

## IF55 LIN CB



DS406 encoder profile

- SSI to CANopen converter
- Suitable for SSI linear encoders
- Accepts MSB & LSB Aligned protocols up to 30 bits
- Cable and M12 connector outputs
- CANopen in compliance with DS 301 and DS 406 profiles

#### Suitable for the following models:

- IF55 LIN CB
- IF55 LIN CB-C

#### General Contents

1 - Safety summary	19
2 - Identification	21
3 - Mechanical installation	22
4 - Electrical connections	25
5 - Getting started	34
6 - CANopen® interface (DS 406)	39
7 - Setting-up	82
8 - Default parameters list	85

This publication was produced by Lika Electronic s.r.l. 2019. All rights reserved. Tutti i diritti riservati. Alle Rechte vorbehalten. Todos los derechos reservados. Tous droits réservés.

This document and information contained herein are the property of Lika Electronic s.r.l. and shall not be reproduced in whole or in part without prior written approval of Lika Electronic s.r.l. Translation, reproduction and total or partial modification (photostat copies, film and microfilm included and any other means) are forbidden without written authorisation of Lika Electronic s.r.l.

The information herein is subject to change without notice and should not be construed as a commitment by Lika Electronic s.r.l. Lika Electronic s.r.l. reserves the right to make all modifications at any moments and without forewarning.

This manual is periodically reviewed and revised. As required we suggest checking if a new or updated edition of this document is available at Lika Electronic s.r.l.'s website. Lika Electronic s.r.l. assumes no responsibility for any errors or omissions in this document. Critical evaluation of this manual by the user is welcomed. Your comments assist us in preparation of future documentation, in order to make it as clear and complete as possible. Please send an e-mail to the following address [info@lika.it](mailto:info@lika.it) for submitting your comments, suggestions and criticisms.

The logo for Lika Electronic s.r.l. features the word "lika" in a bold, lowercase, sans-serif typeface. The letters are black and have a modern, clean appearance.

# General contents

User's guide.....	1
General contents.....	3
Subject index.....	6
Typographic and iconographic conventions.....	7
Preliminary information.....	8
Glossary of CANopen terms.....	9
<b>1 – Safety summary.....</b>	<b>19</b>
1.1 Safety.....	19
1.2 Electrical safety.....	19
1.3 Mechanical safety.....	20
<b>2 – Identification.....</b>	<b>21</b>
<b>3 – Mechanical installation.....</b>	<b>22</b>
3.1 Overall dimensions.....	22
3.2 Installation on panel (Figure 2).....	23
3.3 Installation with DIN rail clip (Figure 3).....	23
<b>4 – Electrical connections.....</b>	<b>25</b>
4.1 Converter's connection cap (Figure 5).....	25
4.2 SSI connection (Figure 4).....	26
4.3 CANopen converter with PGs: CB version (Figure 4 and Figure 6).....	27
4.4 CANopen converter with M12 connectors: CB-C version (Figure 4 and Figure 7).....	28
4.5 Ground connection (Figure 4).....	28
4.6 Shield connection.....	29
4.7 POWER SUPPLY DIP switch (Figure 8).....	29
4.8 Baud rate: DIP A (Figure 6 and Figure 7).....	30
4.9 Node number: DIP B (Figure 6 and Figure 7).....	31
4.10 RT Bus termination (Figure 6 and Figure 7).....	32
4.11 Diagnostic LEDs (Figure 4).....	33
<b>5 – Getting started.....</b>	<b>34</b>
5.1 Quick reference.....	37
<b>6 – CANopen® interface (DS 406).....</b>	<b>39</b>
6.1 EDS file.....	39
6.2 State machine.....	39
6.2.1 Initialization state.....	40
6.2.2 Pre-operational state.....	40
6.2.3 Operational state.....	40
6.2.4 Stopped state.....	40
6.3 Communication objects.....	41
6.3.1 Pre-defined connection set.....	41
6.4 NMT objects.....	42
6.5 Boot-up objects.....	42
6.6 PDO objects.....	42
PDO1 Cyclic mode: cyclic transmission.....	42
PDO2 and PDO3 Sync mode: synchronous transmission.....	43
6.7 SDO objects.....	43
6.7.1 Command.....	44

6.8 Object dictionary.....	45
6.8.1 Communication Profile Area objects (DS 301).....	47
1000-00 Device type.....	47
1001-00 Error register.....	47
1003 Predefined error field.....	47
1005-00 COB-ID SYNC message.....	47
1008-00 Manufacturer Device name.....	47
1009-00 Manufacturer Hardware version.....	47
100A-00 Manufacturer Software version.....	47
100C-00 Guard time.....	48
100D-00 Life time factor.....	48
1010-01 Store parameters.....	48
1011-01 Restore default parameters.....	48
1014-00 COB-ID EMCY.....	49
1015-00 Inhibit time EMCY.....	49
1018 Identity object.....	49
1800 TPDO1 parameters.....	49
1801 TPDO2 parameters.....	51
1802 TPDO3 parameters.....	52
1A00-01 TPDO1 mapping parameter.....	54
1A01-01 TPDO2 mapping parameter.....	54
1A02-01 TPDO3 mapping parameter.....	54
6.8.2 Manufacturer Specific Profile Area objects.....	55
2104-00 Limit switch min.....	55
2105-00 Limit switch max.....	55
2200-01 Code Type (BIN/GRAY).....	55
2200-02 SSI Protocol.....	56
2200-03 Number of SSI clocks.....	56
2200-04 Physical Total Resolution [bits].....	57
2200-05 Physical pulse measure [nm].....	58
2200-06 Bypass.....	59
3000-00 Baud rate.....	59
3001-00 Node-ID.....	59
6.8.3 Standardised Device Profile Area objects (DS 406).....	60
6000-00 Operating parameters.....	60
Scaling function control.....	60
Measuring direction.....	62
Limit switch min.....	62
Limit switch max.....	62
6001-00 Total measuring range.....	62
6002-00 Programmable total resolution.....	65
6003-00 Preset value.....	66
6004-00 Position value.....	68
6005-01 Measuring step setting.....	68
6008-00 High precision position value.....	72
6200-00 Cyclic timer.....	72
6500-00 Operating status.....	73
Scaling.....	73
Measuring direction.....	73

Limit switch min.....	73
Limit switch max.....	74
Current operating state.....	74
6501-00 Measuring step.....	74
6502-00 Number of distinguishable revolutions.....	74
6504-00 Supported alarms.....	75
6506-00 Supported warnings.....	75
6507-00 Profile and software version.....	75
6508-00 Operating time.....	75
6509-00 Offset value.....	75
650A-01 Manufacturer offset value.....	76
650B-00 Serial number.....	76
6.9 SDO abort codes.....	77
6.10 Emergency (EMCY) objects.....	78
6.11 Node guarding protocol.....	79
<b>7 – Setting-up.....</b>	<b>82</b>
7.1 Setting the Operational, Pre-operational state.....	82
7.2 Reading the value of the measuring step.....	82
7.3 Setting the value of the measuring step.....	82
7.4 Setting the operating parameters.....	82
7.5 Setting the Preset value.....	83
7.6 Setting the SYNC counter.....	83
7.7 Disabling the SYNC mode.....	83
7.8 Enabling the Cyclic mode.....	84
<b>8 – Default parameters list.....</b>	<b>85</b>

# Subject index

## 1

1000-00 Device type.....	47
1001-00 Error register.....	47
1003 Predefined error field.....	47
1005-00 COB-ID SYNC message.....	47
1008-00 Manufacturer Device name.....	47
1009-00 Manufacturer Hardware version.....	47
100A-00 Manufacturer Software version.....	47
100C-00 Guard time.....	48
100D-00 Life time factor.....	48
1010-01 Store parameters.....	48
1011-01 Restore default parameters.....	48
1014-00 COB-ID EMCY.....	49
1015-00 Inhibit time EMCY.....	49
1018 Identity object.....	49
1800 TPDO1 parameters.....	49
1801 TPDO2 parameters.....	51
1802 TPDO3 parameters.....	52
1A00-01 TPDO1 mapping parameter.....	54
1A01-01 TPDO2 mapping parameter.....	54
1A02-01 TPDO3 mapping parameter.....	54

## 2

2104-00 Limit switch min.....	55
2105-00 Limit switch max.....	55
2200-01 Code Type (BIN/GRAY).....	55
2200-02 SSI Protocol.....	56
2200-03 Number of SSI clocks.....	56
2200-04 Physical Total Resolution [bits].....	57
2200-05 Physical pulse measure [nm].....	58
2200-06 Bypass.....	59

## 3

3000-00 Baud rate.....	59
3001-00 Node-ID.....	59

## 6

6000-00 Operating parameters.....	60
6001-00 Total measuring range.....	62
6002-00 Programmable total resolution.....	65
6003-00 Preset value.....	66
6004-00 Position value.....	68
6005-01 Measuring step setting.....	68
6008-00 High precision position value.....	72
6200-00 Cyclic timer.....	72
6500-00 Operating status.....	73
6501-00 Measuring step.....	74

6502-00 Number of distinguishable revolutions.....	74
6504-00 Supported alarms.....	75
6506-00 Supported warnings.....	75
6507-00 Profile and software version.....	75
6508-00 Operating time.....	75
6509-00 Offset value.....	75
650A-01 Manufacturer offset value.....	76
650B-00 Serial number.....	76

## C

COB-ID of PDO1.....	49
COB-ID of the PDO2.....	51
COB-ID of the PDO3.....	52
Current operating state.....	74

## F

Flash memory error.....	78
-------------------------	----

## I

Initialization.....	40
---------------------	----

## L

Last error occurred.....	47
Limit switch max.....	74
Limit switch max.....	62
Limit switch min.....	73
Limit switch min.....	62

## M

Measuring direction.....	62, 73
--------------------------	--------

## N

Node guarding error.....	78
Number of occurred errors.....	47

## O

Operational.....	40
------------------	----

## P

Pre-operational.....	40
Previous errors occurred.....	47
Product code.....	49

## R

Revision number.....	49
----------------------	----

## S

Scaling.....	73
Scaling function control.....	60
Stopped.....	40

## T

Transmission type.....	50p., 53
------------------------	----------

## V




Vendor-ID.....	49
----------------	----

# Typographic and iconographic conventions

In this guide, to make it easier to understand and read the text the following typographic and iconographic conventions are used:

- parameters and objects both of the device and the interface are coloured in **GREEN**;
- alarms are coloured in **RED**;
- states are coloured in **FUCSIA**.

When scrolling through the text some icons can be found on the side of the page: they are expressly designed to highlight the parts of the text which are of great interest and significance for the user. Sometimes they are used to warn against dangers or potential sources of danger arising from the use of the device. You are advised to follow strictly the instructions given in this guide in order to guarantee the safety of the user and ensure the performance of the device. In this guide the following symbols are used:

	This icon, followed by the word <b>WARNING</b> , is meant to highlight the parts of the text where information of great significance for the user can be found: user must pay the greatest attention to them! Instructions must be followed strictly in order to guarantee the safety of the user and a correct use of the device. Failure to heed a warning or comply with instructions could lead to personal injury and/or damage to the unit or other equipment.
	This icon, followed by the word <b>NOTE</b> , is meant to highlight the parts of the text where important notes needful for a correct and reliable use of the device can be found. User must pay attention to them! Failure to comply with instructions could cause the equipment to be set wrongly: hence a faulty and improper working of the device could be the consequence.
	This icon is meant to highlight the parts of the text where suggestions useful for making it easier to set the device and optimize performance and reliability can be found. Sometimes this symbol is followed by the word <b>EXAMPLE</b> when instructions for setting parameters are accompanied by examples to clarify the explanation.

# Preliminary information

This guide is designed to describe the technical characteristics, installation and use of the SSI to CANopen gateways of the **IF55 series**.

IF55 series gateways allow the **integration of SSI encoders**, both rotary and linear, **into conventional fieldbuses or industrial Ethernet networks**.

The present manual is specifically designed to describe the SSI to CANopen IF55 model for linear encoders (order code IF55 LIN CB). For information on the SSI to CANopen IF55 model for rotary encoders (order code IF55 ROT CB) refer to the specific documentation.

For information on the gateways designed for the integration of other fieldbus/Ethernet encoders (for example SSI to Profibus: order codes IF55 ROT PB and IF55 LIN PB; and SSI to EtherCAT: order codes IF55 ROT EC and IF55 LIN EC), refer to the specific documentation.

Please note that the present manual does not prescind from the user's guide of the SSI encoder it has to be connected to. Please read carefully the encoder's documentation before installing, connecting and operating the measuring system.

For detailed technical specifications please refer also to the product datasheet.

CANopen connection cap as follows:

CB	CANopen interface with PGs
CB-C	CANopen interface with M12 connectors

For any further information please refer to the product data sheet.

To make it easier to read the text, this guide can be divided into two main sections.

In the first section general information concerning the safety, the mechanical installation and the electrical connection as well as tips for setting up and running properly and efficiently the unit are provided.

In the second section, entitled **CANopen Interface (DS406)**, both general and specific information is given on the CANopen interface. In this section the interface features and the objects implemented in the unit are fully described.



# Glossary of CANopen terms

CANopen, like many other networking systems, has a set of unique terminology. Table below contains a few of the technical terms used in this guide to describe the CANopen interface. They are listed in alphabetical order. The Glossary is owned and copyrighted by the CAN in Automation international users' and manufacturers' group.

<b>Application layer</b>	The application layer is the communication entity of the OSI (Open System Interface) reference model. It provides communication services to the application program.
<b>Application objects</b>	Application objects are signals and parameters of the application program visible at the application layer API (application programming interface).
<b>Application profile</b>	Application profiles define all communication objects and application objects in all devices that the network consists of.
<b>Asynchronous PDO</b>	An asynchronous PDO is transmitted whenever a defined internal event occurs. This event may also be the elapsing of the PDO's event timer. If an asynchronous PDO is received the protocol software immediately updates the mapped objects in the Object Dictionary.
<b>Boot-up message</b>	CANopen communication service transmitted whenever a node enters the <b>Pre-operational</b> state after initialization.
<b>Bus</b>	Topology of a communication network, where all nodes are reached by passive links, which allows transmission in both directions.
<b>Bus analyser</b>	Tool, which monitors the bus and displays the transmitted bits. There are bus analysers available on the physical layer, the data link layer, and different application layers (e.g. CANopen or DeviceNet).
<b>Bus arbitration</b>	If at the very same moment several nodes try to access the bus, an arbitration process is necessary. At the end of this process, only one node has bus access. The bus arbitration process used in CAN protocol is CMA/CD (Carrier Sense Multiple Access/Collision Detection) with AMP (Arbitration on Message Priority). This allows bus arbitration without destruction of messages.
<b>Bus length</b>	The network cable length between the both termination resistors. The bus length of CANopen networks is limited by the used transmission rate. At 1 Mbps the maximum length is 25 m. When using lower transmission rates, longer bus lines may be used: at 50 kbps a length of 1 km is possible.
<b>Bus off state</b>	The CAN controllers switch to bus off state when the TEC (transmit error counter) has reached 255. During bus off state, the CAN controller transmits recessive bits. When a CANopen

	device recovers from bus off state, it has to transmit the boot-up message and it is recommended to send an Emergency message with the appropriate error code.
<b>CAN</b>	Controller Area Network (CAN) is a serial bus system originally developed by the Robert Bosch GmbH. It is internationally standardized by ISO 11898-1. CAN has been implemented by many semiconductor manufacturers.
<b>CAN protocol controller</b>	The CAN protocol controller is part of a CAN module performing data en-/de-capsulation, bit-timing, CRC, bit-stuffing, error handling, failure confinement, etc.
<b>CAN transceiver</b>	The CAN transceiver is connected to the CAN controller and to the bus lines. It provides the line transmitter and the receiver. There are high-speed, fault-tolerant, and single-wire transceivers available as well as transceivers for power-line or fiber optic transmissions.
<b>CANopen</b>	Family of profiles for embedded networking in industrial machinery, medical equipment, building automation (e.g. lift control systems, electronically controlled doors, integrated room control systems), railways, maritime electronics, truck-based superstructures, off-highway and off-road vehicles, etc.
<b>CANopen application layer</b>	The CANopen application layer and communication profile is standardized by EN 50325-4. It defines communication services and objects. In addition, it specifies the Object Dictionary and the network management (NMT).
<b>CANopen Manager</b>	The CANopen manager is responsible for the management of the network. The CANopen manager device shall include the NMT (network management) Master, the SDO (service data object) manager, and the Configuration manager.
<b>CANopen Safety</b>	Communication protocol allowing transmission of safety-relevant data. The protocol requires just one physical CAN network. Redundancy is achieved by sending each message twice with bit-wise inverted content using two identifiers differing at least in two bits.
<b>Certification</b>	Official compliance test of components or devices to a specific standard. CiA officially certifies CANopen devices.
<b>CiA DR 303</b>	Draft recommendation for CANopen cabling and connector pin assignments, coding of prefixes and SI unit as well as LED usage.
<b>CiA DS 102</b>	Draft standard for high-speed transmission according to ISO 11898-2 using 9-pin D-sub connectors.
<b>CiA DS 301</b>	The CANopen application layer and communication profile specification covers the functionality of CANopen NMT (network management) Slave devices.
<b>CiA DS 401</b>	The CANopen device profile for generic I/O modules covers the definition of digital and analogue input and output devices.
<b>CiA DS 404</b>	The CANopen device profile for measuring devices and closed-

