. Go to the **Communication settings** tab of the target system (device) and click the **Add gateway** button.

Then go to the Gateway dialog box and enter the **IP address** of the controller. (By double-clicking the predefined value it can be overwritten.)

2. Confirm the entry by clicking **OK**.
3. Click the **Scan network** button.

![Scan network button](image)

4. Select the suitable controller for the IP address entered under 2. and activate it by means of the **Set active path** button (or by double-click).

![Set active path button](image)

5. Now you can execute the following action using the »PLC Designer«:
   - **Logging in on the Lenze Controller with the »PLC Designer«** (p. 66)
8 Commissioning the CANopen Logic bus

8.7 Importing missing devices / device description files

The device description file contains the data of the fieldbus peripherals required for the master control. This file is required to program the control system.

With the »PLC Designer«, device descriptions for the following Lenze device series are installed as well:

- i700 servo inverter
- Servo Drives 9400
- Inverter Drives 8400
- I/O system 1000 (EPM-Sxxx)
- Fieldbus communication cards for controller 3200 C / p500 (EtherCAT, CANopen, PROFIBUS)

In order to furthermore integrate missing devices or devices of other manufacturers, the corresponding device description files of the manufacturer are required.

In the »PLC Designer« you can import device description files of the *.XML, *.devdesc.XML, *.EDS, *.DCF, and *.GSx type via the menu command Tools → Device Repository....

Tip!

Current device description files for Lenze devices can be found in the "Download" area at:

www.lenze.com
8 Commissioning the CANopen Logic bus

8.8 Creating a control configuration (adding field devices)

How to create the control configuration in the «PLC Designer»:

1. Go to the context menu of the target system and use the command **Add Device** to add the "CANbus" to the control configuration.
2. Use the CANbus tab to set the baud rate.

![CANbus tab](image)

2. Use the **CANbus** tab to set the baud rate.

3. Use the **Add Device** to add the **"CANopen Manager MC-CAN2"** to the control configuration.

![Add Device](image)

3. Use the **Add Device** to add the **"CANopen Manager MC-CAN2"** to the control configuration.

**Note!**

The baud rate set in the »PLC Designer« overwrites the baud rate set for the field devices via »Engineer« or »EASY Starter«.

In a CANopen network, set the same baud rate for all nodes.
4. Use the **CANopen_Manager** tab to set the parameters for Sync generation.

The sync producing is required if ...
- at least one PDO with sync-controlled processing is used on the bus;
- the applications are to run in synchronism in several field devices;
- Motion devices are to be operated on the fieldbus.

If you want to use CAN synchronisation, tick the input field **1 Enable Sync Producing**.

Go to the input field and set the **2 Cycle Period**.
5. Use the command **Add Device** to add a Logic device (slave) to the CANopen_Manager (master).

Select a field device from the selection list. You can only select devices the CAN device description files of which have been imported into the »PLC Designer«.

- **Importing missing devices / device description files** (§ 46)

For EDS files created in the »Engineer«, the field device appears in the selection list with the same name as during the export of the EDS file in the »Engineer«, extended by the name of the interface and device type.

6. Repeat the **Add Device** command until all slaves connected to the fieldbus are included in the control configuration.
7. Give the inserted slaves suitable names (e.g. "Drive_vertical").

The names must …
  • only contain the characters "A ... Z", "a ... z", "0 ... 9" or "_";
  • must not begin with a digit.

You can enter a name by clicking the element.

Example:

- CANbus (CANbus)
- CANopen_Manager_MC_CAN (CANopen Manager MC-CAN)
- Drive_vertical (5400 Highline CA 402)
- SM_Drive_CAN_9400HL (SM_Drive_CAN_9400HL)
8.9 Setting of CAN parameters and PDO mapping

Set the CAN parameters and the PDO mapping for each Logic device connected to the bus.

How to set CAN parameters and CAN mapping:
1. Go to the CANopen Remote Device tab of the respective slave.

Use the input field **1 Node ID** to set the node address corresponding to the settings in the field devices:

   - **2 Enable Expert Settings** option is ticked.
   - Make the settings required for your application here.

The following possible settings are only displayed if the **2 Enable Expert Settings** option is ticked.
2. Go to the **PDO Mapping** tab.

By default, the PDO mapping is set to a position mapping. This mapping can be changed manually (by setting a checkmark). Due to the limited bandwidth of the CAN bus, this is only useful in special cases. The default PDO properties, too, are useful default settings and should not be changed.

By double-clicking a single PDO, you can see its transmission properties.

- The **Transmission Type** "cyclic - synchronous (type 1-240)" and the setting, at which sync the PDOs are to be sent, must not be changed.
- The settings for the **Inhibit Time** and the **Event Time** are not evaluated.
- Confirm the settings with the **OK** button.

On the **CANopen I/O Mapping** tab, PLC variables can be assigned to the process image.

---

**Note!**

The Lenze Controller does not support any monitoring times for asynchronous receive-PDOs. This is only possible with slave field devices.
3. The selection of a special bus cycle task on the **CANopen I/O Mapping** tab of the CANopen manager is not essential.

The standard setting automatically uses the task with the shortest cycle time that accesses the devices of the CAN master as bus cycle task:

The "Cycle settings of the higher-level bus" serve to use the bus cycle task set via the **PLC settings** tab of the Lenze Controller (device):
8.9.1 Cross communication between the slaves

With control via the CAN bus, cross communication between the slaves is possible. For this purpose, you must configure the CAN communication and the PDO mapping in the »Engineer« and write it to the inverters.

Since the mapping for the cross communication between the slaves is not available in the control configuration, you must set the "No initialisation" option in the »PLC Designer« on the CANopen Remote Device tab of the respective slaves.

Thus the CAN settings and mapping settings in the slave drives will not be overwritten by the Lenze Controller when the PLC program starts.

**Note!**

In addition to the cross communication between the slaves, communication with the master must also be configured in the »Engineer«.

If the "No initialisation" option is active, this part of the PDO mapping is also not overwritten by the Lenze Controller.
When using the EPM-S600 ... S604 counter modules at the EPM-S110 head station, the following sequence must be observed when creating the configuration in the »PLC Designer«.