	loop controllers supports also multi-channel devices.
<b>CiA DS 406</b>	The CANopen device profile for encoders defines the communication of rotating as well as linear sensors.
<b>CiA DSP 302</b>	The draft standard proposal for programmable CANopen devices includes CANopen manager functions, dynamic SDO connections, standardized boot-up procedure for NMT Slaves as well as program download.
<b>CiA DSP 304</b>	The CANopen safety protocol specification is approved by German authorities and is compliant to SIL class 3 applications.
<b>CiA DSP 305</b>	The Layer Setting Services (LSS) specify how to set node-ID and transmission rate via the CANopen network.
<b>CiA DSP 306</b>	This draft standard proposal defines format and content of Electronic Data Sheets (EDS) to be used in configuration tools.
<b>CiA DSP 308</b>	The CANopen framework for maritime applications defines redundancy of networks including swapping mechanism for SDOs and PDOs.
<b>CiA DSP 309</b>	Set of gateway specifications for CANopen to Ethernet-based networks (e.g. Modbus TCP(IP)).
<b>CiA DSP 402</b>	The CANopen device profile for drives and motion controllers defines the interface to frequency inverters, servo controllers as well as stepper motors.
<b>CiA DSP 405</b>	The CANopen device and interface profile for IEC 61131-3 compatible controllers is based on the CiA DSP 302 specification using network variables to be mapped into PDOs, and function blocks for SDO services, etc.
<b>CiA DSP 407</b>	The CANopen application profile for passenger information systems developed in cooperation with the German VDV specifies interfaces for a range of devices including displays, ticket printers, passenger counting units, main onboard computer, etc.
<b>CiA DSP 408</b>	The CANopen device profile for hydraulic controllers and proportional valves is compliant to the bus-independent VDMA device profile.
<b>CiA DSP 410</b>	The CANopen device profile for inclinometer supports 16-bit as well as 32-bit sensors.
<b>CiA DSP 412</b>	The CANopen device profiles for medical equipment specify the interfaces for x-ray collimators, x-ray generators, stands and tables.
<b>CiA DSP 413</b>	The CANopen interface profiles for in-vehicle truck gateways specify gateways to ISO 11992, J1939, and other in-vehicle networks. The CANopen network is mainly used for truck- or trailer-based superstructures, e.g. as in garbage trucks, truck-mounted cranes, and concrete mixers.
<b>CiA DSP 414</b>	The CANopen device profile for weaving machines specifies

	the interface for feeder sub-systems.
<b>CiA DSP 415</b>	The CANopen application profile for asphalt pavers specifies interfaces to different devices used in road construction machinery.
<b>CiA DSP 416</b>	The CANopen application profile for building doors specifies interfaces for locks, sensors, and other devices used in electronically controlled building doors.
<b>CiA DSP 417</b>	The CANopen application profile for lift control specifies the interfaces for car controller, door controller, call controller and other controllers as well as for car units, door units, input panels, and display units, etc.
<b>CiA DSP 418</b>	The CANopen device profile for battery modules specifies the interface to communicate with battery chargers.
<b>CiA DSP 419</b>	The CANopen device profile for battery charger specifies the interface to communicate with the battery module.
<b>CiA DSP 420</b>	The CANopen device profile family for extruder downstream devices defines interfaces for puller, corrugator and saw devices.
<b>CiA DSP 421</b>	The CANopen device profile for railways specifies interfaces to sub-systems such as diesel engines, brake controllers, door controllers, etc.
<b>CiA DSP 422</b>	The CANopen application profile for municipal vehicles defines the communication of sub-systems used in garbage trucks.
<b>CiA TR 308</b>	This technical report specifies some timings for CANopen performance testing tools.
<b>Client / Server communication</b>	In a Client/Server communication the Client initiates the communication with the Server. It is always a point-to-point communication.
<b>Client SDO</b>	The Client SDO initiates the SDO communication by means of reading or writing to the Object Dictionary of the Server device.
<b>COB ID</b>	The COB ID is the object specifying the CAN message identifier and additional parameters such as valid/invalid and remote frame support.
<b>Communication object (COB)</b>	A communication object is one or more CAN messages with a specific functionality, e.g. PDO, SDO, Emergency, Time, or Error Control.
<b>Communication profile</b>	A communication profile defines the content of communication objects such as Emergency, Time, Sync, Heartbeat, NMT, etc. in CANopen.
<b>Configuration Manager</b>	The Configuration Manager (CMT) provides mechanisms for configuration of CANopen devices during boot-up.
<b>Confirmed communication</b>	Confirmed communication services require a bi-directional communication, meaning that the receiving node sends a confirmation that the message has been received successfully.

<b>Conformance test plan</b>	Definitions of test cases that have to be passed successfully in order to achieve conformance to a communication standard. The conformance test plan for CAN is standardized by ISO 16845.
<b>Conformance test tool</b>	A conformance test tool is the implementation of a conformance test plan.
<b>Consumer</b>	In CAN networks a receiver of messages is called a consumer meaning the acceptance filter is opened.
<b>D-sub connector</b>	Standardized connectors. Most common in use is the 9-pin D-sub connector (DIN 41652); its pin-assignment for CAN networks is specified in CiA DS 102.
<b>Data link layer</b>	Second layer in the OSI reference model providing basic communication services. The CAN data link layer defines data, remote, error, and overload frames.
<b>Data type</b>	Object attribute in CANopen defining the format, e.g. UNSIGNED8, INTEGER16, BOOLEAN, etc.
<b>Default value</b>	Object attribute in CANopen defining the pre-setting of not user-configured objects after power-on or application reset.
<b>Device profile</b>	A device profile defines the device-specific communication services including the configuration services in all details.
<b>Draft Recommendation (DR)</b>	This kind of recommendation is not fixed, but it is published. CiA's draft recommendations are not changed within one year.
<b>Draft Standard (DS)</b>	This kind of standard is not fixed, but it is published. CiA's draft standards are not changed within one year.
<b>Draft Standard Proposal (DSP)</b>	This kind of standard is a proposal, but it is published. CiA's draft standard proposals may be changed anytime without notification.
<b>EDS checker</b>	Software tool that checks the conformity of electronic data sheets. The CANopen EDS checker is available on CiA's web-site to be downloaded.
<b>EDS generator</b>	Software tool that generates CANopen electronic data sheets.
<b>Electronic Data Sheet (EDS)</b>	Electronic data sheets describe the functionality of a device in a standardized manner.
<b>Emergency message</b>	Pre-defined communication service in CANopen mapped into a single 8-byte data frame containing a 2-byte standardized error code, the 1-byte error register, and 5-byte manufacturer-specific information. It is used to communicate device and application failures.
<b>EN 50325-4</b>	CENELEC standard defining the CANopen application layer (version 4.0).
<b>Entry category</b>	Object attribute in CANopen defining whether this object is mandatory or optional.
<b>Error code</b>	CANopen specifies standardized error codes transmitted in emergency messages.

<b>Error control message</b>	The CANopen error control messages are mapped to a single 1-byte CAN data frame assigned with a fixed identifier that is derived from the device's Node ID. It is transmitted as boot-up message before entering <b>Pre-operational</b> state after initialization, and it is transmitted if remotely requested by the NMT Master (node guarding) or periodically by the device (heart-beat).
<b>Event driven</b>	Event driven messages are transmitted when a defined event occurs in the node. This may be a change of input states, elapsing of a local timer, or any other local event.
<b>Event timer</b>	The event timer is assigned in CANopen to one PDO. It defines the frequency of transmission.
<b>Expedited SDO</b>	This is a confirmed communication service of CANopen (peer-to-peer). It is made up by one SDO initiate message of the Client node and the corresponding confirmation message of the Server node. Expedited SDOs are used if not more than 4 byte of data has to be transmitted.
<b>Flying Master</b>	In safety-critical applications, it may be required that a missing NMT Master is substituted automatically by another stand-by NMT Master. This concept of redundancy is called Flying Master.
<b>Form error</b>	A corruption of one of the pre-defined recessive bits (CRC delimiter, ACK delimiter and EOF) is regarded as a form error condition that will cause the transmission of an error frame in the very next bit-time.
<b>Function code</b>	First four bits of the CAN identifier in the CANopen pre-defined identifier set indicating the function of the communication object (e.g. TPDO_1 or Error Control message).
<b>Galvanic isolation</b>	Galvanic isolation in CAN networks is performed by optocouplers or transformers placed between CAN controller and CAN transceiver chip.
<b>Gateway</b>	Device with at least two network interfaces transforming all seven OSI (open system interconnection) protocol layers, e.g. CANopen-to-Ethernet gateway.
<b>Heartbeat</b>	CANopen uses heartbeat message to indicate that a node is still alive. This message is transmitted periodically.
<b>Heartbeat consumer time</b>	The heartbeat consumer time defines the time when a node is regarded as no longer alive due to a missing heartbeat message.
<b>Heartbeat producer time</b>	The heartbeat producer time defines the transmission frequency of a heartbeat message.
<b>Identifier</b>	In general, the term identifier refers to a CAN message identifier. The CAN message identifier identifies the content of a data frame. The identifier of a remote frame corresponds to the identifier of the requested data frame. The identifier includes implicitly the priority for the bus arbitration.

<b>Index</b>	16-bit address to access the CANopen dictionary; for array and records the address is extended by an 8-bit Subindex.
<b>Inhibit timer</b>	Object in CANopen for PDOs and Emergency messages that forbids for the specified time (inhibit time) a transmission of this communication object.
<b>Initialization state</b>	NMT Slave state in CANopen that is reached automatically after power-on and communication or application reset.
<b>Interface profile</b>	CANopen profile that describes just the interface and not the application behaviour of device, e.g. gateway and bridge devices.
<b>ISO 11898-1</b>	International standard defining the CAN data link layer including LLC, MAC and PLS sub-layers.
<b>ISO 11898-2</b>	International standard defining the CAN high-speed MAU.
<b>Life guarding</b>	Method in CANopen to detect that the NMT Master does not guard the NMT Slave any more. This not recommended for new systems designs.
<b>Line topology</b>	Networks, where all nodes are connected directly to one bus line. CAN networks use theoretically just line topologies without any stub cable. However in practice you find tree and star topologies as well.
<b>Master</b>	Communication or application entity that is allowed to control a specific function. In networks this is for example the initialization of a communication service.
<b>Multiplexed PDO (MPDO)</b>	The MPDO is made of 8 byte including one control byte, three multiplexer bytes (containing the 24-bit Index and Subindex), and four bytes of object data.
<b>Network length</b>	Bus length. The network cable length between both termination resistors. The bus length of CANopen networks is limited by the used transmission rate. At 1 Mbps the maximum length is 25 m. When using lower transmission rates, longer bus lines may be used: at 50 kbps a length of 1 km is possible.
<b>Network management</b>	Entity responsible for the network boot-up procedure and the optional configuration of nodes. It also may include node-supervising functions such as node guarding.
<b>Network variables</b>	Network variables are used in programmable CANopen devices to be mapped into PDOs after programming the device.
<b>NMT</b>	Network management in CANopen.
<b>NMT Master</b>	The NMT Master device performs the network management by means of transmitting the NMT message. With this message, it controls the state machines of all connected NMT Slave devices.
<b>NMT Slave</b>	The NMT Slaves receive the NMT message, which contains commands for the NMT state machine implemented in CANopen devices.

<b>NMT state machine</b>	The NMT state machines support different states and the highest prior CAN message transmitted controls the transition to the states by the NMT Master.
<b>Node guarding</b>	Mechanism used in CANopen and CAL to detect bus off or disconnected devices. The NMT Master sends a remote frame to the NMT Slave that is answered by the corresponding error control message.
<b>Node ID</b>	Unique identifier for a device required by different CAN-based higher-layer protocols in order to assign CAN identifiers to this device, e.g. in CANopen and DeviceNet. In the pre-defined connection set of CANopen some of the CAN message identifiers are derived from the assigned Node ID.
<b>Object Dictionary</b>	Heart of each CANopen device containing all communication and application objects.
<b>Operational state</b>	In the NMT <b>Operational</b> state all CANopen communication services are available.
<b>PDO mapping</b>	In PDOs, there may be mapped up to 64 objects. The PDO mapping is described in the PDO mapping parameters.
<b>Pin assignment</b>	Definition of the use of connector pins.
<b>Pre-defined connection set</b>	The pre-defined connection set is a default assignment of CAN message identifiers to CANopen communication objects. Some CANopen communication objects are distributed in broadcast (NMT message, Sync message, Time message) and others are transmitted between NMT Master device and dedicated NMT Slave devices (PDO, SDO, Emergency, and Error Control). This default assignment guarantees that the CAN message identifiers are uniquely assigned in the network, if the node-ID has been assigned uniquely.
<b>Pre-operational state</b>	In the NMT <b>Pre-operational</b> state no CANopen PDO communication is allowed.
<b>Process Data Object (PDO)</b>	Communication object defined by the PDO communication parameter and PDO mapping parameter objects. It is an unconfirmed communication service without protocol overhead.
<b>Producer</b>	In CAN networks a transmitter of messages is called a producer.
<b>Protocol</b>	Formal set of conventions and rules for the exchange of information between nodes, including the specification of frame administration, frame transfer and physical layer.
<b>Receiver</b>	A CAN node is called receiver or consumer, if it is not transmitter and the bus is not idle.
<b>Redundant networks</b>	In some safety-critical applications (e.g. maritime systems), redundant networks may be required that provide swapping capability in case of detected communication failures.
<b>Remote frame</b>	With a remote frame another node is requested to transmit



	the corresponding data frame identified by the very same identifier. The remote frame's DLC has the value of the corresponding data frame DLC. The data field of the remote frame has a length of 0 byte.
<b>Remote transmission request (RTR)</b>	Bit in the arbitration field indicating if the frame is a remote frame (recessive value) or a data frame (dominant value).
<b>Repeater</b>	Passive component that refreshes CAN bus signals. It is used to increase the maximum number of nodes, or to achieve longer networks (>1 km), or to implement tree or meshed topologies.
<b>Reset application</b>	This NMT command resets all objects in CANopen devices to the default values or the permanently stored configured values.
<b>Reset communication</b>	This NMT command resets only the communication objects in CANopen devices to the default values or the permanently stored configured values.
<b>RPDO</b>	The Receive Process Data Object (RPDO) is a communication object that is received by a CANopen device.
<b>SDO block transfer</b>	SDO block transfer is a CANopen communication service for increasing downloading. In SDO block transfer, the confirmation is sent after the reception of a number of SDO segments.
<b>SDO Manager</b>	The SDO Manager handles the dynamic establishment of SDO connections. It resides on the very same node as the NMT Master.
<b>Segmented SDO</b>	If objects longer than 4 bytes are transmitted by means of SDO services, a segmented transfer is used. The number of segments is theoretically not limited.
<b>Server SDO</b>	The Server SDO receives the SDO messages from the corresponding SDO Client and responds to each SDO message or to a block of SDO messages (SDO block transfer).
<b>Service Data Object (SDO)</b>	SDOs provide the access to entries in the CANopen Object Dictionary. An SDO is made up of at least two CAN messages with different identifiers. SDOs are always confirmed point-to-point communication services.
<b>SI unit</b>	International system of units for physical values as specified in ISO 1000:1983.
<b>Stopped state</b>	NMT state in which only NMT messages are performed and under some conditions error control messages are transmitted.
<b>Sub-index</b>	8-bit sub-address to access the sub-objects of arrays and records.
<b>Suspend transmission</b>	CAN controllers in error passive mode have to wait additional 8 bit-times before the next data or remote frame may be transmitted.
<b>SYNC message</b>	Dedicated CANopen message forcing the receiving nodes to

	sample the inputs mapped into synchronous TPD0s. Receiving this message causes the node to set the outputs to values received in the previous synchronous RPDO.
<b>Termination resistor</b>	In CAN high-speed networks with bus topology, both ends are terminated with resistors in order to suppress reflections.
<b>TIME message</b>	Standardized message in CANopen containing the time as a 6-byte value given as ms after midnight and days after 1st January 1984.
<b>TPDO</b>	The Transmit Process Data Object (TPDO) is a communication object that is transmitted by a CANopen device.
<b>Transmission type</b>	CANopen object defining the scheduling of a PDO.
<b>Value definition</b>	Detailed description of the value range in CANopen profiles.
<b>Value range</b>	Object attribute in CANopen defining the allowed values that this object supports.

## 1 – Safety summary



### 1.1 Safety

- Always adhere to the professional safety and accident prevention regulations applicable to your country during device installation and operation;
- installation and maintenance operations have to be carried out by qualified personnel only, with power supply disconnected and stationary mechanical parts;
- device must be used only for the purpose appropriate to its design: use for purposes other than those for which it has been designed could result in serious personal and/or the environment damage;
- high current, voltage and moving mechanical parts can cause serious or fatal injury;
- warning! Do not use in explosive or flammable areas;
- failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of design, manufacture, and intended use of the equipment;
- Lika Electronic assumes no liability for the customer's failure to comply with these requirements.



### 1.2 Electrical safety

- Turn off power supply before connecting the device;
- connect according to explanation in the "4 – Electrical connections" section on page 25;
- in compliance with the 2014/30/EU norm on electromagnetic compatibility, following precautions must be taken:
  - before handling and installing, discharge electrical charge from your body and tools which may come in touch with the device;
  - power supply must be stabilized without noise, install EMC filters on device power supply if needed;
  - always use shielded cables (twisted pair cables whenever possible);
  - avoid cables runs longer than necessary;
  - avoid running the signal cable near high voltage power cables;
  - mount the device as far as possible from any capacitive or inductive noise source, shield the device from noise source if needed;
  - to guarantee a correct working of the device, avoid using strong magnets on or near by the unit;
  - minimize noise by connecting the shield and/or the connector housing and/or the frame to ground. Make sure that ground is not affected by noise. The connection point to ground can be situated both on the device side and on user's side. The best solution to minimize the interference must be carried out by the user. Provide the ground connection as close as possible to the encoder. We suggest using the ground screw provided in the cap (use a TCEI M3 x 6 cylindrical head screw with 2 tooth lock washers).





### 1.3 Mechanical safety

- Install the device following strictly the information in the "3 - Mechanical installation" section on page 22;
- mechanical installation has to be carried out with stationary mechanical parts;
- do not disassemble the encoder;
- do not tool the encoder or its shaft;
- delicate electronic equipment: handle with care; do not subject the device and the shaft to knocks or shocks;
- respect the environmental characteristics declared by manufacturer.

## 2 - Identification

The device can be identified through the **order code** and the **serial number** printed on the label applied to its enclosure. Information is listed in the delivery document too. Please always quote the order code and the serial number when reaching Lika Electronic for purchasing spare parts or needing assistance. For any information on the technical characteristics of the product refer to the technical catalogue.



**Warning:** devices having order code ending with "/Sxxx" may have mechanical and electrical characteristics different from standard and be supplied with additional documentation for special connections (Technical info).

## 3 - Mechanical installation



### WARNING

Installation and maintenance operations have to be carried out by qualified personnel only, with power supply disconnected and mechanical parts compulsorily in stop.

### 3.1 Overall dimensions

(values are expressed in mm)

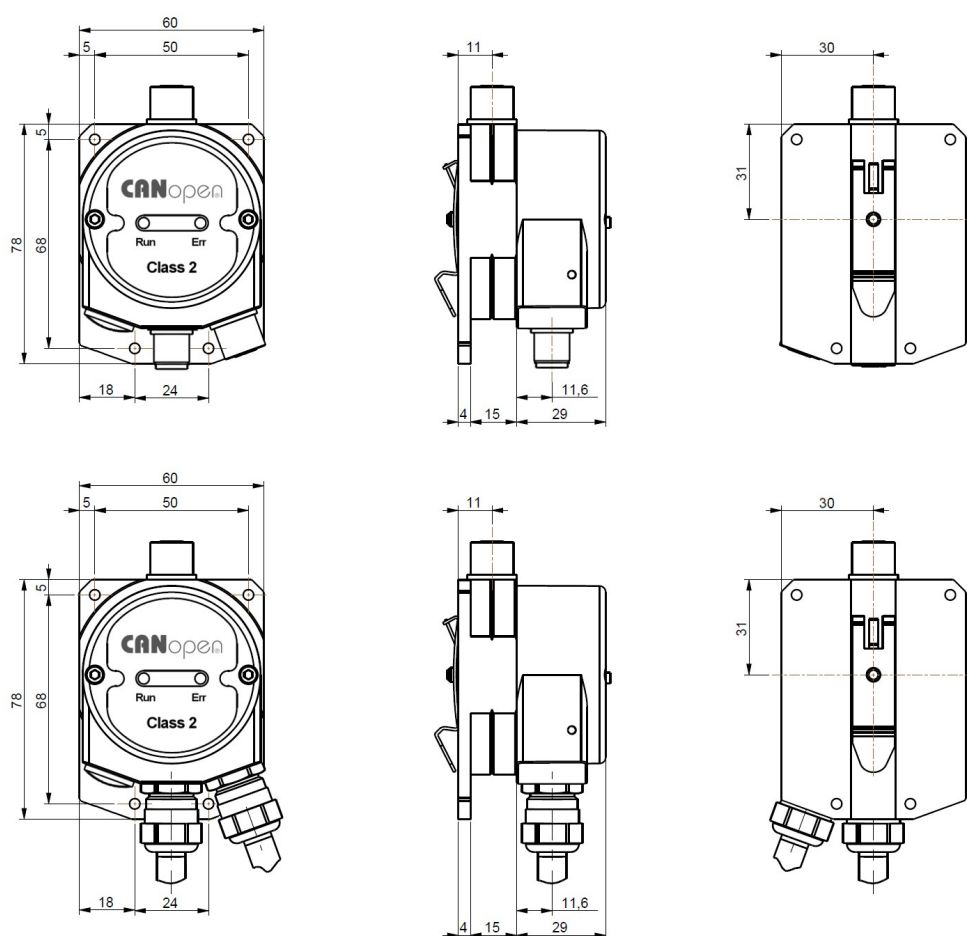


Figure 1

### 3.2 Installation on panel (Figure 2)

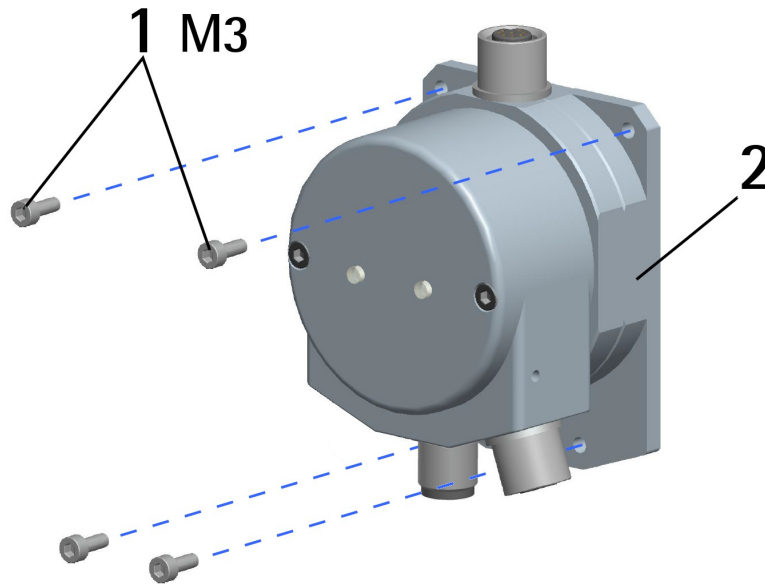


Figure 2

The unit is designed for installation on the even surface of a panel.  
The back flange **2** is fitted with four holes for inserting the fixing screws **1**. Tighten the four fixing screws **1** until the unit is properly fastened to the support. Use **four M3 8 mm min. long cylinder head screws**. The recommended tightening torque is **1.1 Nm**.

### 3.3 Installation with DIN rail clip (Figure 3)

The unit can be installed on DIN profiles inside a rack. A clip **3** for direct fitting on DIN TS35 rails is supplied for free. It has to be fixed on the back of the flange **2** by means of the provided screw **4**.



#### WARNING

To mount the clip **3** you need to remove the cap **5** and drill a hole **A** in the back flange **2**. Delicate electronic circuits and wirings are located inside the cap **5**. Thus this operation has to be accomplished by skilled personnel only. Please pay careful attention and observe great precaution when carrying out this operation.

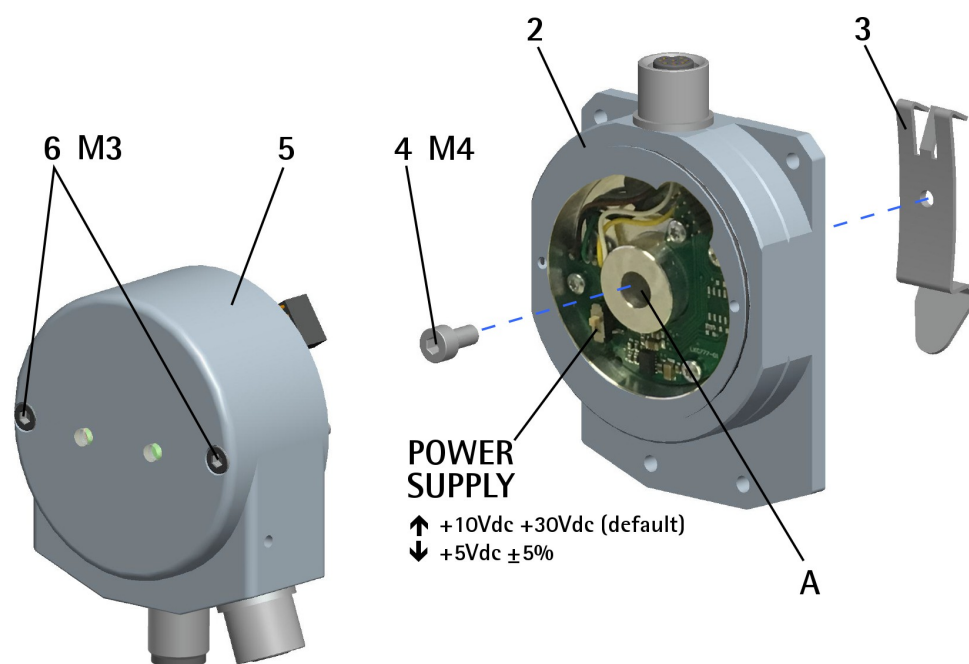


Figure 3

- Loosen the two screws **6** that fasten the cap **5** to the back flange **2**;
- open the cap **5** and separate it from the flange **2**; please pay attention to the internal wirings;
- drill a 4.5 mm diameter hole **A** in the flange **2**; use the notch in the inside of the flange **2** to guide the drill bit;



#### WARNING

Carefully remove the scrap material after drilling.

- mount the clip **3** on the back of the flange **2** and fix it by means of the provided M4 x 8 screw **4**; it has to be screwed on the inner side of the flange **2**;
- replace the cap **5** and fix it by means of the screws **6**.



## 4 - Electrical connections



### WARNING

Power supply must be turned off before performing any electrical connection!

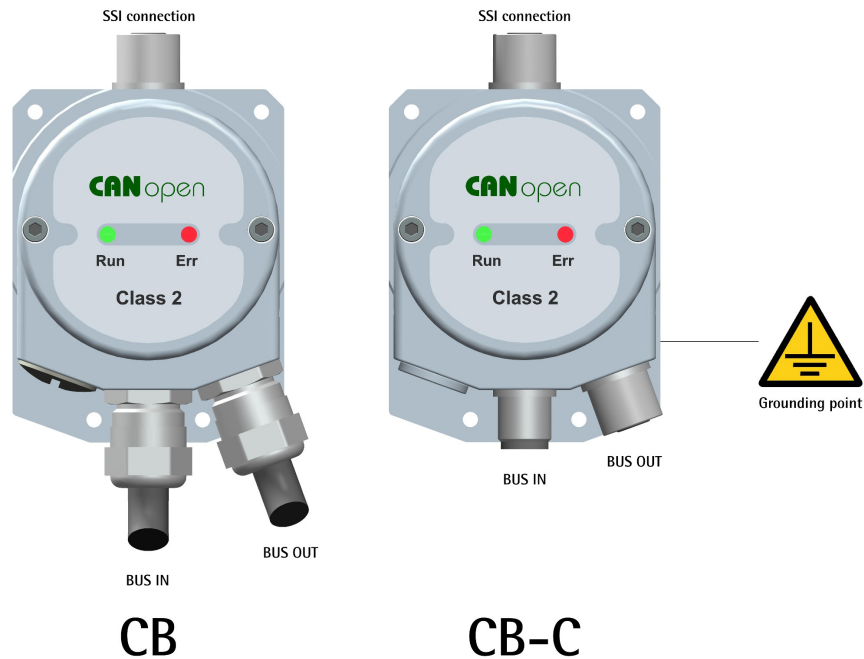


Figure 4

### 4.1 Converter's connection cap (Figure 5)



### WARNING

Do not remove or mount the connection cap with power supply switched ON. Damage may be caused to internal components.

The terminal connectors for connecting the power supply and the BUS IN and BUS OUT cables (CB connection cap) as well as the DIP switches meant to set the power supply to be provided to the connected encoder, thenode ID and the baud rate and activate the termination resistance (CB and CB-C connection caps) are located inside the converter connection cap. Thus you must remove the connection cap to access any of them.

### NOTE

Be careful not to damage the internal components when you perform this operation.

To remove the connection cap loosen the two screws **1** (Figure 5). Please be careful with the internal connector.

Always replace the connection cap at the end of the operation. Take care in re-connecting the internal connector. Tighten the screws **1** using a tightening torque of approx. 2.5 Nm.



### WARNING

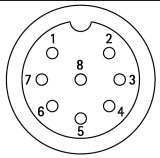
You are required to check that the converter back flange and the connection cap are at the same potential before replacing the connection cap!



Figure 5

## 4.2 SSI connection (Figure 4)

The converter is fitted with one M12 8-pin female connector to network the IF55 gateway and the SSI encoder.

M12 8-pin (frontal side)	SSI connection
	 <p>A coding female</p>

Pin	Description
1	0Vdc power supply voltage
2	+Vdc power supply voltage *
3	Clock IN +
4	Clock IN -
5	Data OUT +
6	Data OUT -
7 and 8	not connected

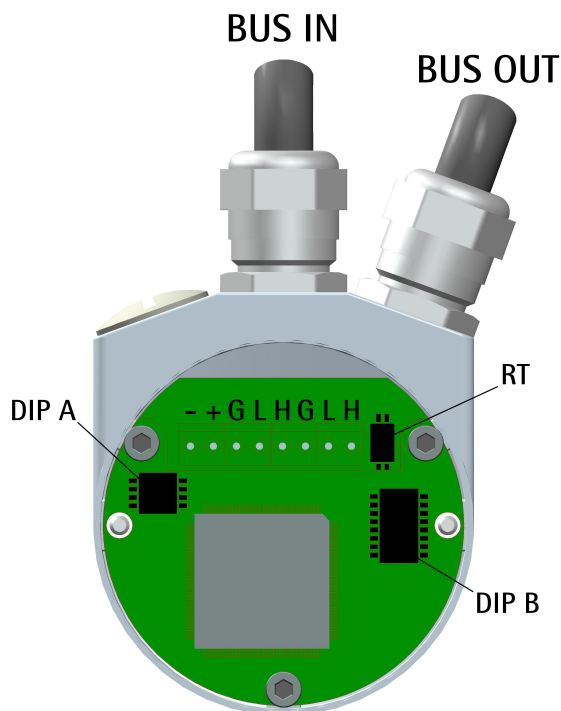
\* The power supply voltage level must be set through the POWER SUPPLY DIP switch located inside the enclosure of the converter, see the "4.7 POWER SUPPLY DIP switch (Figure 8)" section on page 29.



**WARNING**

The max. length of the SSI cable must not exceed 30 m / 98.425 ft.

**4.3 CANopen converter with PGs: CB version (Figure 4 and Figure 6)**



**Figure 6**

The converter is fitted with two PG9 cable glands for BUS IN and BUS OUT connections. The bus cables can be connected directly to the terminal connectors in front of each cable gland. We recommend CANbus certificated cables to be used. Core diameter should not exceed Ø 1.5 mm (0.06").

Terminal connector	Description
-	0Vdc power supply voltage
+	+10Vdc +30Vdc power supply voltage
G	CAN GND <sup>1</sup>
L	CAN Low
H	CAN High
PG	CAN Shield <sup>2</sup>

<sup>1</sup> CAN GND is the 0V reference of CAN signals, it is not connected to 0Vdc supply voltage.

<sup>2</sup> Connect cable shield to cable gland.

#### 4.4 CANopen converter with M12 connectors: CB-C version (Figure 4 and Figure 7)

The converter is fitted with two M12 5-pin connectors with pin-out in compliance with the CANopen® standard. Therefore you can use standard CANopen cables commercially available.

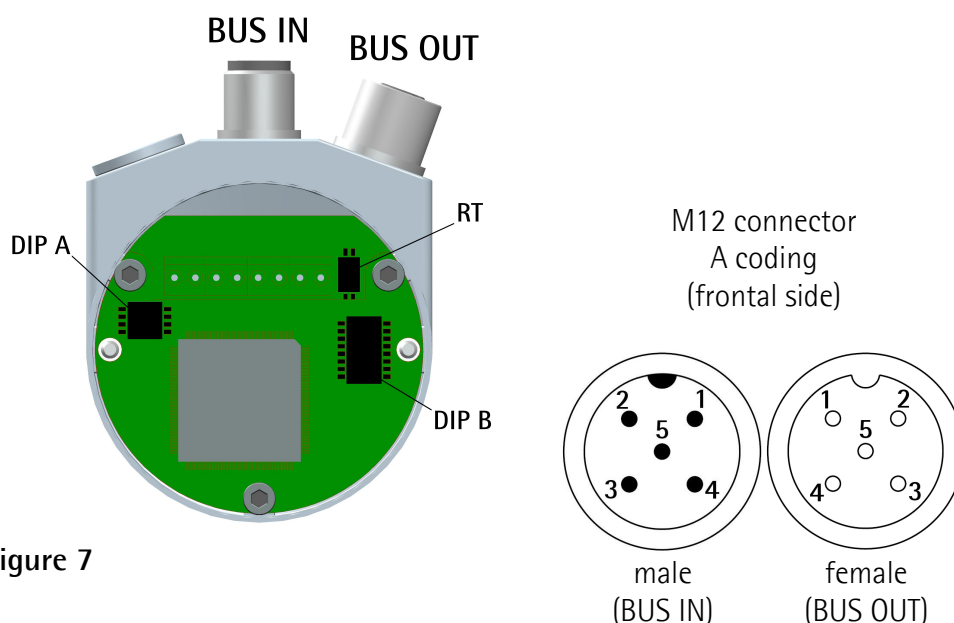


Figure 7

M12	Description
Case	CAN Shield
1 <sup>1</sup>	
2	+10Vdc +30Vdc power supply voltage
3	0Vdc power supply voltage
4	CAN High
5	CAN Low

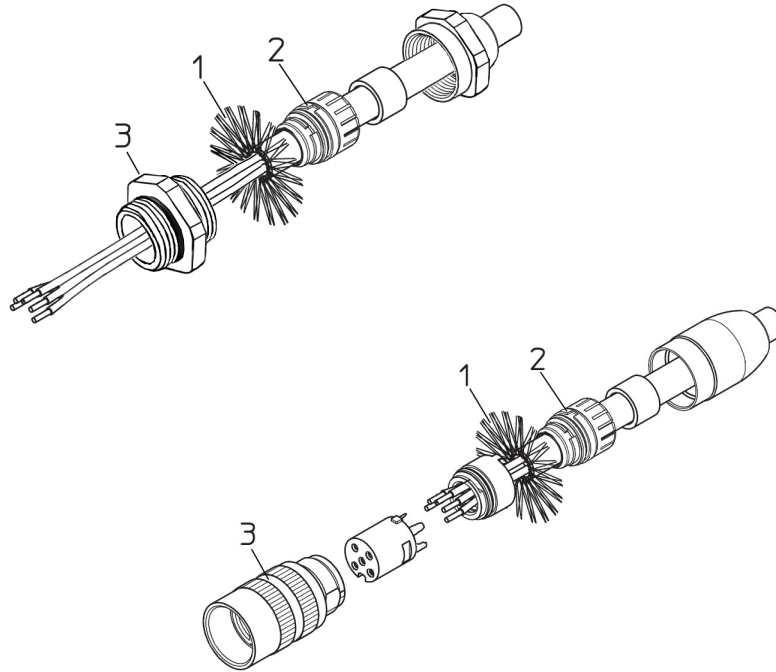
<sup>1</sup> CAN Shield is also connected to pin 1 to allow the connection of the shield even if the plug connector has a plastic case.