**Note!**

Before adding I/O modules below an **EPM-S110** head station in the »PLC Designer«, the head station tab must be closed.

If the I/O 1000 modules are added when the head station tab is open, there will not be shown any I/O image for the modules.

When all modules have been added, the EPM-S110 head station tab can be opened again.

**How to configure the EPM-S600 ... S604 counter modules:**

1. Add EPM-S110 to the »PLC Designer« device configuration.
2. Add I/O discs and counter discs.
3. For EPM-S110,
   - 1. activate **Enable Expert Settings**;
   - 2. deactivate **Autoconfig. PDO Mapping**.
4. Execute the menu command **Window → Close All Editors**.
5. Open EPM-S110 again with a double-click.
6. Add the PDO configuration for the counter discs.

The following example uses a send PDO and a mapping element to describe the basic configuration process.

Proceed accordingly for configuring the send PDOs ("Send PDO Mapping" tab).

A) Use the **Add PDO** button and add the receive PDO.

- Depending on the counter discs used, one or more PDOS must be added in send and receive direction.
- Please refer to the module documentation to see which additional elements of the process image of the counter modules are required for operation.
- The properties of the PDO can be defined via the "PDO properties" dialog box.
B) Manually create the mapping of the new PDOs.

- Select PDO and click **Add Mapping**.
- Select the corresponding settings in the dialog box that appears.

**Note!**

If **Autoconfig. PDO Mapping** (see step 3) is reactivated after adding the manual configuration, the entire configuration that has been added manually will be deleted.
8.10 Creating the program code for controlling the Logic field device

If the device descriptions for logic devices that are supplied with the »PLC Designer« are used, the process data are copied to the subordinate logic drive node automatically.

If the process data are to be continued to be linked manually, go to the LenzeLogicDrive Configuration tab and activate this option:

Then continue to create the program code.
How to create the program code:

1. Create the program code for controlling the field device.

   The device must be used in the program code in order that the SDO initialisation takes place.

   If other field devices are added to the control configuration, this may change the object addresses (%Qxx, %Ixx) of the existing variables. For this reason, do not use addresses directly in the program code to access the input and output objects or to assign values to them. Use the CANopen I/O Mapping tab for this purpose and assign own unique variable names according to the IEC 61131 syntax (no blanks and leading digits in the variable name):

   ![CANopen I/O Mapping](image)

   Already existing variables (e.g. global variables from the function libraries) can be integrated via the button .

   The manual assignment of object addresses in the Address column is not supported. Hence, only use the automatically assigned addresses of the process image. A manual assignment causes malfunctions.

2. Completely compile the »PLC Designer« project and transfer it to the Lenze Controller.

   Menu command: Build→Build

Special features of the I/O-System 1000 (EPM-Sxxx)

- When the program starts, the Lenze Controller initialises the I/O system. It changes to the "Operational" status.

- When the Lenze Controller initialises the I/O system, the »Engineer« must not be online on the same SDO channel.
8.11 Preparing the restart

In the control technology system you can use the Lenze Controller to transmit the complete parameter setting via SDO initialisation to the field devices when the machine is switched on. According to DS301, the Lenze Controller always initialises the CAN parameters of the field devices. Additionally the controller can initialise further parameters. The values for this must be stored in the Control configuration under the Service Data Objects tab.

Usually, the Lenze Controller only transmits the SDO objects for which you have stored another value than the standard value. The controller does not compare these values with the existing values in the field device. Thus, not all parameters changed there may be set correctly.

If you want a factory adjustment to be carried out in the field device before the SDO initialisation is carried out, tick under the CANopen Remote Device tab ...

1. "Enable Expert Settings",
2. "Default Settings":

![Canopen Remote Device Tab](image)

**Note!**

During a factory adjustment, the parameter setting in the field device, which you have carried out with the »Engineer«, gets lost. In this case, you have to transmit all parameter values manually to the Service Data Objects tab. This only makes sense when commissioning is completed and all parameters are optimised. If you change something afterwards via the »Engineer«, you have to maintain it in the PLC program.

The Service Data Objects tab contains the codes which are written in the EDS file. The EDS file contains all writable codes.
8 Commissioning the CANopen Logic bus

8.11 Preparing the restart

8.11.1 Special features of the 9400 Servo Drives HighLine

Servo Drives 9400 are not purely parameterisable devices. They require an application download, where several files are transmitted to the memory module.

To put a Servo Drive 9400 into operation, you can:

- plug on the memory module.
- transmit the application using the »Engineer«. For this, you must keep the original »Engineer« project.
- transmit the application using the »Loader«. For this, you must export and keep the required files from the »Engineer« project.

8.11.2 Special features of the Inverter Drives 8400

Inverter Drives 8400 are purely parameterisable devices.

To put an Inverter Drives 8400 into operation, you can transmit the application using the »Engineer«. For this, you must keep the original »Engineer« project.
8.11.3 Special features of the I/O-System 1000 (EPM-Sxxx)

There are different possibilities for restarting the I/O system:
- Automatic
- Automatically with factory adjustment
- Using the »Engineer«

For the corresponding settings, first tick the "Expert settings" on the CANopen Remote Device tab.

8.11.3.1 Automatic restart

Settings for automatic initialisation by the Lenze Controller:

1. Enter all desired parameter values under the All parameters tab.

2. Tick "Create all SDOs" under the CANopen Remote Device tab.

When the I/O system has been replaced

1. Set the node address and the baud rate at the code switch.
2. Start the Lenze Controller.
8 Commissioning the CANopen Logic bus

8.11 Preparing the restart

8.11.3.2 Automatic restart with factory adjustment

Settings for automatic initialisation with factory adjustment by the Lenze Controller:

1. Enter all desired parameter values under the All parameters tab.

2. Tick "Default settings" under the CANopen Remote Device tab.

When the I/O system has been replaced

1. Set the node address and the baud rate at the code switch.

2. Start the Lenze Controller.
8 Commissioning the CANopen Logic bus
8.11 Preparing the restart

8.11.3.3 Restart with the »Engineer«

**Conditions**
- You have implemented the I/O system successfully into the PLC program.
- You have parameterised some CANopen indexes of the I/O system using the »Engineer«.