#### 4.5 Ground connection (Figure 4)

Minimize noise by connecting the shield and/or the connector housing and/or the enclosure to ground. Make sure that ground is not affected by noise. The connection point to ground can be situated both on the device side and on user's side. The best solution to minimize the interference must be carried out by the user. You are advised to provide the ground connection as close as possible to the unit. We suggest using the ground point provided in the connection cap (see the Figure 4, use one TCEI M3 x 6 cylindrical head screw with two tooth lock washers).

#### 4.6 Shield connection

Disentangle and shorten the shielding **1** and then bend it over the part **2**; finally place the ring nut **3** of the connector. Be sure that the shielding **1** is in tight contact with the ring nut **3**.



#### 4.7 POWER SUPPLY DIP switch (Figure 8)



##### WARNING

Power supply must be turned off before performing this operation!

The power supply voltage level to be provided to the connected encoder must be set through the POWER SUPPLY DIP switch located inside the enclosure of the converter. It must be according to the power supply voltage level required by the connected SSI encoder. To access the POWER SUPPLY DIP switch refer to the "4.1 Converter's connection cap (Figure 5)" section on page 25.

Set the POWER SUPPLY DIP switch to UP position to provide +10Vdc +30Vdc power supply voltage level to the encoder (default setting); set the POWER SUPPLY DIP switch to DOWN position to provide +5Vdc  $\pm 5\%$  power supply voltage level to the encoder.

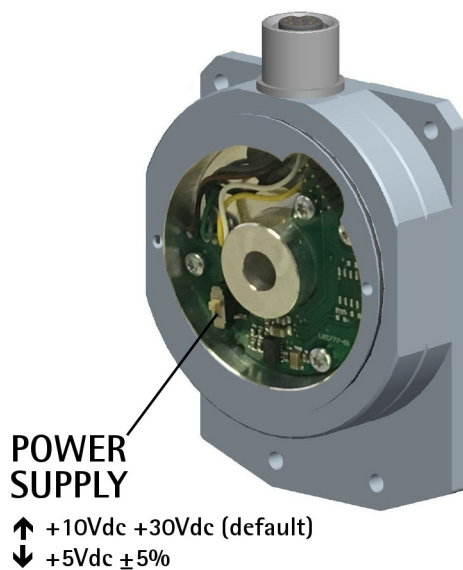


Figure 8

#### 4.8 Baud rate: DIP A (Figure 6 and Figure 7)

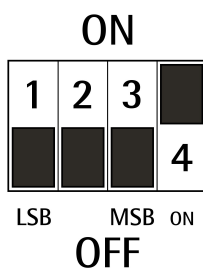


##### WARNING

Power supply must be turned off before performing this operation!

The transmission rate must be set **via hardware** by using the DIP A DIP switch. The bit 4 of DIP A must be always set to "ON".

DIP A:



Switch off the device and set the binary value of the transmission rate considering that: ON = 1, OFF = 0.

bit	1 LSB	2	3 MSB	4
	2 <sup>0</sup>	2 <sup>1</sup>	2 <sup>2</sup>	ON

Available baud rate values:

Decimal value	Binary value	Baud rate
0	000	20 Kbit/s
1	001	50 Kbit/s
2	010	100 Kbit/s
3	011	125 Kbit/s
4	100	250 Kbit/s
5	101	500 Kbit/s (default)
6	110	800 Kbit/s
7	111	1000 Kbit/s

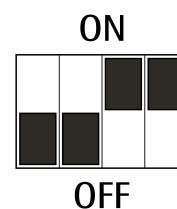


#### EXAMPLE

Set the baud rate to 250Kbit/s:

$4_{10} = 100_2$  (binary value, see the table above)

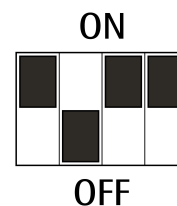
bit	1	2	3	4
	$2^0$	$2^1$	$2^2$	$2^3$
	OFF	OFF	ON	ON



Set the baud rate to 500Kbit/s:

$5_{10} = 101_2$  (binary value, see the table above)

bit	1	2	3	4
	$2^0$	$2^1$	$2^2$	$2^3$
	ON	OFF	ON	ON



#### 4.9 Node number: DIP B (Figure 6 and Figure 7)



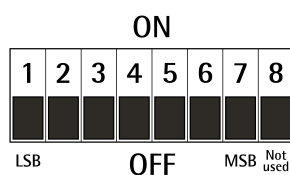
#### WARNING

Power supply must be turned off before performing this operation!

The node number must be set via hardware by using the DIP B DIP switch.

Allowed node addresses are range between 1 and 127. **The default value is 1.**

DIP B:



Switch off the device and set the binary value of the node number considering that: ON = 1, OFF = 0

bit	1 LSB	2	3	4	5	6	7 MSB	8 not used
	$2^0$	$2^1$	$2^2$	$2^3$	$2^4$	$2^5$	$2^6$	

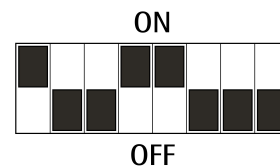


#### EXAMPLE

Set the node number = 25:

$25_{10} = 0001\ 1001_2$  (binary value)

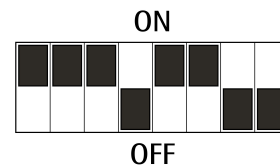
bit	1	2	3	4	5	6	7	8
	$2^0$	$2^1$	$2^2$	$2^3$	$2^4$	$2^5$	$2^6$	
	ON	OFF	OFF	ON	ON	OFF	OFF	OFF



Set the node number = 55:

$55_{10} = 0011\ 0111_2$  (binary value)

bit	1	2	3	4	5	6	7	8
	$2^0$	$2^1$	$2^2$	$2^3$	$2^4$	$2^5$	$2^6$	
	ON	ON	ON	OFF	ON	ON	OFF	OFF



#### 4.10 RT Bus termination (Figure 6 and Figure 7)



#### WARNING

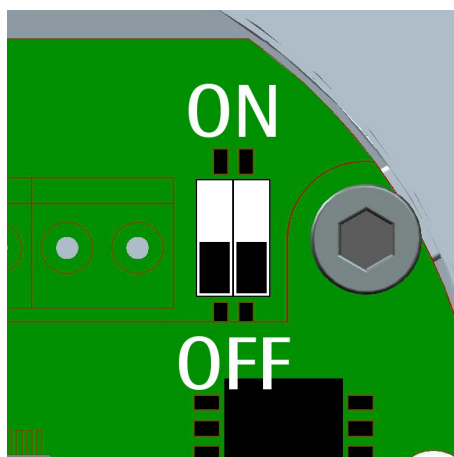
Power supply must be turned off before performing this operation!

A bus termination resistance is provided inside the connection cap and must be activated as line termination if the device is at the ends of the transmission line (i.e. it is either the first or the last device in the transmission line).

Use RT Switch to activate or deactivate the bus termination.

RT	Description
1 = 2 = ON	Activated: if the device is either the first or the last device in the transmission line
1 = 2 = OFF	Deactivated: if the device is neither the first nor the last device in the transmission line





#### 4.11 Diagnostic LEDs (Figure 4)

Two LEDs located in the rear side of the converter are designed to show the operating or fault status of the CANopen® interface.

GREEN LED	Description
ON	The encoder is in <b>Operational</b> state
Single flash	The encoder is in <b>Stopped</b> state
Blinking	The encoder is in <b>Pre-Operational</b> state

RED LED	Description
ON	Bus off, the CAN controller is switched off
Double flash	<b>Node guarding error</b> , see on page 78 ff
Single flash	Max. number of warning errors reached
Blinking	Generic error or <b>Flash memory error</b> , see on page 78 ff
OFF	No error

During initialization, the device carries out a hardware test to check LEDs operation. Both LEDs light up.

## 5 – Getting started



The following instructions allow the operator to quickly and safely set up the converter in a standard operational mode and to execute its main functions. For complete and detailed information please read the mentioned pages thoroughly.

- Mechanically install the device, see on page 22;
- execute the electrical and network connections, see on page 25;
- if required, set the power supply voltage level of the connected encoder, see on page 29;
- set the node address, see on page 31;
- set the transmission rate, see on page 30;
- set the line termination if required, see on page 32;
- switch on the +10Vdc +30Vdc power supply;
- in the software tool install the EDS file, see on page 39;
- set the characteristics of the connected SSI encoder:
  - set the output code used to arrange the output information next to the **2200-01 Code Type (BIN/GRAY)** object;
  - set the protocol used to arrange the absolute information next to the **2200-02 SSI Protocol** object;
  - set the number of SSI clocks next to the **2200-03 Number of SSI clocks** object;
  - set the physical singleturn resolution of the SSI encoder next to the **2200-05 Physical pulse measure [nm]** object; the **6005-01 Measuring step setting** and **6501-00 Measuring step** objects are automatically set accordingly;
  - set the max. number of information the SSI encoder can output for the max. measuring range next to the **2200-04 Physical Total Resolution [bits]** object; the **6001-00 Total measuring range** and **6002-00 Programmable total resolution** objects are automatically set accordingly;
- the **6001-00 Total measuring range / 6002-00 Programmable total resolution** objects are automatically set according to the value in the **2200-04 Physical Total Resolution [bits]** object; the user can set a custom measuring range;
- if you want to use the default resolution (physical resolution, see the **2200-05 Physical pulse measure [nm]** object and the **6501-00 Measuring step** object), please check that the **Scaling function control** parameter is disabled (the bit 2 in the **6000-00 Operating parameters** object = 0; see on page 60);
- otherwise, if you need a custom resolution, enable the **Scaling function control** parameter (the bit 2 in the **6000-00 Operating parameters** object = 1; see on page 60) and then set the resolution you need for

your application next to the **6005-01 Measuring step setting** object, see on page 68;

- now, if you need you can enter the Preset value next to the **6003-00 Preset value** object and then set it in the desired position; see on page 66;
- save the new setting values (**1010-01 Store parameters** object; see on page 48).



#### NOTE

Please consider that if the **2200-06 Bypass** object (see on page 59) is set to "0" = disabled, the position value read by the encoder can be processed according to needs, so the user can scale the value, set a preset, and change the counting direction. On the contrary, if the **2200-06 Bypass** object (see on page 59) is set to "1" = enabled, the information from the encoder is transmitted "as it is" and not processed in any way. The preset, scaling and counting direction functions -even if set and enabled- are ignored; also the output code setting is ignored. If, for example, the user sets a preset while the bypass mode is enabled, the value is accepted, but not activated. As soon as the bypass mode is disabled, the preset, scaling and counting direction functions -if set and enabled- become active and the **6004-00 Position value** will be accordingly.



#### EXAMPLE 1

We need to connect a **SMA5-GA-50** linear encoder.

The main features of the linear encoder are:

Resolution: **0.05 mm** (-50-, see the order code in the product datasheet).

Max. measuring length: **5,050 mm** (see the "Mechanical Specifications" in the product datasheet).

Output code: **Gray code** (-GA-, see the order code in the product datasheet).

SSI protocol: **25-bit "LSB Right Aligned" protocol** (see the User's manual).

**2200-01 Code Type (BIN/GRAY)** = 1h (= Gray code)

**2200-02 SSI Protocol** = 0h (= 25-bit "LSB Right Aligned" protocol)

**2200-03 Number of SSI clocks** = 19h (= 25 dec)

**2200-04 Physical Total Resolution [bits]** = 11h (= Max. measuring length/Resolution = 5,050/0.05 = 101,000  $\approx 2^{17}$  = 17 bits)

**2200-05 Physical pulse measure [nm]** = C350h (0.05 mm resolution = 50,000 nm resolution)

**6001-00 Total measuring range / 6002-00 Programmable total resolution** = 0002 0000h (= 5,050/0.05 = 101,000 information; default and max. value  $2^{17}$  = 131,072 dec = 0002 0000h) as a default; the user can set a custom measuring range

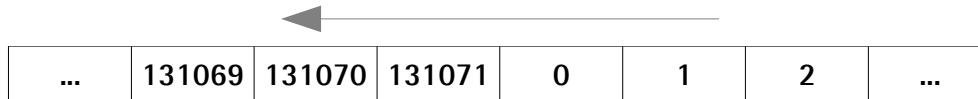
If you want to use the physical resolution:

**Scaling function control** in the **6000-00 Operating parameters** object = 0

If you need a custom resolution:

**Scaling function control** in the **6000-00 Operating parameters** object = 1  
**6005-01 Measuring step setting**  $\geq$  **2200-05 Physical pulse measure [nm]**

If you set a 0 preset along the path, when the encoder moves back and cross the zero, the value immediately after 0 will be  $2^{2200-04 \text{ Physical Total Resolution [bits]}} - 1$ , i.e. 131,071 (assuming that **6001-00 Total measuring range** / **6002-00 Programmable total resolution** = 131,072).



## EXAMPLE 2

We need to connect a **SMAx-BG-100** linear encoder.

The main features of the linear encoder are:

Resolution: **0.1 mm** (-100-, see the order code in the product datasheet).

Max. measuring length: **600 mm** (see the "Mechanical Specifications" in the product datasheet).

Output code: **Binary code** (-BG-, see the order code in the product datasheet).

SSI protocol: **"MSB Left Aligned" protocol** (see the User's manual).

**2200-01 Code Type (BIN/GRAY)** = 0h (= Binary code)

**2200-02 SSI Protocol** = 1h (= "MSB Left Aligned" protocol)

**2200-03 Number of SSI clocks** = 0Dh (= 13 dec), according to **2200-04 Physical Total Resolution [bits]**

**2200-04 Physical Total Resolution [bits]** = 0Dh (= Max. measuring length/Resolution = 600/0.1 = 6,000  $\approx 2^{13}$  = 13 bits)

**2200-05 Physical pulse measure [nm]** = 0001 86A0h (0.1 mm resolution = 100,000 nm resolution)

**6001-00 Total measuring range** / **6002-00 Programmable total resolution** = 0000 2000h (= 600/0.1 = 6,000 information; default and max. value  $2^{13}$  = 8,192 dec = 0000 2000h) as a default; the user can set a custom measuring range

If you want to use the physical resolution:

**Scaling function control** in the **6000-00 Operating parameters** object = 0

If you need a custom resolution:

**Scaling function control** in the **6000-00 Operating parameters** object = 1  
**6005-01 Measuring step setting**  $\geq$  **2200-05 Physical pulse measure [nm]**

If you set a 0 preset along the path, when the encoder moves back and cross the zero, the value immediately after 0 will be  $2^{2200-04 \text{ Physical Total Resolution [bits]}} - 1$ , i.e. 8,191 (assuming that **6001-00 Total measuring range** / **6002-00 Programmable total resolution** = 8,192).

←							
...	8189	8190	8191	0	1	2	...

## 5.1 Quick reference

After setting the SSI encoder parameters and then using the default settings provided by the manufacturer, you can read immediately the position value.

Follow the instructions below to:

- read the value of the physical measuring step: **6501-00 Measuring step**;
- set the desired cycle time **6200-00 Cyclic timer** ≠ 0;
- set the **Operational** mode;
- read the current position (in a cyclic and/or in a synchronous mode).