**No transmission of the parameter setting to the Control configuration**
You do not want to transmit the parameter setting to the Control configuration now, since you can assume that after a possible device replacement the configuring software will be available.
- Operate the I/O system in CANopen mode.
- Do not tick the "Default Settings" under the CANopen Remote Device tab in order that the Lenze Controller does not execute a factory adjustment.

1. Set the node address and the baud rate at the code switch.
2. Transmit the archived parameter setting to the I/O system.
3. Start the Lenze Controller.

- Save the »Engineer« project near the machine.

When the I/O system has been replaced
1. Set the node address and the baud rate at the code switch.
2. Transmit the archived parameter setting to the I/O system.
3. Start the Lenze Controller.
8 Commissioning the CANopen Logic bus

8.12 Compiling the PLC program code

In order to compile the PLC program code, select the menu command Build → Build, or press function key <F11>.

- If errors occur during translation, they can be located and corrected on the basis of the »PLC Designer« error messages. Then re-translate the program code.
- If no errors have occurred during the compilation process, save the »PLC Designer« project in the project folder.

8.13 Logging in on the Lenze Controller with the »PLC Designer«

Use the menu command Online → Login or <Alt>+<F8> to log in on the Lenze Controller.

- For this, the PLC program must be error-free.
- With the log-in, the fieldbus configuration and the PLC program are loaded to the controller. Any configuration or a PLC program that is possibly available is overwritten.

8.14 Starting the PLC program

Before the start, the PLC program must be loaded to the Lenze Controller using the menu command Online → Login.

Use the menu command Debug → Start or function key <F5> to start the PLC program.

8.15 Start parameters of the Servo Drives 9400 HighLine CiA 402

When the Lenze Controller is started, the following "start parameters" are automatically loaded to the Servo Drives 9400 HighLine CiA 402:

<table>
<thead>
<tr>
<th>Index: Subindex [hex]</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x5EEB:0x00</td>
<td>Signal source torque setpoint (C00275)</td>
</tr>
<tr>
<td>0x5B9E:0x00</td>
<td>Sync cycle time (C01121)</td>
</tr>
<tr>
<td>0x5B95:0x00</td>
<td>CAN SYNC application cycle (C01130)</td>
</tr>
<tr>
<td>0x60C0:0x01</td>
<td>Interpolation time unit</td>
</tr>
<tr>
<td>0x60C0:0x02</td>
<td>Interpolation time value</td>
</tr>
<tr>
<td>0x60C2:0x00</td>
<td>Selection of the interpolation submode</td>
</tr>
</tbody>
</table>
9 Commissioning the CANopen Motion bus

This chapter provides information on commissioning the CANopen Motion field devices in the Lenze automation system.

The commissioning of a Motion device does not differ fundamentally from the commissioning of a Logic device. Below a Motion device, an additional "SoftMotion" node is displayed in the device tree. Via this node, further settings must be made.

Depending on the field devices used, the following Lenze Engineering tools are required:

- »EASY Starter«
- »Engineer«
- »PLC Designer«
9 Commissioning the CANopen Motion bus

9.1 Sample projects (Application Samples)

There already exist sample projects (device application + PLC program) for commissioning of Lenze Controllers.

The Lenze sample projects can be found in the MS Windows start menu under:

Start → All programs → Lenze → AppSamples → ...

The Lenze sample projects can also be opened in the »PLC Designer« via the menu command File → New project... , or using <Ctrl>+<N>.

Detailed information on the sample projects can be found in the following documentation:

• SW_ApplicationSample_i700_(PLC Designer V3)_Vx-y_DE/EN.pdf
• SW_ApplicationSamples_(Controller-based)_Vx-y_DE/EN.pdf
### Overview of the commissioning steps

<table>
<thead>
<tr>
<th>Step</th>
<th>Activity</th>
<th>Lenze Engineering tools to be used</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Create a project folder</td>
<td></td>
</tr>
</tbody>
</table>
| 2.   | Commission the field devices | »Engineer« / »EASY Starter«  
| 3.   | Creating a PLC program with target system (Motion) | »PLC Designer«  
| 4.   | Configuring the communication parameters |  
| 5th  | Creating a Motion task |  
| 6.   | Creating a control configuration |  
| 6.   | Setting SoftMotion parameters |  
| 7.   | Setting of CAN parameters and PDO mapping |  
| 8.   | Creating the program code for controlling the Motion field device |  
| 10.  | Preparing the restart |  
| 11.  | Compiling the PLC program code |  
| 12.  | Logging in on the Lenze Controller with the »PLC Designer« | With the log-in, the fieldbus configuration and the PLC program are loaded to the controller.  
| 13.  | Starting the PLC program |  

In the following sections, the individual commissioning steps are described. Follow the instructions of these sections step by step in order to commission your system.

> More detailed information about how to work with the Lenze Engineering tools can be found in the corresponding manuals and online helps.
9 Commissioning the CANopen Motion bus

9.3 Create a project folder

Create a project folder on the Engineering PC.

Use this project folder to store the data generated in the following different project configuration steps:

• Project data created in the »Engineer« or »EASY Starter«
• The project file created in the »PLC Designer«

Tip!

Create a separate project folder for every CAN configuration for storing the project files.
9 Commissioning the CANopen Motion bus

9.4 Commission the field devices

Parameterise the Lenze field devices connected to the CANopen network by means of the »Engineer« or »EASY Starter«.

Die CANopen is exclusively configured using the »PLC Designer«.

CANopen settings of the field devices which have possibly been carried out with the »Engineer«/ »EASY Starter« are overwritten.

Tip!

- We recommend to commission each field device individually and then integrate them into the PLC program.
- There already exist sample projects (device application + PLC program) for commissioning of Lenze Controllers.
  - Sample projects (Application Samples) (p. 68)

For the Servo Drive 9400 Highline CiA 402 the following parameters must be set manually via the »Engineer«:

- Homing mode (C02640, set in machine-dependent manner)
- Touch-probe interface (set in machine-dependent manner)
- Control of the holding brake (0x60FB/2 | Brake control)
  Depending on the setting of this parameter, the holding brake is applied for a short time after the conclusion of the home position path. In order to avoid this, set bit 2 in this parameter ("disable stop": does not apply the brake at standstill).

These parameters are not set via the Lenze Controller.
9 Commissioning the CANopen Motion bus

9.5 Creating a PLC program with target system (Motion)

The »PLC Designer« serves to model the network topology in the control configuration.

Tip!
The »PLC Designer« can be used to configure CANopen nodes and nodes on other fieldbus systems.

» Mixed operation of CANopen and EtherCAT (94)

How to create a PLC program in the »PLC Designer«:

1. Use the menu command File → New project to create a new »PLC Designer« project.

2. Select 1 "Standard project" in the New Project dialog window.

   A "Standard object" simplifies the structure of a project in the »PLC Designer«; for instance, a device tree structure with a target system, PLC logic, etc. is provided.

   • Go to the 2 Name input field and enter a name for your »PLC Designer« project.
   • Select the previously created project folder as storage location in the 3 Location selection field.

   » Create a project folder (70)

3. Confirm the entries by clicking OK.
4. Go to the Standard project dialog window and select the target system in the Controller type selection field:

Further optional project settings

2 Selection of the Controller firmware version

3 Selection of the compiler version

4 Selection of the programming language:
   • Sequential function chart (AS)
   • Instruction list (AWL)
   • Continuous Function Chart (CFC)
   • Function block diagram (FUP)
   • Ladder diagram (KOP)
   • Structured text (ST)

5. Confirm the selection by clicking OK.
9 Commissioning the CANopen Motion bus

9.6 Configuring the communication parameters

Set the communication parameters to establish an online connection to the Lenze Controller later on.

How to configure the communication parameters

1. Go to the Communication settings tab of the target system (device) and click the **Add gateway** button.

Then go to the Gateway dialog box and enter the **IP address** of the controller. (By double-clicking the predefined value it can be overwritten.)

2. Confirm the entry by clicking **OK**.
3. Click the Scan network button.

4. Select the suitable controller for the IP address entered under 2. and activate it by means of the Set active path button (or by double-click).

5. Now you can execute the following action using the »PLC Designer«:
   - Logging in on the Lenze Controller with the »PLC Designer« (p. 93)
9 Commissioning the CANopen Motion bus

9.7 Creating a Motion task

How to create a Motion task:

1. Go Task configuration in the configuration tree.
2. Create a new task with the Add Object command.
   Assign a sensible task name (e.g. "MotionTask").

3. Enter a reasonable cycle time in milliseconds in the Interval input field.

The cycle time to be entered depends on the number of Motion axes and the runtime of the PLC application. In case of a small PLC application, the minimum cycle time \(T_{\text{cycl}}\) is determined by the number of Motion axes due to the transfer rate of the CANopen bus.

\[ T_{\text{cycl}} \text{[ms]} = \frac{\text{number of Motion axes}}{3} \]
4. Go to **Application** in the configuration tree.

5. Use the **Add Object** command to create a new program block (POU) in the application. Assign a reasonable POU name (e.g. "Motion_PRG").
6. Add this program call to the task using the Add POU command.
The following task configuration is caused:
9 Commissioning the CANopen Motion bus

9.8 Creating a control configuration

---

9.8 Creating a control configuration

**Note!**

The configuration of a Lenze Controller in a CANopen network must be created in the »PLC Designer«, because the complete configuration is written to the connected slaves when a controller is started. This process overwrites the previous slave settings.

How to create the control configuration in the »PLC Designer«:

1. Go to the context menu of the target system and use the command 1 **Add Device** to add the 2 **"CANbus"** to the control configuration.
2. Use the **CANbus** tab to set the baud rate.

![CANbus tab](image)

**Note!**

The baud rate set in the »PLC Designer« overwrites the baud rate set for the field devices via »WebConfig« or »Engineer«.

In a CANopen network, set the same baud rate for all nodes.

3. Use the command **1 Add Device** to add the **2 “CANopen Manager”** to the control configuration.

![Add Device](image)
4. Use the **CANopen Manager** tab to set the parameters for Sync generation.

![CANopen Manager](image)

**General**
- Node ID: 127
- Autostart CANopenManager
- Start Serves
- NMT Start All (if possible)

**Sync**
- Enable Sync Producing
- PDO ID (Hex): 166
- Cycle Period (μs): 1000
- Window Length (μs): 1200
- Enable Sync Consuming

---

**Note!**

To prevent the sync telegram from jittering, all Motion phases must be assigned to the same task.

The sync producing is required if ...
- at least one PDO with sync-controlled processing is used on the bus;
- the applications are to run in synchronism in several field devices;
- motion devices are to be connected to the bus.

If you want to use CAN synchronisation, tick the input field **1 Enable Sync Producing**.

Go to the input field **2** and set the **Cycle Period**.
5. Use the command **Add Device** to add a Motion device (slave) to the CANopen_Manager (master).

![Add Device](image)

Select a field device from the **2** selection list.

For EDS files created in the »Engineer«, the field device appears in the selection list with the same name as during the export of the EDS file in the »Engineer«, extended by the name of the interface and device type.

6. Repeat the command **Add Device** until all slaves connected to the bus are integrated into the control configuration.

7. Give the inserted slaves suitable names (e.g. "Drive_vertical").

The names must ...  
- only contain the characters "A ... Z", "a ... z", "0 ... 9" or ";
- must not begin with a digit.

You can enter a name by clicking the element.

**Example:**

```
- CANbus (CANbus)
  - CANopen_Manager_MC_CAN (CANopenManager MC-CAN)
    - Drive_vertical (9400 Highline CA 402)
```


9.9 Parallel operation of two synchronised CAN buses

The MC-CAN2 communication card is provided with two CAN interfaces. Thus, basically two CAN buses can be operated independently of each other. The two buses can also be operated in a sync-controlled manner.

In the basic setting, which is sufficient for most application cases, there is an interval of the sync telegrams of approx. 50 μs on the two buses.