Default Baud rate and Node-ID are:

**Baud rate = 500 Kbit/s**

**Node-ID = 1**

### Read the value of the measuring step **6501-00 Measuring step**

Master → Slave encoder

COB-ID	Cmd	Index		Sub	Process data			
601	40	01	65	00	-	-	-	-

Slave encoder → Master

COB-ID	Cmd	Index		Sub	Process data			
581	43	01	65	00	88	13	00	00
					Low	...		High

→ 0000 1388h = 5 000 nm = 5 µm (Data bytes as an example)

### Set the cyclic time **6200-00 Cyclic timer** (100 ms = 64h)

Master → Slave encoder (Set request)

COB.ID	Cmd	Index		Sub	Data bytes			
600+ID	2B	00	62	00	64	00	-	-

Slave encoder → Master (Set confirmation)

COB.ID	Cmd	Index		Sub	Data bytes			
580+ID	60	00	62	00	00	00	-	-

## Set the **Operational** mode

Master → Slave encoder

COB-ID	Cmd	Node
000	01	01

## Read the position every 100 ms

Slave encoder → Master

COB-ID	Byte 0	Byte 1	Byte 2	Byte 3
181	05	10	01	00
	Low	...	...	High

Data bytes as an example

To convert the read position value into nanometres [nm] (and into micrometres or millimetres or other engineering unit afterwards) you must multiply the read position by the value set next to the **6501-00 Measuring step** object (if the bit 2 **Scaling function control** in **6000-00 Operating parameters** is disabled = 0); otherwise you must multiply the read position by the value set next to the **6005-01 Measuring step setting** object (if the bit 2 **Scaling function control** in **6000-00 Operating parameters** is enabled = 1).



### EXAMPLE

Read position = 0001 1005h = 69,637 dec.

Position = **6004-00 Position value** \* **6501-00 Measuring step** = 0001 1005h

\* 0000 1388h = 14C0 E1A8h = 348,185,000 nm

348,185,000 nm = 348,185 µm = 348.185 mm

For further information refer to the **6004-00 Position value** object on page 68.



### NOTE

For further examples refer to the "7 – Setting-up" section on page 82.

## 6 - CANopen® interface (DS 406)

CANopen® converters for SSI encoders are always Slave devices and comply with the "Device profile for encoders", Class 2.

For any omitted information concerning the CANopen protocol, refer to the "CiA Draft Standard Proposal 301. Application Layer and Communication Profile" and to the "CiA Draft Standard Proposal 406. Device profile for encoders" documents available at the address [www.can-cia.org](http://www.can-cia.org).

### 6.1 EDS file

CANopen® converters are supplied with their own EDS file **Lika\_IF55\_\_LIN\_DS406\_Vx.eds**, see at [www.lika.biz](http://www.lika.biz) > **DISPLAYS & INTERFACES** > **SIGNAL CONVERTERS & INTERFACES (POSICONTROL)**.

EDS file is available in both English (**Lika\_IF55\_\_LIN\_DS406\_Vx\_en.eds**) and Italian (**Lika\_IF55\_\_LIN\_DS406\_Vx\_it.eds**) versions.

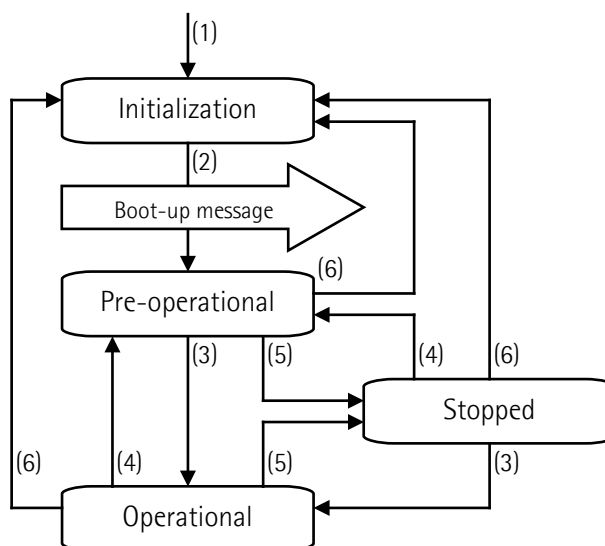
Vx is intended to indicate the file version.

EDS file has to be installed in the CANopen® Master device.

Please note that the rotary encoder converters and the linear encoder converters have different EDS files. Files for rotary encoders are marked with -ROT- in the file name; while files for linear encoders are marked with -LIN- in the file name.

### 6.2 State machine

CANopen® devices are designed to operate using different states. Transition from one state to another is made by sending specific NMT messages (see the Figure below).



(1)	Power on
(2)	Initialization carried out, boot-up message is sent automatically
(3)	NMT message: <b>Start remote node</b>
(4)	NMT message: <b>Enter pre-operational</b>

(5)	NMT message: <b>Stop remote node</b>
(6)	NMT message: <b>Reset node</b> or <b>Reset communication</b> command

### 6.2.1 Initialization state

This is the first state the CANopen® device enters after the power is turned on or after a hardware reset (**Reset node** command). As soon as the basic CANopen® device initialization is carried out, the device reads and loads the parameters saved on EPROM, sends a boot-up message and then switches automatically to **Pre-operational** state.

### 6.2.2 Pre-operational state

In this state the communication between the Master and the Slave is possible using SDO messages. They allow working parameters to be set. The Slave cannot send PDO messages. The state is signalled through the green LED (see on page 33).

To switch the Slave device to **Operational** state the Master must send a **Start remote node** command using an NMT message (see on page 82).

### 6.2.3 Operational state

In this state the Slave device is active and all communication objects are available. The Slave device can use the parameters available in the "Object dictionary" (see on page 45) and is allowed to send process data using PDO messages. The "Object dictionary" can be accessed by using SDO messages. To switch the Slave device back to the **Pre-operational** state the Master must send an **Enter pre-operational** command using an NMT message (see on page 82).

### 6.2.4 Stopped state

In this state the Slave device is forced to cut off the communication with the Master (except the node guarding, if active).

The communication using PDO and SDO messages is not allowed.

To switch the Slave device to either the **Pre-operational** or the **Operational** state the Master must send the specific commands **Enter pre-operational** or **Start remote node** using an NMT message (see on page 82).



#### NOTE

Refer to the "7 – Setting-up" section on page 82 for an example of how the states are to be set.



### 6.3 Communication objects

Four different kinds of communication messages are used in a CANopen® network:

- Network management NMT protocol: NMT protocols are used by the Master to manage the nodes and the network, issue the state machine change commands (i.e. to start and stop the devices), detect the remote device boot-ups and the error conditions.
- Process Data Objects PDO protocol: used to process real time data (transmission of process data in real time).
- Service Data Objects SDO protocol: used to set and read values from the "Object dictionary" of a remote device.
- Special Function Objects:
  - SYNC: synchronization message used by the Master to enable the Slave devices to transmit process data (encoder position information).
  - Emergency: error messages are triggered by each error event.
  - Node Guarding: used to request the state of the Slave; the NMT Master checks the NMT Slaves at regular intervals.

Relation between the device states and the communication objects:

	Initial.	Pre-oper.	Operat.	Stopped
NMT		X	X	X
PDO			X	
SDO		X	X	
Sync			X	
Emerg		X	X	
Boot-up	X			
Node guarding		X	X	X

#### 6.3.1 Pre-defined connection set

Master → Slave broadcast		
Type of COB (Object)	Function code (binary)	COB-ID (hex)
NMT	0000	000
SYNC	0001	080
peer-to-peer object		
EMERGENCY	0001	081 - 0FF
PDO 1 (tx)	0011	181 - 1FF
PDO 2 (tx)	0101	281 - 2FF
PDO 3 (tx)	0111	381 - 3FF
SDO (tx)	1011	581 - 5FF
SDO (rx)	1100	601 - 67F
Nodeguard	1110	701 - 77F
Boot-up	1110	701 - 77F

The type of COB (transmitted/tx or received/rx) is viewed from the Slave device.

## 6.4 NMT objects

Structure of NMT objects:

COB-ID (11 bits)		2 CAN Data Bytes	
Func.code	Node ID	Command	Slave ID
0000	0	NMT Func.	Slave ID

If the Slave ID = 00h, the NMT message is sent to all the nodes in the network.

NMT Function:

Command	NMT Function	State of the node
01 hex	Start remote node	Operational
02 hex	Stop remote node	Stopped
80 hex	Enter pre-operational	Pre-operational
81 hex	Reset node	Pre-operational
82 hex	Reset communication	Pre-operational

## 6.5 Boot-up objects

Structure of the Boot-up message:

COB-ID(hex)	1 CAN Data Byte
700+Node ID	00

## 6.6 PDO objects

PDO (tx) messages are always made up of 4 CAN Data Bytes and are used by the unit to transmit the position value.

Structure of PDO objects:

IDENTIFIER		4 CAN Data Bytes			
COB-ID(hex)		Byte 0	Byte 1	Byte 2	Byte 3
F.C.	Node-ID	$2^7 - 2^0$	$2^{15} - 2^8$	$2^{23} - 2^{16}$	$2^{31} - 2^{24}$
		Low	...	...	High
		Position value (with PDO1, PDO2, PDO3)			

Three types of PDO messages are defined, they are:

### **PDO1 Cyclic mode: cyclic transmission**

The device uses the PDO1 message to transmit the position value **cyclically**, i.e. periodically and independently from the Master.

The interval between two issues is set in the **6200-00 Cyclic timer** object.

To activate (or deactivate) the cyclic mode it is necessary to set to 0 (or 1) the most significant bit of COB-ID used by PDO1 (**1800 TPD01 parameters**, sub 1 object).

#### PD02 and PD03 Sync mode: synchronous transmission

The transmission of the position value is operated by the Master **by sending a SYNC message**.

SYNC message is a high-priority COB transmitted by the Master to request the position value of the Slave through a PDO.

If several nodes (Slave devices) are connected to the network, the Master receives the position values from the Slaves according to the order of the node addresses.

The unit can be programmed to send a reply after a set number of SYNC messages by setting a counter.

The PDO message will be transmitted after having received the set number of SYNC messages.

For PDO2 the value of the counter must be set in the **1801 TPD02 parameters**, sub 2 object.

For PDO3 refer to the **1802 TPD03 parameters**, sub 2 object.

The SYNC transmission mode can be enabled (or disabled) by setting to 0 (or 1) the most significant bit (MSB) of COB-IB used by PDO (**1801 TPD02 parameters** / **1802 TPD03 parameters**, sub1 objects).



#### NOTE

More than one transmission mode can be active at the same time.

### 6.7 SDO objects

SDO messages are used to set and read values from the Object dictionary of the Slave device. These parameters are described in the "Object dictionary" section (see on page 45).

4 bytes at the most are used for CAN data, other 4 bytes are used for Command, Index and Sub-index fields. SDO messages are always followed by confirmation. Thus when the Master sends an SDO message to the Slave, then the Slave always sends a reply (and a warning, should an error occur).

Structure of SDO message:

IDENTIFIER		from 4 to 8 CAN data bytes							
COB-ID(hex)		0	1	2	3	4	5	6	7
F.C.	Node-ID	Com	Index		Sub	Data			
		1 byte	LSB	MSB	1 byte	LSB	...	...	MSB

**Com**            command  
**Index**        parameter index  
**Sub**            parameter sub-index  
**Data**          parameter value

### 6.7.1 Command

The command byte contains the type of telegram transmitted to the CAN network.

Three types of telegram are available:

- Set: it is used to send the configuration parameters to a device;
- Req: it is used by the Master to read data from a Slave device;
- Warnings: they are used by the Slave to send error messages to the Master (e.g. following a wrong SDO message: **Object does not exist in the object dictionary, ...**).

Command	COB	COB type	Data length
22h	Set	M → S request	not spec.
23h	Set	M → S request	4 bytes
2Bh	Set	M → S request	2 bytes
2Fh	Set	M → S request	1 byte
60h	Set	S → M confirmation	0 byte
40h	Req	M → S request	0 byte
42h	Req	S → M reply	not spec.
43h	Req	S → M reply	4 bytes
4Bh	Req	S → M reply	2 bytes
4Fh	Req	S → M reply	1 byte
41h	Req	S → M reply                      segmented SDO	
80h	Error	S → M reply	4 bytes

## 6.8 Object dictionary

The most important part of a device profile is the Object Dictionary. The Object Dictionary is essentially a grouping of objects accessible via the network in an ordered, pre-defined fashion.

The user-related objects are grouped in three main areas: the Communication Profile Area, the Manufacturer Specific Profile Area and the Standardised Device Profile Area. The objects are all described in the EDS file.

The **Communication Profile Area** at indexes from 1000h to 1FFFh contains the communication specific parameters for the CANopen network. These entries are common to all devices. NMT services, PDO objects and SDO objects are described in this section. The Communication Profile Area objects comply with the "CiA Draft Standard Proposal 301 CANopen Application layer and communication profile". Refer to the "6.8.1 Communication Profile Area objects (DS 301)" section on page 47.

The **Manufacturer Specific Profile Area** at indexes from 2000h to 5FFFh is free to add manufacturer-specific functionality. Refer to the "6.8.2 Manufacturer Specific Profile Area objects" section on page 55.

The **Standardised Device Profile Area** at indexes from 6000h to 9FFFh contains all data objects common to a class of devices that can be read or written via the network. The device profiles may use entries from 6000h to 9FFFh to describe the device parameters and the device functionality. The Standardised Device Profile Area objects comply with the "CiA Draft Standard 406 CANopen Device profile for encoders". Refer to the "6.8.3 Standardised Device Profile Area objects (DS 406)" section on page 60.

In the following pages the objects implemented are listed and described as follows:

**Index-subindex** Object name  
[data types, attribute]

- Index and subindex are expressed in hexadecimal notation.
- Attribute:
  - ro = read only access
  - rw = read and write access

Unsigned/Signed8 data type:

Process data bytes							
byte 4							
7	6	5	4	3	2	1	0
MSbit		...				LSbit	

Unsigned/Signed16 data type:

Process data bytes	
byte 4	byte 5
LSByte	MSByte

Unsigned/Signed32 data type:

Process data bytes			
byte 4	byte 5	byte 6	byte 7
LSByte	...	...	MSByte

Unsigned/Signed64 data type:

Process data bytes							
byte 4	byte 5	byte 6	byte 7	byte 8	byte 9	byte 10	byte 11
LSByte	...	...	...	...	...	...	MSByte

## 6.8.1 Communication Profile Area objects (DS 301)

### 1000-00 Device type

[Unsigned32, ro]

It contains information about the device type. The object describes the type of device and its functionality.

Default = 0008 0196h = absolute linear encoder, DS 406)

### 1001-00 Error register

[Unsigned8, ro]

Should an error occur, the bit 0 in this object will be set to "1".

Default = 00h

### 1003 Predefined error field

This object is intended to show the last four errors which caused an emergency message to be triggered. For any information refer to the "6.10 Emergency (EMCY) objects" section on page 78.

- **00 Number of occurred errors** [Unsigned8, rw]  
(write 00h to delete the error list)
- **01 Last error occurred** [Unsigned32, ro]
- **02-05 Previous errors occurred** [Unsigned32, ro]

### 1005-00 COB-ID SYNC message

[Unsigned32, rw]

This object indicates the configured COB-ID of the synchronisation object (SYNC). Further, it defines whether the CANopen device generates the SYNC.

Default = 0000 0080h (CANopen device generates SYNC message)

### 1008-00 Manufacturer Device name

[String, ro]

It shows the name of the device.

Default = "IF55LIN\_CB"

### 1009-00 Manufacturer Hardware version

[String, ro]

It shows the hardware version of the device.

Default = device dependent

### 100A-00 Manufacturer Software version

[String, ro]

It shows the software version of the device.

Default = device dependent

### 100C-00 Guard time

[Unsigned16, rw]

It allows to set the Guard time expressed in milliseconds (msec).

The **100C-00 Guard time** object is used in the "Node guarding protocol" controlled by the Master. For more details see the "6.11 Node guarding protocol" section on page 79.

Default = 0000h

### 100D-00 Life time factor

[Unsigned8, rw]

The **100D-00 Life time factor** object is used in the "Node guarding protocol" controlled by the Master. For more details see the "6.11 Node guarding protocol" section on page 79.

Default = 00h

### 1010-01 Store parameters

[Unsigned32, rw]

This object is used to save all parameters on non-volatile memory.

Write **"save"** (ASCII code in hexadecimal format) in the data bytes:

Master → Slave

COB-ID	Cmd	Index		Sub	Data bytes			
600+ID	23	10	10	01	73	61	76	65
					s	a	v	e

Slave → Master (confirmation)

COB-ID	Cmd	Index		Sub	Data bytes			
580+ID	60	10	10	01	00	00	00	00

### 1011-01 Restore default parameters

[Unsigned32, rw]

This object allows the operator to restore all parameters to default values (default values are set at the factory by Lika Electronic engineers to allow the operator to run the device for standard operation in a safe mode).

Write **"load"** (ASCII code in hexadecimal format) in the data bytes and then issue a **Reset node** command:

Master → Slave

COB-ID	Cmd	Index		Sub	Data bytes			
600+ID	23	11	10	01	6C	6F	61	64
					l	o	a	d

Slave → Master (confirmation)

COB-ID	Cmd	Index		Sub	Data bytes			
580+ID	60	11	10	01	00	00	00	00



Master → Slave (Reset node)

COB-ID	Cmd	Slave ID
000	81	ID

Slave → Master (Boot-up)

COB-ID	Cmd
700+ID	00



#### NOTE

Save the default values after upload using the store parameters function (see the **1010-01 Store parameters** object).

#### 1014-00 COB-ID EMCY

[Unsigned32, rw]

This object defines the COB-ID used to send emergency messages (EMCY).

At power on, this object is forced to the default value.

Default = 0000 0080h+Node-ID

#### 1015-00 Inhibit time EMCY

[Unsigned16, rw]

Inhibit time of the emergency messages (EMCY) expressed in multiples of 100 µs. When set to 0, this function is disabled.