If special cases require a different setting, the following optimisation can be carried out via the »WebConfig« (see software manual for the Lenze Controller):

- Set the parameter "Sync master interface index" = 1 for "CAN interface 2".
- This reduces the interval of the sync telegrams on the two buses to approx. 20 μs.
In the device tree of the »PLC Designer« project, the CAN bus which works via CAN interface 1 (CAN1) must be first:
For the **Servo Drive 9400 Highline CiA 402** the following parameters must be set manually via the »Engineer«:

- Homing mode (C02640, set in application-dependent manner)
- Touch-probe interface (set in application-dependent manner)
- Control of the holding brake (0x60FB/2 | Brake control)

Depending on the setting of this parameter, the holding brake is applied for a short time after the conclusion of the home position path. In order to avoid this, set bit 2 in this parameter ("disable stop": does not apply the brake at standstill).

These parameters are not set via the Lenze Controller.

**Example of a minimum configuration with a Motion device (Servo Drive 9400 HighLine CiA 402)**
How to set the SoftMotion parameters

1. Open the tab **SoftMotion drive: Scaling/Mapping** and adapt the conversion factors in the "Scaling" area.
2. Open the tab **SoftMotion drive: basic parameters** and set the axis types and limitations.
   - Do not use the "virtual mode" setting.
   - Virtual axes are located in the "SoftMotion General Drive Pool".

**Configuration of a Motion device**

**Rotary axis**
(type: Modulo, 360°/revolution, ratio 1:1):

For configuring a **linear axis** Motion device (type: Finite), you can activate and determine the software limit switches:

3. Repeat steps 1 and 2 for all Motion devices connected to the field bus.

**Online help of »PLC Designer«**

Here, you can find detailed descriptions of the **SoftMotion tabs**.
9 Commissioning the CANopen Motion bus

9.11 Setting of CAN parameters and PDO mapping

Set the CAN parameter and the PDO mapping for each Motion device connected to the bus.

How to set CAN parameters and CAN mapping:

1. Go to the CANopen Remote Device tab of the respective slave.

Use the input field **Node ID** to set the node address corresponding to the settings in the field devices:

   The following possible settings are only displayed if the **Enable Expert Settings** option is ticked.

   Make the settings required for your application here.
9 Commissioning the CANopen Motion bus
9.11 Setting of CAN parameters and PDO mapping

2. Go to the PDO Mapping tab.

By default, the PDO mapping is set to a position mapping. This mapping can be changed manually (by setting a checkmark). Due to the limited bandwidth of the CAN bus, this is only useful in special cases. The default PDO properties, too, are useful default settings and should not be changed.

By double-clicking a single PDO, you can see its transmission properties.

- The **Transmission Type** "cyclic - synchronous (type 1-240)" and **3** the setting, at which sync the PDOs are to be sent, must not be changed.
- The settings for the **1** Inhibit Time and the **4** Event Time are not evaluated.
- Confirm the settings with the **OK** button.

On the CANopen I/O Mapping tab, PLC variables can be assigned to the process image.

**Note!**

The Lenze Controller does not support any monitoring times for asynchronous receive-PDOs. This is only possible with slave field devices.
3. The selection of a special bus cycle task on the CANopen I/O Mapping tab of the CANopen manager is not essential.

The standard setting automatically uses the task with the shortest cycle time that accesses the devices of the CAN master as bus cycle task:

The "Cycle settings of the higher-level bus" serve to use the bus cycle task set via the PLC settings tab of the Lenze Controller (device):
9.12 Creating the program code for controlling the Motion field device

This depends on the automation task, the use of PLCopen blocks or the CNC programming.

**Note!**

All SoftMotion function blocks, SoftMotion functions, and the read/write block parameters that access the SoftMotion devices (e.g. MC_WriteParameter or MC_WriteBoolParameter) must only be called from the Motion task.

If they are called from another task, their execution may be incorrect.

Also see the chapter "Creating the program code for controlling the Logic field device" (§ 59).

9.13 Preparing the restart

Saving the parameter set of the drive via the »Engineer«.

See also Preparing the restart (§ 61).
9 Commissioning the CANopen Motion bus

9.14 Compiling the PLC program code

In order to compile the PLC program code, select the menu command Build → Build, or press the function key <F11>.

- If errors occur during translation, they can be located and corrected on the basis of the »PLC Designer« error messages. Then re-translate the program code.
- If no errors have occurred during the compilation process, save the »PLC Designer« project in the project folder.

9.15 Logging in on the Lenze Controller with the »PLC Designer«

Use the menu command Online → Login or <Alt>+<F8> to log in on the Lenze Controller.

- For this, the PLC program must be error-free.
- With the log-in, the fieldbus configuration and the PLC program are loaded to the controller. Any configuration or a PLC program that is possibly available is overwritten.

9.16 Starting the PLC program

Before the start, the PLC program must be loaded to the Lenze Controller using the menu command Online → Login.

Use the menu command Debug → Start or the function key <F5> to start the PLC program.

9.17 Start parameters of the Servo Drives 9400 HighLine CiA 402

When the Lenze Controller is started, the following "start parameters" are automatically loaded to the Servo Drives 9400 HighLine CiA 402:

<table>
<thead>
<tr>
<th>Index: Subindex [hex]</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0xSEEB:0x00</td>
<td>Signal source torque setpoint (C00275)</td>
</tr>
<tr>
<td>0x5B9E:0x00</td>
<td>Sync cycle time (C01121)</td>
</tr>
<tr>
<td>0x5B95:0x00</td>
<td>CAN SYNC application cycle (C01130)</td>
</tr>
<tr>
<td>0x60C0:0x01</td>
<td>Interpolation time unit</td>
</tr>
<tr>
<td>0x60C0:0x02</td>
<td>Interpolation time value</td>
</tr>
<tr>
<td>0x60C2:0x00</td>
<td>Selection of the interpolation submode</td>
</tr>
</tbody>
</table>
10 Mixed operation of CANopen and EtherCAT

Within the Lenze Controller-based Automation, CANopen can be used in parallel to the EtherCAT bus system. This is useful if not all field devices are available for the same bus system or if, in parallel to the CANopen bus as Logic bus, a Motion bus (EtherCAT) is required.