Default = 0000h

#### 1018 Identity object

- **01 Vendor-ID** provided by CIA organization [Unsigned32, ro]  
Default = 0000 012Eh
- **02 Product code** [Unsigned32, ro]  
Default = 0000 000Ah
- **03 Revision number** [Unsigned32, ro]  
Default = 0001 0001h

#### 1800 TPDO1 parameters

PDO1 message is used by default for cyclic transmission of the position value.

For more information refer to the "6.6 PDO objects" section on page 42.

See the **6200-00 Cyclic timer** object to set the cyclic timer.

- **01 COB-ID of PDO1** [Unsigned32, rw]

Bit number	Value	Meaning
31 (MSB)	0	PDO exists / is valid
	1	PDO does not exist / is not valid

<b>30</b>	0	RTR allowed on this PDO ( <b>not implemented</b> )
	1	no RTR allowed on this PDO
<b>29</b>	0	11-bit ID (CAN 2.0A)
	1	29-bit ID (CAN 2.0B)
<b>28 ... 11</b>	0	if bit 29 = 0
	X	if bit 29 = 1: bits 28-11 of 29-bit-COB-ID
<b>10 ... 0 (LSB)</b>	X	bits 10-0 of COB-ID

Default = 4000 0180h+Node-ID (no RTR, COB-ID)



#### WARNING

It is mandatory to set the bit 30 of COB-ID to 1 (value 0 is not allowed). This means that "No RTR is allowed on the PDO".  
At power on, this object is forced to the default value.

- **02 Transmission type** [Unsigned8, rw]

Transmission type	PDO transmission	
<b>00h (0)</b>	Acyclic, synchronous	<b>not implemented</b>
<b>01h ... F0h (1 ... 240)</b>	Cyclic, synchronous	<b>implemented</b>
<b>F1h ... FBh (241 ... 251)</b>	<b>not implemented - reserved</b>	
<b>FCh (252)</b>	Synchronous, RTR only	<b>not implemented</b>
<b>FDh (253)</b>	Asynchronous, RTR only	<b>not implemented</b>
<b>FEh (254)</b>	Asynchronous, manufacturer specific	<b>implemented</b>
<b>FFh (255)</b>	Asynchronous, device profile specific	<b>not implemented</b>

Default = FEh (cyclic transmission, see hereafter and the **6200-00 Cyclic timer** object)



#### WARNING

Following an attempt to set the **Transmission Type** to 0, the value is accepted but the PDO message is not sent; following an attempt to change the **Transmission Type** to any other value that is not supported by the device, an abort message (abort code = 0609 0030h: **Invalid value for parameter**) is generated.

If the value next to the **6200-00 Cyclic timer** object  $\neq$  0, the PDO message is sent cyclically and the interval between two messages is the time set next to the **6200-00 Cyclic timer** object; otherwise, if the value next to the **6200-00 Cyclic timer** object = 0, the PDO message is not sent.



# NOTE

Refer to the "7 – Setting-up" section on page 82 for an example of how the **1800 TPD01 parameters** object is to be set.

## 1801 TPD02 parameters

PD02 message is used by default for synchronous transmission of the position value. For more information refer to the "6.6 PDO objects" section on page 42.

- **01 COB-ID of the PD02** [Unsigned32, rw]

Bit number	Value	Meaning
31 (MSB)	0	PDO exists / is valid
	1	PDO does not exist / is not valid
30	0	RTR allowed on this PDO ( <b>not implemented</b> )
	1	no RTR allowed on this PDO
29	0	11-bit ID (CAN 2.0A)
	1	29-bit ID (CAN 2.0B)
28 ... 11	0	if bit 29 = 0
	X	if bit 29 = 1: bits 28-11 of 29-bit-COB-ID
10 ... 0 (LSB)	X	bits 10-0 of COB-ID

Default = 4000 0280h + Node-ID (no RTR, COB-ID)



# WARNING

It is mandatory to set the bit 30 of COB-ID to 1 (value 0 is not allowed). This means that "No RTR is allowed on the PDO".  
At power on, this object is forced to the default value.

- **02 Transmission type** [Unsigned8, rw]

Transmission type	PDO transmission	
00h (0)	Acyclic, synchronous	<b>not implemented</b>
01h ... F0h (1 ... 240)	Cyclic, synchronous	<b>implemented</b>
F1h ... FBh (241 ... 251)	<b>not implemented - reserved</b>	
FCh (252)	Synchronous, RTR only	<b>not implemented</b>
FDh (253)	Asynchronous, RTR only	<b>not implemented</b>
FEh (254)	Asynchronous, manufacturer specific	<b>implemented</b>
FFh (255)	Asynchronous, device profile specific	<b>not implemented</b>

Default = 01h (synchronous transmission at each SYNC command)  
 The position value is transmitted after the set number of SYNC commands.  
 The interval between the SYNC commands must be set next to this **1801 TPD02 parameters**, sub 2 object.



#### WARNING

Following an attempt to set the **Transmission Type** to 0, the value is accepted but the PDO message is not sent; following an attempt to change the **Transmission Type** to any other value that is not supported by the device, an abort message (abort code = 0609 0030h: **Invalid value for parameter**) is generated.

If the value next to the **6200-00 Cyclic timer** object  $\neq 0$ , the PDO message is sent cyclically and the interval between two messages is the time set next to the **6200-00 Cyclic timer** object; otherwise, if the value next to the **6200-00 Cyclic timer** object = 0, the PDO message is not sent.



#### NOTE

Refer to the "7 – Setting-up" section on page 82 for an example of how the **1801 TPD02 parameters** object is to be set.

#### 1802 TPD03 parameters

PD03 message is used by default for synchronous transmission of the position value. For more information refer to the "6.6 PDO objects" section on page 42.

- **01 COB-ID of the PD03** [Unsigned32, rw]

Bit number	Value	Meaning
<b>31 (MSB)</b>	0	PDO exists / is valid
	1	PDO does not exist / is not valid
<b>30</b>	0	RTR allowed on this PDO ( <b>not implemented</b> )
	1	no RTR allowed on this PDO
<b>29</b>	0	11-bit ID (CAN 2.0A)
	1	29-bit ID (CAN 2.0B)
<b>28 ... 11</b>	0	if bit 29 = 0
	X	if bit 29 = 1: bits 28-11 of 29-bit-COB-ID
<b>10 ... 0 (LSB)</b>	X	bits 10-0 of COB-ID

Default = C000 0380h + Node-ID (disable, no RTR)



### WARNING

It is mandatory to set the bit 30 of COB-ID to 1 (value 0 is not allowed). This means that "No RTR is allowed on the PDO".  
At power on, this object is forced to the default value.

- **02 Transmission type** [Unsigned8, rw]

Transmission type	PDO transmission	
00h (0)	Acyclic, synchronous	not implemented
01h ... F0h (1 ... 240)	Cyclic, synchronous	implemented
F1h ... FBh (241 ... 251)	not implemented - reserved	
FCh (252)	Synchronous, RTR only	not implemented
FDh (253)	Asynchronous, RTR only	not implemented
FEh (254)	Asynchronous, manufacturer specific	implemented
FFh (255)	Asynchronous, device profile specific	not implemented

Default = 01h (synchronous transmission at each SYNC command).  
The position value is transmitted after the set number of SYNC commands.  
The interval between the SYNC commands must be set next to this **1802 TPD03 parameters**, sub 2 object.



### WARNING

Following an attempt to set the **Transmission Type** to 0, the value is accepted but the PDO message is not sent; following an attempt to change the **Transmission Type** to any other value that is not supported by the device, an abort message (abort code = 0609 0030h: **Invalid value for parameter**) is generated.

If the value next to the **6200-00 Cyclic timer** object  $\neq 0$ , the PDO message is sent cyclically and the interval between two messages is the time set next to the **6200-00 Cyclic timer** object; otherwise, if the value next to the **6200-00 Cyclic timer** object = 0, the PDO message is not sent.



### NOTE

Refer to the "7 – Setting-up" section on page 82 for an example of how the **1802 TPD03 parameters** object is to be set.



### NOTE

- The transmission of PDO1, PDO2 and PDO3 messages can be enabled (or disabled) by setting to "0" (or "1") the most significant bit (MSB) used by PDO (object **180xh**, sub1).

- The cyclic transmission or synchronous transmission can be modified by setting the **180xh** sub 2 object. If you need the position value to be transmitted every "n" SYNC commands, you must set the "n" value next to the **180xh** sub 2 object
  - 01h: synchronous transmission at each SYNC command;
  - 02h: synchronous transmission every two SYNC commands;
  - ...
  - FEh: cyclic transmission.
    - If **6200-00 Cyclic timer**  $\neq 0 \rightarrow$  "cyclic transmission": cycle time is set next to the **6200-00 Cyclic timer** object;
    - if **6200-00 Cyclic timer** = 0  $\rightarrow$  the PDO message is not sent.

#### 1A00-01 TPD01 mapping parameter

[Unsigned32, rw]

This object contains the mapping of the PDO the encoder uses to transmit the position value, according to the DS 406 device profile specifications.

This object describes the content of the PDO by its index, sub-index and length.

The length contains the length of the application object expressed in bits.

31	24	23	16	15	8	7	0
Index				Sub-Index		Length	
MSB				LSB			

Default = 6004 0020h = **6004-00 Position value** object, the length is 32 bits.

#### 1A01-01 TPD02 mapping parameter

[Unsigned32, rw]

See the **1A00-01 TPD01 mapping parameter** object.

Default = 6004 0020h = **6004-00 Position value** object, the length is 32 bits.

#### 1A02-01 TPD03 mapping parameter

[Unsigned32, rw]

See the **1A00-01 TPD01 mapping parameter** object.

Default = 6008 0040h = **6008-00 High precision position value** object, the length is 64 bits.

## 6.8.2 Manufacturer Specific Profile Area objects

### 2104-00 Limit switch min

[Unsigned32, rw]

This object is used to set the lowest software limit switch (-) in the travel.

If the encoder position is greater than the value set in this object, then the bit 12 of the **6500-00 Operating status** object is set to "0".

If the encoder position is less than the value set in this object, then the bit 12 of the **6500-00 Operating status** object is set to "1".

To enable this function set the bit 12 **Limit switch min.** of the **6000-00 Operating parameters** object to "1".

Default = 0000 0010h

### 2105-00 Limit switch max

[Unsigned32, rw]

This object is used to set the highest software limit switch (+) in the travel.

If the encoder position is less than the value set in this object, then the bit 13 of the **6500-00 Operating status** object is set to "0".

If the encoder position is greater than the value set in this object, then the bit 13 of the **6500-00 Operating status** object is set to "1".

To enable this function set the bit 13 **Limit switch max.** of the **6000-00 Operating parameters** object to "1".

Default = 0001 FFF0h

### 2200-01 Code Type (BIN/GRAY)

[Unsigned8, rw]

It sets the output code used by the SSI encoder to output the absolute position information. The output code can be Binary (00h) or Gray (01h). For any information on the output code please refer to the "User's manual" of the connected encoder.

Default = 00h



#### EXAMPLE

We need to connect the following linear encoder: **SMA5-GA-50**.

SMA5 encoder uses the Gray code to output the absolute position information. Thus you have to set the value 01h here. For further information refer to the "User's manual".



#### EXAMPLE

We need to connect the following linear encoder: **SMA5-BG-100**.

"BG" in the order code means that "MSB Left Aligned" protocol and Binary code are used to arrange the absolute position information. Thus you have to set the value 00h here. For further information refer to the "User's manual".

## 2200-02 SSI Protocol

[Unsigned8, rw]

It sets the protocol used by the SSI encoder to arrange the absolute position information. The SSI protocol can be the 25-bit "LSB Right Aligned" protocol (00h) or the "MSB Left Aligned" protocol (01h). For any information on the SSI protocol please refer to the "User's manual" of the connected encoder.

Default = 00h



### EXAMPLE

We need to connect the following linear encoder: **SMA5-GA-50**.

SMA5 encoder uses the 25-bit "LSB Right Aligned" protocol to arrange the absolute position information. Thus you have to set the value 00h here. For further information refer to the "User's manual".



### EXAMPLE

We need to connect the following linear encoder: **SMA5-BG-100**.

"BG" in the order code means that "MSB Left Aligned" protocol and Binary code are used to arrange the absolute position information. Thus you have to set the value 01h here. For further information refer to the "User's manual".

## 2200-03 Number of SSI clocks

[Unsigned8, rw]

It sets the number of SSI clocks required by the SSI encoder to send the complete data word. The number of clocks depends on the max. number of information and the type of SSI protocol. The value has to be comprised between 1 and 32. For any information on the SSI clocks required please refer to the "User's manual" of the connected encoder.

Default = 20h



### NOTE

If the **2200-02 SSI Protocol** object is set to 01h = "MSB Left Aligned" protocol, the **2200-03 Number of SSI clocks** must be equal to the number of bits of the **total physical resolution** (**2200-04 Physical Total Resolution [bits]** object).



### EXAMPLE

We need to connect the following linear encoder: **SMA5-GA-50**.

SMA5 encoder always requires 25 clocks (the length of the word is always 25 bits, regardless of the max. number of information to provide). Thus you have to set 19h here. For further information refer to the "User's manual".





#### EXAMPLE

We need to connect the following linear encoder: **SMAX-BG-100**.

The number of clocks depends on the max. number of information (see the example in the following parameter). Let's say the max. number of information is 6,000, thus it requires 13 clocks. You have to set 0Dh here. For further information refer to the "User's manual".

#### 2200-04 Physical Total Resolution [bits]

[Unsigned8, rw]



#### WARNING

This object is active only if the bit 2 **Scaling function control** in the **6000-00 Operating parameters** object is set to "=0"; otherwise it is ignored and the system uses the custom values (**6005-01 Measuring step setting** and **6001-00 Total measuring range**) to calculate the position information.

Furthermore, if the **2200-06 Bypass** parameter (see on page 59) is set to "1" = enabled, the scaling function -even if enabled- is ignored and the position information is outputted as it is.

It sets the max. number of physical information (expressed in bits) the SSI encoder can output for the max. measuring length. The value depends on the encoder resolution and the max. measuring length and has to be comprised between 1 and 30. As soon as you confirm the value, the system automatically sets the default value of the **6001-00 Total measuring range** and **6002-00 Programmable total resolution** objects accordingly. For any information on the max. number of information please refer to the "User's manual" of the connected encoder.

Default = 1Eh



#### EXAMPLE

We need to connect the following linear encoder: **SMA5-GA-50**. Its resolution is 0.05 mm (see the order code).

The max. measuring length of the the SMA5 linear encoder on the MTA5 scale is 5,050 mm.

The max. number of information the encoder can output results from the following calculation:

$$\text{Total Physical Resolution} = \frac{\text{Max. measuring range}}{\text{Resolution}}$$

$$\text{Total Physical Resolution} = \frac{5,050}{0.05} = 101,000$$

Now you have to "round up" the result to the next highest power of 2, that is:  $131,072 = 2^{17}$ . Thus the number of bits is "17". The value to set here is 11h.


**EXAMPLE**

We need to connect the following linear encoder: **SMAX-BG-100**. Its resolution is 0.1 mm (see the order code).

The max. measuring length of the SMAX linear encoder on the MTAX scale is 600 mm.

The max. number of information the encoder can output results from the following calculation:

$$\text{Total Physical Resolution} = \frac{\text{Max. measuring range}}{\text{Resolution}}$$

$$\text{Total Physical Resolution} = \frac{600}{0.1} = \mathbf{6,000}$$

Now you have to "round up" the result to the next highest power of 2, that is:  $8,192 = 2^{13}$ . Thus the number of bits is "13". The value to set here is 0Dh.

**2200-05 Physical pulse measure [nm]**

[Unsigned32, rw]


**WARNING**

This object is active only if the bit 2 **Scaling function control** in the **6000-00 Operating parameters** object is set to "0"; otherwise it is ignored and the system uses the custom values (**6005-01 Measuring step setting** and **6001-00 Total measuring range**) to calculate the position information.

Furthermore, if the **2200-06 Bypass** parameter (see on page 59) is set to "1" = enabled, the scaling function -even if enabled- is ignored and the position information is outputted as it is.

It sets the physical resolution of the linear encoder expressed in nanometres (nm). The value has to be comprised between 1 and 1 000 000 (1 mm). Usually the physical resolution can be read in the order code (see the product datasheet). As soon as the user confirms the value, the system automatically sets the default value of the **6005-01 Measuring step setting** and **6501-00 Measuring step** objects accordingly.

Default = 0000 1388h


**EXAMPLE**

We need to connect the following linear encoder: **SMA5-GA-50**.

As you can see in the product datasheet, "50" in the order code means a 0.05 mm resolution = 50,000 nm resolution. Thus you have to set the value 0000 C350h here. For further information refer also to the "User's manual".

**EXAMPLE**

We need to connect the following linear encoder: **SMAX-BG-100**.

As you can see in the product datasheet, "100" in the order code means a 0.1 mm resolution = 100,000 nm resolution. Thus you have to set the value 0001 86A0h here. For further information refer also to the "User's manual".

**2200-06 Bypass**

[Unsigned8, rw]

If **2200-06 Bypass** = 0 = disabled, the "Bypass mode" is disabled, that is: the position value (refer to the **6004-00 Position value** parameter on page 68) read by the encoder can be processed according to needs, so the user can scale the value, set a preset and change the counting direction.

If **2200-06 Bypass** = 1 = enabled, the "Bypass mode" is enabled, that is: the information from the encoder is transmitted "as it is" and not processed in any way. The preset, scaling and counting direction functions -even if set and enabled- are ignored. If, for example, the user sets a preset while the "Bypass mode" is enabled, the value is accepted, but not activated. As soon as the "Bypass mode" is disabled, the preset, scaling and counting direction functions -if set and enabled- become active and the **6004-00 Position value** will be accordingly.

Default = 00h

**3000-00 Baud rate**

[Unsigned8, rw]

This object is not active and any attempt to write will fail.

The bit rate has to be set via hardware using the DIP A dip switch. For any information refer to the "4.8 Baud rate: DIP A (Figure 6 and Figure 7)" section on page 30.

**3001-00 Node-ID**

[Unsigned8, rw]

This object is not active and any attempt to write will fail.

The node number has to be set via hardware using the DIP B dip switch. For any information refer to the "4.9 Node number: DIP B (Figure 6 and Figure 7)" section on page 31.

## 6.8.3 Standardised Device Profile Area objects (DS 406)

### 6000-00 Operating parameters

[Unsigned16, rw]

Bit	Function	bit = 0	bit = 1
0 - 1	not used		
2	<b>Scaling function control</b>	<b>disabled</b>	enabled
3	<b>Measuring direction</b>	<b>standard</b>	inverted
4 ... 11	not used		
12	<b>Limit switch min.</b>	<b>disabled</b>	enabled
13	<b>Limit switch max.</b>	<b>disabled</b>	enabled
14 - 15	not used		

Default values are highlighted in bold

Default = 0000h



#### NOTE

Refer to the "7 – Setting-up" section on page 82 for an example of how the **6000-00 Operating parameters** object is to be set.

#### Scaling function control

If this option is disabled (bit 2 = 0), the device uses the physical resolution and the max. number of physical information to arrange the absolute position information (see the **2200-05 Physical pulse measure [nm]** and **2200-04 Physical Total Resolution [bits]** objects; the **6005-01 Measuring step setting** and **6501-00 Measuring step** objects are automatically set accordingly); the values set in the **6005-01 Measuring step setting** and **6001-00 Total measuring range** objects are ignored.

On the contrary, if it is enabled (bit 2 = 1), the system will use the values set in the **6005-01 Measuring step setting** and **6001-00 Total measuring range** objects to calculate the position information.

To know whether the **Scaling function control** is currently enabled, you can read the bit 2 **Scaling** of the **6500-00 Operating status** object, see on page 73.



#### WARNING

When you enable the scaling function (**Scaling function control** = 1), please enter scaled values next to the **6005-01 Measuring step setting** and **6001-00 Total measuring range** objects that are consistent with the physical values.



#### WARNING

When you enable the scaling function (**Scaling function control** = 1), a counting error, i.e. a jump in the position count, may occur if the following conditions arise:

- a physical zero setting has been performed in the linear sensor;
- the **6005-01 Measuring step setting** object value is not a multiple of the physical resolution as set next to the **2200-05 Physical pulse measure [nm]** object;
- the measuring range (**6001-00 Total measuring range** and **6002-00 Programmable total resolution** objects) is not a power of 2 submultiple of the maximum measuring range.

If the above described conditions arise, a counting error may occur when the sensor crosses the physical zero point.

If the scaling function is disabled (**Scaling function control** = 0), the transmitted position values are always consistent.

If the scaling function is enabled (**Scaling function control** = 1) yet no physical zero setting has been performed in the linear sensor, the transmitted position values are always consistent.

If the scaling function is enabled (**Scaling function control** = 1), the **6005-01 Measuring step setting** object value is a multiple of the physical resolution and the measuring range (**6001-00 Total measuring range** and **6002-00 Programmable total resolution** objects) is a power of 2 submultiple of the maximum measuring range, the transmitted position values are consistent, regardless of the physical zero setting.

The **scaling conversion constant (k)** has to be as follows:

$$k = \frac{\text{6501-00 Measuring step}}{\text{6005-01 Measuring step setting}} \leq 1$$

The value in the **6005-01 Measuring step setting** object has to be equal to or greater than the value in the **6501-00 Measuring step** object. If you set "0" next to the **6005-01 Measuring step setting** object, the scaling conversion constant k is automatically forced to "1".



#### WARNING

Every time you enable the scaling function and/or change the scaling values (see the **6001-00 Total measuring range**, **6002-00 Programmable total resolution** and **6005-01 Measuring step setting** objects) then you are required to set a new preset value (see the **6003-00 Preset value** object) and finally save the new parameters (see the **1010-01 Store parameters** object).



#### NOTE

Please consider that if the **2200-06 Bypass** object (see on page 59) is set to "1" = enabled, the scaling function -if set differently from default- is ignored.

### Measuring direction

This is intended to set if the count is increasing (count up information) either when the encoder moves in the standard direction (it is indicated in the encoder's manual) or when the encoder moves in reverse of the standard direction. Setting 0 (bit 3 = 0) causes the encoder counting to increment when the encoder moves in the standard direction; setting 1 (bit 3 = 1) causes the encoder counting to increment when the encoder moves in reverse of the standard direction. For any information on the standard and inverted counting direction please refer to the specific manual of the encoder.

To know whether the **Measuring direction** is currently enabled, you can read the bit 3 **Measuring direction** of the **6500-00 Operating status** object, see on page 73.



#### WARNING

Every time you change the **Measuring direction**, then you are required to set a new preset value (see the **6003-00 Preset value** object) and finally save the new parameters (see the **1010-01 Store parameters** object).



#### NOTE

Please consider that if the **2200-06 Bypass** object (see on page 59) is set to "1" = enabled, the counting direction function -if set differently from default- is ignored.

#### Limit switch min.

#### Limit switch max.

They allow to enable / disable the function of the **2104-00 Limit switch min** and **2105-00 Limit switch max** objects. For further information see on page 55.

To know whether the **Limit switch min. / Limit switch max.** is currently enabled, you can read the bit 12 **Limit switch min** and the bit 13 **Limit switch max** of the **6500-00 Operating status** object, see on page 73.

### 6001-00 Total measuring range

[Unsigned32, rw]



#### WARNING

This object is active only if the bit 2 **Scaling function control** in the **6000-00 Operating parameters** object is set to "1"; otherwise it is ignored and the system uses the physical values to calculate the position information. As soon as the user confirms the value in the **2200-04 Physical Total Resolution [bits]** object, the program automatically sets the default value of the **6001-00 Total**

**measuring range** and **6002-00 Programmable total resolution** objects accordingly.

Furthermore, if the **2200-06 Bypass** object (see on page 59) is set to "1" = enabled, the scaling function -even if enabled- is ignored and the position information is outputted as it is.

If the **Scaling function control** is disabled (the bit 2 in the **6000-00 Operating parameters** object is set to "0"), then the **6001-00 Total measuring range / 6002-00 Programmable total resolution** objects =  $2^{2200-04 \text{ Physical Total Resolution [bits]}}$ .

It sets the length of the travel the encoder has to measure. The value is expressed in number of information. It has to be comprised between 1 and  $2^{30} = 1\,073\,741\,824$ .

It can be either the number of information for the max. measuring length (for instance, if the application needs the whole path); or the number of information for just a part of the scale if the application only uses a section of the scale. Thus this value must be lower than or equal to the number of information resulting from the scale max. measuring length ( $2^{2200-04 \text{ Physical Total Resolution [bits]}}$ ).

We suggest setting a value that is a power of 2 submultiple of the maximum measuring range (**2200-04 Physical Total Resolution [bits]**) not to cause a counting error, i.e. a jump in the position count when the sensor crosses the physical zero point (see the WARNING below).

Default = 0002 0000h



#### WARNING

When you enable the scaling function (**Scaling function control** = 1), please enter scaled values next to the **6005-01 Measuring step setting** and **6001-00 Total measuring range** objects that are consistent with the physical values.



#### EXAMPLE

We need to connect the following linear encoder: **SMA5-GA-50**.

As you can see in the product datasheet, "50" in the order code means a **0.05 mm** resolution. Let's say the mechanical travel of our application is the max. measuring length the SMA5 linear encoder is allowed to run on the MTA5 scale, i.e. **5050 mm**. Thus the max. number of information is **101,000 ≈ 17 bits** (for the complete explanation refer to the **2200-04 Physical Total Resolution [bits]** object). After having set the **2200-04 Physical Total Resolution [bits]** object, the system automatically sets the value 0002 0000h =  $131,072 = 2^{17}$  here. If you need a custom measuring length, you need to enable the **Scaling function control** and then set a value lower than  $2^{17} = 131,072$  here.

If you set a preset along the path, when the encoder moves back and cross the zero, the value immediately after 0 will be  $2^{2200-04 \text{ Physical Total Resolution [bits]}} - 1$ , i.e. 131,071.

←							
...	131069	131070	131071	0	1	2	...



#### EXAMPLE

We need to connect the following linear encoder: **SMAX-BG-100**.

As you can see in the product datasheet, "100" in the order code means a **0.1 mm** resolution. Let's say the mechanical travel of our application is the max. measuring length the SMAX linear encoder is allowed to run on the MTAX scale, i.e. **600 mm**. Thus the max. number of information is **6,000 ≈ 13 bits** (for the complete explanation refer to the **2200-04 Physical Total Resolution [bits]** object). After having set the **2200-04 Physical Total Resolution [bits]** object, the system automatically sets the value 0000 2000h = 8,192 =  $2^{13}$ . If you need a custom measuring length, you need to enable the **Scaling function control** and then set a value lower than  $2^{13} = 8,192$  here.

If you set a preset along the path, when the encoder moves back and cross the zero, the value immediately after 0 will be  $2^{2200-04 \text{ Physical Total Resolution [bits]}} - 1$ , i.e. 8,191.

←							
...	8189	8190	8191	0	1	2	...



#### EXAMPLE

We need to connect a SMA5-GA-50, its physical resolution is **0.05 mm**. Let's say the mechanical travel of our application is **1000 mm**. Thus the max. number of information is **20,000 ≈ 15 bits** (for the complete explanation refer to the **2200-04 Physical Total Resolution [bits]** object). Thus you must enable the **Scaling function control** and set here the value 0000 4E20h (instead of the default value 0002 0000h).

In this way you will obtain several 20,000 information sections following each other all along the whole measuring length. The position information will be from 0 to 19,999; then again from 0 to 19,999 and so on.

...	19997	19998	19999	0	1	2	...	19997	19998	19999	0	1	2	...
-----	-------	-------	-------	---	---	---	-----	-------	-------	-------	---	---	---	-----

← max measuring length →



#### WARNING

When you enable the scaling function (**Scaling function control** = 1), a counting error, i.e. a jump in the position count, may occur if the following conditions arise:

- a physical zero setting has been performed in the linear sensor;



- the **6005-01 Measuring step setting** object value is not a multiple of the physical resolution as set next to the **2200-05 Physical pulse measure [nm]** object;
- the measuring range (**6001-00 Total measuring range** and **6002-00 Programmable total resolution** objects) is not a power of 2 submultiple of the maximum measuring range.

If the above described conditions arise, a counting error may occur when the sensor crosses the physical zero point.

If the scaling function is disabled (**Scaling function control** = 0), the transmitted position values are always consistent.

If the scaling function is enabled (**Scaling function control** = 1) yet no physical zero setting has been performed in the linear sensor, the transmitted position values are always consistent.

If the scaling function is enabled (**Scaling function control** = 1), the **6005-01 Measuring step setting** object value is a multiple of the physical resolution and the measuring range (**6001-00 Total measuring range** and **6002-00 Programmable total resolution** objects) is a power of 2 submultiple of the maximum measuring range, the transmitted position values are consistent, regardless of the physical zero setting.



#### WARNING

When you change the value next to **6001-00 Total measuring range** or **6002-00 Programmable total resolution** objects, then you must check the value in the **6003-00 Preset value** object and perform the preset operation.



#### NOTE

The **6001-00 Total measuring range** and **6002-00 Programmable total resolution** objects are exactly the same thing. When you change the value next to this **6001-00 Total measuring range** object you also change the value next to the **6002-00 Programmable total resolution** object; and vice versa.

#### 6002-00 Programmable total resolution

[Unsigned32, rw]



#### WARNING

This object is active only if the bit 2 **Scaling function control** in the **6000-00 Operating parameters** object is set to "1"; otherwise it is ignored and the system uses the physical values to calculate the position information. As soon as the user confirms the value in the **2200-04 Physical Total Resolution [bits]** object, the program automatically sets the default value of the **6001-00 Total measuring range** and **6002-00 Programmable total resolution** objects accordingly.

Furthermore, if the **2200-06 Bypass** object (see on page 59) is set to "1" = enabled, the scaling function -even if enabled- is ignored and the position information is outputted as it is.

This object is exactly the same as the **6001-00 Total measuring range** object. When you change the value next to this **6002-00 Programmable total resolution** object you also change the value next to the **6001-00 Total measuring range** object; and vice versa. For any information refer to the **6001-00 Total measuring range** object.

Default = 0002 0000h

### 6003-00 Preset value

[Unsigned32, rw]

This object allows to set the encoder position to a Preset value. The Preset function is meant to assign a desired value to a physical position of the encoder along the scale. The chosen physical position will get the value set next to this object and all the previous and following positions will get a value according to it. This function is useful, for example, when the zero position of the encoder and the zero position of the axis need to match. The preset value will be set for the position of the encoder in the moment when the preset value is sent. We suggest setting the preset value when the encoder is in stop.

Default = 0000 0000h



### EXAMPLE

Let's take a look at the following example to better understand the preset function and the meaning and use of the related objects and commands: **6003-00 Preset value** and **6509-00 Offset value**.

The encoder position which is transmitted results from the following calculation:

**Transmitted value** = **read position** (it does not matter whether the position is physical or scaled) + **6003-00 Preset value** - **6509-00 Offset value**.

If you never set the **6003-00 Preset value** and you never performed the preset setting, then the transmitted value and the read position are necessarily the same as **6003-00 Preset value** = 0 and **6509-00 Offset value** = 0.

When you set the **6003-00 Preset value** and then execute the preset setting, the system saves the current encoder position in the **6509-00 Offset value** object. It follows that the transmitted value and the **6003-00 Preset value** are the same as **read position** - **6509-00 Offset value** = 0; in other words, the value set next to the **6003-00 Preset value** object is paired with the current position of the encoder as you wish.

For example, let's assume that the value "50" is set next to the **6003-00 Preset value** object and you execute the preset setting when the encoder position is "1000". In other words, you want to receive the value "50" when the encoder reaches the position "1000".

We will obtain the following information sequence:

**Transmitted value** = **read position** (= "1000") + **6003-00 Preset value** (= "50") - **6509-00 Offset value** (= "1000") = 50.

The following transmitted value will be:

**Transmitted value** = **read position** (= "1001") + **6003-00 Preset value** (= "50") - **6509-00 Offset value** (= "1000") = 51.

And so on.

To set the preset value you must send the following command:

Set Preset value **6003-00 Preset value** (preset = 1000 = 0000 03E8h)

Master → Encoder (Set request)

COB-ID	Cmd	Index	Sub	Process data			
600+ID	23	03	60	00	E8	03	00

Encoder → Master (Set confirmation)

COB-ID	Cmd	Index	Sub	Process data			
580+ID	60	03	60	00	00	00	00



#### NOTE

- If the scaling function is disabled (bit 2 **Scaling function control** in the **6000-00 Operating parameters** object = 0), then **6003-00 Preset value** must be less than or equal to **2200-04 Physical Total Resolution [bits]** - 1 (for instance: **2200-04 Physical Total Resolution [bits]** = 13 bits;  $2^{13} - 1 = 8191$ ).
- If the scaling function is enabled (bit 2 **Scaling function control** in the **6000-00 Operating parameters** object = 1), then **6003-00 Preset value** must be less than or equal to **6001-00 Total measuring range** / **6002-00 Programmable total resolution** - 1.



#### WARNING

Check the value in the **6003-00 Preset value** object and perform the preset operation every time you set a new **Measuring direction** or change the value next to **6001-00 Total measuring range**, **6002-00 Programmable total resolution** or **6005-01 Measuring step setting** objects.



#### NOTE

Please consider that if the **2200-06 Bypass** object (see on page 59) is set to "1" = enabled, the preset function -even if set and activated- is ignored. If the user sets a preset while the "Bypass mode" is enabled, the operation is not carried out.



#### NOTE

Please refer to the "7 - Setting-up" section on page 82 for an example of how the **6003-00 Preset value** is to be set.

### 6004-00 Position value

[Unsigned32, ro]

This object contains the current position value of the encoder.

The output value is scaled according to the scaling parameters (if the scaling function is enabled), see the bit 2 **Scaling function control** of the **6000-00 Operating parameters** object.

The position value is transmitted cyclically or synchronously according to the settings in the **1800 TPDO1 parameters** and **1801 TPDO2 parameters** objects (see on page 49). See also the **6008-00 High precision position value** object.

To convert the read position value into nanometres [nm] (and into micrometres or millimetres or other engineering unit afterwards) you must multiply the read position by the value set next to the **6501-00 Measuring step** object (if the bit 2 **Scaling function control** in the **6000-00 Operating parameters** object is disabled = 0); otherwise you must multiply the read position by the value set next to the **6005-01 Measuring step setting** object (if the bit 2 **Scaling function control** in **6000-00 Operating parameters** is enabled = 1).



#### EXAMPLE

We have the following linear encoder: **SMA5-GA-50**.