Note!

- Due to the demands on the real-time behaviour of the fieldbus system and the limited transfer capacity, for CANopen it is useful to operate Logic and Motion devices on separate fieldbus lines – on a Logic bus and a Motion bus.
- In mixed operation, ensure that the CAN Motion task has the highest priority. The task assigned to the EtherCAT bus should have the second-highest priority. The tasks assigned to the Logic bus systems should be configured with a lower priority.

Controller-based Automation EtherCAT communication manual

Here you can find detailed information on how to commission EtherCAT components.
The **SM3_Drive_Lenze.lib** function library supports the control of Motion devices (e.g. Servo Drives 9400 HighLine CiA 402).

<table>
<thead>
<tr>
<th>Name</th>
<th>Namespace</th>
</tr>
</thead>
<tbody>
<tr>
<td>IoStandard = IoStandard, 3.4.2.0 (System)</td>
<td>IoStandard</td>
</tr>
<tr>
<td>SM3_Basic = SM3_Basic, 3.5.0.0 (3S - Smart Software Solutions GmbH)</td>
<td>SM3_Basic</td>
</tr>
<tr>
<td>SM3_CNC = SM3_CNC, 3.5.0.30 (3S - Smart Software Solutions GmbH)</td>
<td>SM3_CNC</td>
</tr>
<tr>
<td>L_Ulti, 3.2.0.0 (Lenze)</td>
<td>L_Ulti</td>
</tr>
<tr>
<td>L_SM3_DriveUNT, 3.3.0.0 (Lenze)</td>
<td>L_SM3</td>
</tr>
<tr>
<td>Standard = Standard, 3.5.0.40 (System)</td>
<td>Standard</td>
</tr>
<tr>
<td>L_IoModules = L_IoModule, 3.1.1.0 (Lenze)</td>
<td>L_IoM</td>
</tr>
</tbody>
</table>
12 Restarting the CAN bus

During operation, the CAN bus may have to be restarted. This is e.g. required after serious disturbances such as a cable break.

How to restart the CAN bus:

1. Activate the controller inhibit in the drive controllers.
2. Call the NMT function block (see below) from the CAA_CiA405.lib function library.
   When an error has occurred, set RESET_COMMUNICATION and then START_REMOTE_NODE separately for each CANopen node.

The following steps are only required for Motion devices:
3. Call the SMC3_ReInitDrive function block from the SM3_Basic.lib function library.
4. Call the MC_Reset function block from the SM3_Basic.lib function library.

<table>
<thead>
<tr>
<th>Identifier/data type</th>
<th>Meaning/possible settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>NETWORK USINT</td>
<td>CAN interface number: “1” or “2”</td>
</tr>
<tr>
<td>ENABLE BOOL</td>
<td>Function block enable: • TRUE: Function block is enabled. • FALSE: Function block is not enabled.</td>
</tr>
<tr>
<td>TIMEOUT UDINT</td>
<td>Maximum execution time of function block in [ms]: • The initial value ‘0’ means that the monitoring is deactivated.</td>
</tr>
<tr>
<td>Device Device</td>
<td>CAN node number of slave</td>
</tr>
<tr>
<td>STAT TRANSITION_STATE</td>
<td>NMT status of slave: • STOP_REMOTE_NODE (16#04) • START_REMOTE_NODE (16#05) • RESET_NODE (16#06) • RESET_COMMUNICATION (16#07) • ENTER_PRE OPERATIONAL (16#7F) • ALL_EXCEPT_NMT_AND_SENDER (16#800)</td>
</tr>
</tbody>
</table>
Defining the minimum cycle time of the PLC project

13.1 Determine the task utilisation of the application

In the online mode, the **Monitor** tab of the **Task Configuration** shows current status details and measurements of the cycles, cycle times, and jitters of the tasks contained.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Free_Task</td>
<td>Valid</td>
<td>365287</td>
<td>365215</td>
<td>227</td>
<td>280</td>
<td>1488</td>
<td>25</td>
<td>2</td>
</tr>
<tr>
<td>Logik_Task</td>
<td>Valid</td>
<td>365589</td>
<td>365217</td>
<td>114</td>
<td>104</td>
<td>650</td>
<td>20</td>
<td>-3</td>
</tr>
<tr>
<td>Motion_Task</td>
<td>Valid</td>
<td>355359</td>
<td>365217</td>
<td>299</td>
<td>271</td>
<td>647</td>
<td>20</td>
<td>-2</td>
</tr>
</tbody>
</table>

The values are updated in the same time interval as that used for monitoring the values from the controller.

If the cursor is on a task name field, the values displayed can be reset to 0 by the **Reset** context menu command (right-click the task name field).
13 Defining the minimum cycle time of the PLC project

13.1 Determine the task utilisation of the application

How to determine the task utilisation:

Initial situation: A complete project, e.g. with a CANopen task and 2 lower priority tasks has been created.

1. For a first measurement of the task utilisation, set the cycle times of all cyclic tasks available in the PLC system "high" (e.g. CANopen task = 10 ms, all other cyclic tasks = 20 ms).

2. Use the menu command Online → Login, or log in on the Lenze Controller with <Alt>+<F8>.
   - For this, the PLC program must be error-free.
   - With the log-in, the fieldbus configuration and the PLC program are loaded to the controller.

3. Reset the values displayed on the Monitor tab of the Task Configuration to 0 after the complete run-up of the system.
   Execute the Reset command from the context menu of the task name field.

4. Read the displayed maximum computing time of the task with the highest priority.
   In the illustration above, the max. cycle time of the CANopen task is 647 μs.

The minimum cycle time ($T_{min}$) for a system can be calculated by means of the formula:

$$T_{min} = \text{Task utilisation} \times \text{safety factor}$$

Note!

A safety factor of 1.5 should be included in the calculation.
13 Defining the minimum cycle time of the PLC project

13.2 Optimising the system

---

13.2 Optimising the system

How to optimise the system:

1. Use the menu command **Online → Login**, or log in on the Lenze Controller with **<Alt>+<F8>**.
   - For this, the PLC program must be error-free.
   - With the log-in, the fieldbus configuration and the PLC program are loaded to the controller.

2. Check the task processing times.

3. Optimising the cycle times:
   - If technologically required, the cycle times of the remaining tasks with lower priorities can be decreased.
   - Condition: No task with a low priority must assign more than 60 percent of the corresponding cycle time in its task utilisation.
14 Diagnostics

14.1 Logbook of the Lenze Controller

The following diagnostic options are available:

- **Logbook of the Lenze Controller**
- "Status" tab of the connected field devices
- **Diagnostic codes**
- System bus configurator of the »Engineer«
- »PCAN-View« for diagnostic purposes

14.1 Logbook of the Lenze Controller

The web browser provides access to the logbook of the Lenze Controller. Use the display filter to search for entries concerning CANopen.

**Note!**

The "Clear Log" button deletes the complete logbook on the Lenze Controller without asking for confirmation.
14 Diagnostics
14.2 "Status" tab of the connected field devices

In the »PLC Designer«, there is a Status tab for each field device integrated into the control configuration. This tab can be used to call diagnostic information about the device.

14.3 Diagnostic codes

In the »Engineer«, you can view the diagnostic codes of the Lenze field devices.
14 Diagnostics

14.4 System bus configurator of the »Engineer«

14.4 System bus configurator of the »Engineer«

1. First directly connect your notebook via the USB system bus adapter EMF2177IB to the CANopen bus.

2. Start the system bus configurator ...
   - via the Online button in the menu bar of the »Engineer«
   - or under "Start - Programs - Lenze - Communication"

3. Activate the USB system bus adapter (EMF2177IB).

4. Check the following under the Settings tab ...
   - the baud rate (identical to the setting on the devices)
   - the parameter data channel (= 0):

   The entire address range is scanned. Devices responding on several SDO channels are displayed with several node addresses.

   Example:

   ![Lenze System Bus Configuration Tool](image)

   5. Click the Communication diagnostics button on the General tab to start the search.
14.5 »PCAN-View« for diagnostic purposes

This chapter describes how to use the »PCAN-View« program for diagnostics of your CANopen network.

»PCAN-View« is the basic version of the »PCAN-Explorer« program of PEAK System Technik GmbH. The program allows for simultaneous transmission and reception of CAN messages which can be sent manually and periodically. Errors on the bus system and memory overflows of the controlled CAN hardware are displayed.

14.5.1 Monitor telegram traffic on the CANopen bus

How to monitor the telegram traffic:

1. Connect the Engineering PC directly to the CANopen bus via the EMF2177IB USB system bus adapter.
2. Start the »PCAN-View« program.
3. Connect »PCAN-View« to the USB system bus adapter with the desired baud rate via the "Connect to CAN Hardware" tab.

Example:
4. Now the CAN telegrams are constantly displayed in the "Receive" and "Transmit" windows.

On the basis of the IDs displayed and the IDs in your overview you can assign the telegrams to the devices.

If no telegrams are displayed, this may be caused by various factors:

- Is your Engineering PC connected to the correct CANopen bus?
- Is the correct system bus adapter activated under "System control, CAN hardware"?
- What does it say in the status bar of the »PCAN-View«?
  If it says "Bus Heavy", mostly a node with an incorrect baud rate interferes with the bus communication.
- Are the devices in the "Operational" state?
14.5.2 Setting all CANopen nodes to the "Operational" status

How to set all CANopen nodes to the "Operational" status:

1. Create the following CAN message under "New transmit message":

   ![Image of CAN message creation interface]

   1. Create the following CAN message under "New transmit message":

   2. Select the CAN message in the "Transmit" window and press the <space bar> once to send the CAN message.
14.6 Notes regarding the visualisation using »VisiWinNET«

Problem description:
For visualisation purposes a "VWGET" timer (1 second) is used to read a variable via the CAN bus (SDO) using the OPC server. If no nodes are connected and no information is returned, a time-out is generated. Waiting for the time-out causes the page-turning function of the visualisation program to slow down.

Solution:
The visualisation program can be operated again if the following lines of the time-out are changed in the PostStart.txt file:

```
[HKEY_LOCAL_MACHINE\Drivers\BuiltIn\CAN2Bus1]
"SDOTimeout"=dword:0xc8
"Timeout"=dword:0xc8
```

Otherwise "VWSET" and "VWGET" must not be used for visualisation purposes.
This chapter complements the parameter list in the online help of the Lenze Controller by the parameters of the **MC-CAN2 communication card**.

These parameters ...
- are for instance shown in the Lenze »WebConfig« (Engineering tool for web-based parameterisation);
- are listed in numerically ascending order.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Name: C1031</th>
<th>Device: type key</th>
<th>Data type: VISIBLE_STRING</th>
<th>Index: 23544 = 0x5BF8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification of the card</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Name: C1032</th>
<th>Device: type version</th>
<th>Data type: VISIBLE_STRING</th>
<th>Index: 23543 = 0x5BF7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version number of the card</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Name: C1033</th>
<th>Device: name</th>
<th>Data type: VISIBLE_STRING</th>
<th>Index: 23542 = 0x5BF6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device name of the card</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Name: C1034</th>
<th>Device: software revision</th>
<th>Data type: VISIBLE_STRING</th>
<th>Index: 23541 = 0x5BF5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software version of the card</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Name: C1035</th>
<th>Device: hardware revision</th>
<th>Data type: VISIBLE_STRING</th>
<th>Index: 23540 = 0x5BF4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware version of the card</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Name: C1036</th>
<th>Device: serial number</th>
<th>Data type: VISIBLE_STRING</th>
<th>Index: 23539 = 0x5BF3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serial number of the card</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Parameter reference

### C1037

<table>
<thead>
<tr>
<th>Parameter / Name:</th>
</tr>
</thead>
<tbody>
<tr>
<td>**C1037</td>
</tr>
<tr>
<td>Manufacturer of the card</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

### C1038

<table>
<thead>
<tr>
<th>Parameter / Name:</th>
</tr>
</thead>
<tbody>
<tr>
<td>**C1038</td>
</tr>
<tr>
<td>Manufacturing date of the card</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
# Index

## A
- Adding devices 47
- Adding field devices 47
- Application notes 10
- Application Samples 38, 68
- Application software of the Lenze Controllers 14

## B
- Basic identifier - 8400 Inverter Drives 32
- Basic identifier - 9400 Servo Drives 31
- Basic identifier - I/O system 1000 (EPM-Sxxx) 33
- Baud rates 22
- Bus cable length 24

## C
- C1031 | Device: Identification 107
- C1032 | Device: Version 107
- C1033 | Device: Name 107
- C1034 | Device: Software version 107
- C1035 | Device: Hardware version 107
- C1036 | Device: Serial number 107
- C1037 | Device: Manufacturer 108
- C1038 | Device: Manufacturing date 108
- Cable specification of bus cable 23
- CAN bus connection (5-pole Phoenix Combicon socket) 23
- CAN bus connection (9-pole SUB-D plug) 23
- CAN cable according to ISO 11898-2 23
- CAN connection (5-pole Phoenix Combicon socket) 23
- CAN connection (9-pole SUB-D plug) 23
- CAN wiring 20
- CANbus 48, 81
- CANopen 16
- CANopen (Logic) 17
- CANopen (Motion) 17
- CANopen and EtherCAT (mixed operation) 94
- CANopen hardware for the Lenze Controller 19
- CANopen I/O image 53, 60, 90
- CANopen Logic bus commissioning 38
- CANopen Manager 49
- CANopen Remote Device 52, 89
- CANopen_Manager 82
- Checking the use of repeaters 26
- COB-IDs acc. to DS301 28
- Codes 107
- Commission the field devices 41, 71
- Commissioning steps (short overview) 39
- Commissioning the CANopen Logic bus 38
- Commissioning the CANopen Motion bus 67
- Communication profile 22
- Communication settings 44, 74
- Compiling the PLC program code 66, 93
- Configuration 76
- Configuring the communication parameters 44, 74

## D
- Connecting the Engineering PC to the Lenze Controller 36
- Controller as gateway 36
- Conventions used 8
- Create a project folder 40, 70
- Creating a control configuration 47
- Creating a control configuration (Motion devices) 80
- Creating a Motion task 76
- Creating a PLC program with a target system (Logic) 42
- Creating a PLC program with a target system (Motion) 72
- Creating target system (Logic) 42
- Creating target system (Motion) 72
- Creating the program code for controlling the Logic field device 59
- Creating the program code for controlling the Motion field device 92
- Cross communication 22
- Cross communication between the slaves 55
- Cycle synchronisation 22
- Cycle time 22

## E
- Defining the minimum cycle time of the PLC project 97
- Determine the task utilisation of the application 97
- Determining cable lengths / the number of repeaters 25
- Device
  - Hardware version (C1035) 107
  - Identification (C1031) 107
  - Manufacturer (C1037) 108
  - Manufacturing date (C1038) 108
  - Name (C1033) 107
  - Serial number (C1036) 107
  - Software version (C1034) 107
  - Version (C1032) 107
- Device specifications of the field devices 30
- Diagnostic codes 101
- Diagnostics 100
- Diagnostics with »PCAN-View« 103
- Direct coupling (PC - CANopen bus) 36

## F
- EASY Navigator 21
- E-mail to Lenze 111
- Engineering software 21
- Engineering tools 21

## G
- General data 22
Index

I
Importing device description files 46
Importing missing devices 46
Installing field devices 35
Interfaces for fieldbus communication 15

L
Layout of the safety instructions 10
Lenze Engineering tools 21
Logbook of the Lenze Controller 100
Logging in on the Lenze Controller with the »PLC Designer« 66, 93

M
MC 19
MC-CAN2 communication card 19
Mixed operation of CANopen and EtherCAT 94
Motion bus (CANopen) commissioning 67

N
Navigator 21
Network topology 22
NMT 96
Node address 22
Number of DI + DO (bits/ms) 22
Number of drives/ms 22
Number of nodes 22

O
Optimising the system 99

P
Parallel operation of two synchronised CAN buses 84
Parameter data 22
Parameter reference 107
PCAN view for CAN diagnostics 103
PDO mapping 53, 90
Planning of CANopen network 27
Planning the CANopen network 27
Preparing the restart 61, 92

R
Receive PDO mapping 57
Restart of the EPM-Sxxx (I/O system 1000) 63
Restart of the I/O system 1000 (EPM-Sxxx) 63
Restarting the CAN bus 96

S
Safety instructions 10, 11
Sample projects (Application Samples) 38, 68
Screenshots 6
Segment cable length 24
Setting of CAN parameters (Logic devices) 52
Setting of CAN parameters (Motion devices) 89
Setting of PDO mapping (Logic devices) 52
Setting of PDO mapping (Motion devices) 89
Setting SoftMotion parameters 86
Setting sync generation 49, 82
Setting the baud rate 35
Setting the node addresses 35
Signal propagation delay 22
SM3_Drive_Lenze.lib 95
SM3_Drive_Lenze.lib function library 95
SMC3_ReInitDrive 96
SoftMotion drive
   Basic Parameters 88
   Scaling/Mapping 87
Software 21
Special case: Different slave and master turn-on times 34
Standards 22
Start parameters of the Servo Drives 9400 HighLine CiA 402 66, 93
Starting the PLC program 66, 93
Status 101
Status tab of the connected field devices 101
System bus (CAN) 16
System bus configurator 102
System structure of Controller-based Automation 13

T
Target group 6
Task configuration 76, 97
Technical data 22
Technical data of the MC-CAN2 communication card 23
Terms 9
Total cable length 24
Transmit PDO mapping 57

U
Use of repeaters 25
Using PCAN view 103
Your opinion is important to us

These instructions were created to the best of our knowledge and belief to give you the best possible support for handling our product. Perhaps we have not succeeded in achieving this objective in every respect. If you have suggestions for improvement, please e-mail us to:

feedback-docu@lenze.com

Thank you very much for your support.

Your Lenze documentation team