**Scaling function control** = 0

**6501-00 Measuring step** = 0000 C350h = 50,000 nm = 0.05 mm

**6004-00 Position value** = 0001 1005h = 69,637 dec

Position = **6004-00 Position value** \* **6501-00 Measuring step** = 0001 1005h

\* 0000 C350h = CF88 D090h = 3,481,850,000 nm

3,481,850,000 nm = 3,481,850 µm = 3,481.85 mm



#### EXAMPLE

We have the following linear encoder: **SMA5-GA-50**.

**Scaling function control** = 1

**6005-01 Measuring step setting** = 0001 86A0h = 100,000 nm = 0.1 mm

**6004-00 Position value** = 0000 1760h = 5,984 dec

Position = **6004-00 Position value** \* **6005-01 Measuring step setting** =

0000 1760h \* 0001 86A0h = 23AA DC00h = 598,400,000 nm

598,400,000 nm = 598,400 µm = 598.4 mm

### 6005-01 Measuring step setting

[Unsigned32, rw]



#### WARNING

This object is active only if the bit 2 **Scaling function control** in the **6000-00 Operating parameters** object is set to "1"; otherwise it is ignored and the system uses the physical resolution, see the **2200-05 Physical pulse measure [nm]** and **6501-00 Measuring step** objects. As soon as the user confirms the value in the **2200-05 Physical pulse measure [nm]** object, the program

automatically sets the default value of the **6005-01 Measuring step setting** and **6501-00 Measuring step** objects accordingly.

Furthermore, if the **2200-06 Bypass** object (see on page 59) is set to "1" = enabled, the scaling function -even if enabled- is ignored and the position information is outputted as it is.

If the bit 2 **Scaling function control** in the **6000-00 Operating parameters** object is set to "=0", then **6501-00 Measuring step** = **2200-05 Physical pulse measure [nm]**.

This object is used to set a custom resolution (otherwise referred to as measuring step) expressed in nanometres [nm].

The resolution can be defined as the smallest change in the underlying quantity that produces a response in the measurement, the response being the information that is provided to output.

The custom resolution value must be greater than or equal to the physical resolution of the connected encoder.

We suggest setting a value that is multiple of the physical resolution as set next to the **2200-05 Physical pulse measure [nm]** object not to cause a counting error, i.e. a jump in the position count when the sensor crosses the physical zero point (see the WARNING below).

Default = according to **2200-05 Physical pulse measure [nm]**



#### EXAMPLE

We need to connect the following linear encoder: **SMA5-GA-50**.

As you can see in the product datasheet, "50" in the order code means a **0.05 mm resolution** = 50,000 nanometres resolution. As soon as the user confirms the value in the **2200-05 Physical pulse measure [nm]** object, the system automatically sets the default value of the **6005-01 Measuring step setting** object accordingly (0000 C350h). If needed, after enabling the **Scaling function control** parameter the user is allowed to set a custom resolution: it must be greater than or equal to 0000 C350h.



#### EXAMPLE

We need to connect the following linear encoder: **SMAX-BG-100**.

As you can see in the product datasheet, "100" in the order code means a **0.1 mm resolution** = 100,000 nanometres resolution. As soon as the user confirms the value in the **2200-05 Physical pulse measure [nm]** object, the system automatically sets the default value of the **6005-01 Measuring step setting** object accordingly (0001 86A0h). If needed, after enabling the **Scaling function control** parameter the user is allowed to set a custom resolution: it must be greater than or equal to 0001 86A0h.



### WARNING

When you enable the scaling function (**Scaling function control** = 1), a counting error, i.e. a jump in the position count, may occur if the following conditions arise:

- a physical zero setting has been performed in the linear sensor;
- the **6005-01 Measuring step setting** object value is not a multiple of the physical resolution as set next to the **2200-05 Physical pulse measure [nm]** object;
- the measuring range (**6001-00 Total measuring range** and **6002-00 Programmable total resolution** objects) is not a power of 2 submultiple of the maximum measuring range.

If the above described conditions arise, a counting error may occur when the sensor crosses the physical zero point.

If the scaling function is disabled (**Scaling function control** = 0), the transmitted position values are always consistent.

If the scaling function is enabled (**Scaling function control** = 1) yet no physical zero setting has been performed in the linear sensor, the transmitted position values are always consistent.

If the scaling function is enabled (**Scaling function control** = 1), the **6005-01 Measuring step setting** object value is a multiple of the physical resolution and the measuring range (**6001-00 Total measuring range** and **6002-00 Programmable total resolution** objects) is a power of 2 submultiple of the maximum measuring range, the transmitted position values are consistent, regardless of the physical zero setting.



### NOTE

If you have set and activated the preset, when you change the value next to the **6005-01 Measuring step setting** object, then you must check the value in the **6003-00 Preset value** object and perform the homing operation.



### EXAMPLE

The main and default features of the **SMAX-BG-100** linear encoder are as follows:

- |                                     |                       |
|-------------------------------------|-----------------------|
| • <b>Default resolution</b>         | = 0.1 mm = 100,000 nm |
| • <b>MTAX max. measuring length</b> | = 600 mm              |
| • <b>Max. number of information</b> | = 6,000 (13 bits)     |

As stated, the max. number of information provided to output is calculated as follows:

$$\text{Total Physical Resolution} = \frac{\text{Max. measuring length}}{\text{Resolution}}$$

Thus, in a default configuration the number of information is:

$$\text{Total Physical Resolution} = \frac{\text{Max. measuring length}}{\text{Resolution}} = \frac{600}{0.1} = 6000$$

Let's assume that you need **2000 information** to be provided to output for the max. measuring length. It follows that you need to calculate and then set a custom resolution.

The resolution value results from the following calculation:

$$\text{Resolution} = \frac{\text{Max. measuring length}}{\text{Number of information}}$$

Thus, in the example the resolution will be:

$$\text{Resolution} = \frac{\text{Max. measuring length}}{\text{Number of information}} = \frac{600}{2000} = 0.3$$

As the value next to the **6005-01 Measuring step setting** object has to be expressed in nanometres, then you have to enter the value **300,000**.

The complete programming sequence will be:

1. Enable the **Scaling function control**: **6000-00 Operating parameters**, bit 2 = 1
2. Set the custom resolution: **6005-01 Measuring step setting** = 0004 93E0 hex (300,000 dec)
3. Set the custom number of information: **6001-00 Total measuring range** = 0000 07D0 hex (2,000 dec)
4. Save the set parameters (**1010-01 Store parameters** object; see on page 48)



#### NOTE

Please note that, if you set a preset along the path, when the encoder moves back and cross the zero, the value immediately after 0 will be 1,999 as shown below.

←										
...	1996	1997	1998	1999	0	1	2	3	4	...



**NOTE**

Refer to the "7 – Setting-up" section on page 82 for an example of setting the **6005-01 Measuring step setting** object.

### 6008-00 High precision position value

[Unsigned64, ro]

This object is the same as the **6004-00 Position value** object, yet it is 64-bit long. This object contains the position value. The value is transmitted cyclically or synchronously according to the settings in the **1800 TPDO1 parameters** and **1802 TPDO3 parameters** object (see on page 49 ff). See also the **6004-00 Position value** object.

### 6200-00 Cyclic timer

[Unsigned16, rw]

The cyclic timer value is used in asynchronous transmission mode (**Transmission Type** = FEh) to set the interval between two PDO issues.

If the value next to the **6200-00 Cyclic timer** object  $\neq 0$ , the PDO message is sent cyclically and the interval between two messages is the time set next to the **6200-00 Cyclic timer** object; otherwise, if the value next to the **6200-00 Cyclic timer** object = 0, the PDO message is not sent.

Value is expressed in milliseconds. See on pages 42 and 49 ff.

Default = 0000h

**Setting the cyclic time: 6200-00 Cyclic timer (100 ms = 64h)**

Master → Encoder

COB-ID	Cmd	Index		Sub	Process data			
600+ID	2B	00	62	00	64	00	-	-

Encoder → Master

COB-ID	Cmd	Index		Sub	Process data			
580+ID	60	00	62	00	00	00	-	-



**NOTE**

Refer to the "7 – Setting-up" section on page 82 for an example of how the **6200-00 Cyclic timer** object is to be set.



## 6500-00 Operating status

[Unsigned16, ro]

Bit	Function	bit = 0	bit = 1
0 - 1	not used		
2	Scaling	Disabled	Enabled
3	Measuring direction	Standard	Inverted
4 ... 11	not used		
12	Limit switch min	position > 2104-00 Limit switch min	position < 2104-00 Limit switch min
13	Limit switch max	position < 2105-00 Limit switch max	position > 2105-00 Limit switch max
14	not used		
15	Current operating state	Stopped / Pre- operational	Operational

### Scaling

It shows the value that is currently set through the bit 2 **Scaling function control** in the **6000-00 Operating parameters** object. In other words, it is intended to show whether the scaling function is enabled or disabled. If the value is "0" the scaling function is disabled; if the value is "1" instead the scaling function is enabled. For any further information on setting and using the scaling function refer to the **6000-00 Operating parameters** object on page 60.

### Measuring direction

It shows the value that is currently set through the bit 3 **Measuring direction** in the **6000-00 Operating parameters** object. If the bit is "0" the output encoder position value has been set to increment when the sensor moves in the standard direction; if the bit is "1" instead the output encoder position value has been set to increment when the sensor moves in reverse of the standard direction. For any further information on setting and using the counting direction function refer to the **6000-00 Operating parameters** object on page 60.

### Limit switch min

If the encoder position is greater than the value set in the **2104-00 Limit switch min** object, the bit 12 of this object is set to "0".

If the encoder position is less than the value set in the **2104-00 Limit switch min** object, the bit 12 of this object is set to "1".

To enable this function set the bit 12 **Limit switch min.** in the **6000-00 Operating parameters** object to "1".

#### Limit switch max

If the encoder position is less than the value set in the **2105-00 Limit switch max** object, the bit 13 of this object is set to "0".

If the encoder position is greater than the value set in the **2105-00 Limit switch max** object, the bit 13 of this object is set to "1".

To enable this function set the bit 13 **Limit switch max.** in the **6000-00 Operating parameters** object to "1".

#### Current operating state

It shows the current operating state of the unit. For further information on the available states see the "6.2 State machine" section on page 39.

bit 15 = 0: **Stopped** or **Pre-operational** state;

bit 15 = 1: **Operational** state.

#### 6501-00 Measuring step

[Unsigned32, ro]



#### WARNING

This object is active only if the bit 2 **Scaling function control** in the **6000-00 Operating parameters** object is set to "0"; otherwise it is ignored and the system uses the custom values (**6005-01 Measuring step setting** and **6001-00 Total measuring range**) to calculate the position information.

Furthermore, if the **2200-06 Bypass** parameter (see on page 59) is set to "1" = enabled, the scaling function -even if enabled- is ignored and the position information is outputted as it is.

This object is intended to show the physical resolution of the connected encoder expressed in nanometres [nm]. The physical resolution must be set next to the **2200-05 Physical pulse measure [nm]** object. As soon as the user confirms the value in the **2200-05 Physical pulse measure [nm]** object, the program automatically sets the value in this object accordingly. If you want to set a custom resolution see the **6005-01 Measuring step setting** object.

Default = according to **2200-05 Physical pulse measure [nm]**

#### 6502-00 Number of distinguishable revolutions

[Unsigned16, ro]

This object is not used in this application, it is listed for full compliance with the CANopen specifications.

Default = 0001h

### 6504-00 Supported alarms

[Unsigned16, ro]

This object contains the information on the alarms supported by the encoder.

No alarms are supported in this encoder.

Default = 0000h (no alarms supported)

### 6506-00 Supported warnings

[Unsigned16, ro]

This object contains the information on the warnings supported by the encoder.

No warnings are supported in this encoder.

Default = 0000h (no warnings supported)

### 6507-00 Profile and software version

[Unsigned32, ro]

It shows the version of both the profile and the software.

Version of the profile for encoders = 3.1

Software version = 1.1

Default = 0301 0101h

### 6508-00 Operating time

[Unsigned32, ro]

This object contains the information on the operating time. The operating time monitor stores the operating time for the encoder expressed in operating hours. The operating time is stored in the encoder non-volatile memory as long as the encoder is power supplied.

This object is currently not used in this encoder.

Default = FFFF FFFFh (not used)

### 6509-00 Offset value

[Integer32, ro]

As soon as you activate the preset, the current position of the encoder is saved in this object. The offset value is then used in the preset function in order to calculate the encoder position value to be transmitted. To zero set the value in this object you must upload the factory default values (see the **1011-01 Restore default parameters** object on page 48).

For any further information on the preset function and the meaning and use of the related objects and commands **6003-00 Preset value** and **6509-00 Offset value** refer to page 66.

Default = 0000 0000h

### 650A-01 Manufacturer offset value

[Integer32, ro]

This object contains the manufacturer-specific offset value. This is the difference between the physical zero position of the encoder (zero set mechanically) and the zero position set by the manufacturer (zero set via software).

Default = 0000 0000h (not used)

### 650B-00 Serial number

[Unsigned32, ro]

This object contains the serial number of the converter.

This object is currently not used in this converter.

Default = FFFF FFFFh (not used)



#### NOTE

Save the new values using the store parameters function (see the [1010-01 Store parameters](#) object), otherwise should the power be turned off or **Reset node** or **Reset communication** commands be sent all data not saved will be lost!

## 6.9 SDO abort codes

Here follows the list and meaning of the SDO abort codes indicated by CANopen but not necessarily supported by the manufacturer. For complete information please refer to the "SDO abort transfer protocol" section in the "CiA Draft Standard 301" document available at the address [www.can-cia.org](http://www.can-cia.org).

Abort code	Description
0503 0000h	Toggle bit not alternated.
0504 0000h	SDO protocol timed out.
0504 0001h	Client/server command specifier not valid or unknown.
0504 0002h	Invalid block size (block mode only).
0504 0003h	Invalid sequence number (block mode only).
0504 0004h	CRC error (block mode only).
0504 0005h	Out of memory.
0601 0000h	Unsupported access to an object.
0601 0001h	Attempt to read a write only object.
0601 0002h	Attempt to write a read only object.
0602 0000h	Object does not exist in the object dictionary.
0604 0041h	Object cannot be mapped to the PDO.
0604 0042h	The number and length of the objects to be mapped would exceed PDO length.
0604 0043h	General parameter incompatibility reason.
0604 0047h	General internal incompatibility in the device.
0606 0000h	Access failed due to an hardware error.
0607 0010h	Data type does not match, length of service parameter does not match
0607 0012h	Data type does not match, length of service parameter too high
0607 0013h	Data type does not match, length of service parameter too low
0609 0011h	Sub-index does not exist.
0609 0030h	Invalid value for parameter (download only).
0609 0031h	Value of parameter written too high (download only).
0609 0032h	Value of parameter written too low (download only).
0609 0036h	Maximum value is less than minimum value.
060A 0023h	Resource not available: SDO connection
0800 0000h	General error
0800 0020h	Data cannot be transferred or stored to the application.
0800 0021h	Data cannot be transferred or stored to the application because of local control.
0800 0022h	Data cannot be transferred or stored to the application because of the present device state.
0800 0023h	Object dictionary dynamic generation fails or no object dictionary is present (e.g. object dictionary is generated from file and generation fails because of an file error).

0800 0024h	No data available
------------	-------------------

## 6.10 Emergency (EMCY) objects

Emergency objects (EMCY) are triggered by the device when an internal error occurs.

EMCY structure:

IDENTIFIER	CAN Data Byte			
COB-ID(hex)	0	1	2	3 ... 7
see the <b>1014-00 COB-ID EMCY</b> object	Error code		Sub of error register	Specific code
	LSB	MSB	01	00 ... 00

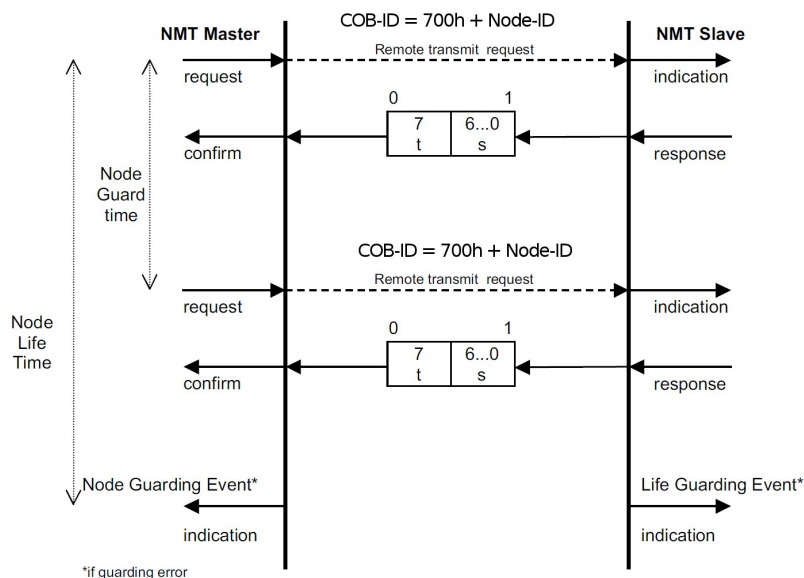
Available error codes indicated by CANopen but not necessarily supported by the manufacturer:

Error code	Description
0000h	Error reset or no error
1000h	<b>Node guarding error</b>
2000h	Current – generic error
2100h	Current, CANopen device input side – generic
2200h	Current inside the CANopen device – generic
2300h	Current, CANopen device output side – generic
3000h	Voltage – generic error
3100h	Mains voltage – generic
3200h	Voltage inside the CANopen device – generic
3300h	Output voltage – generic
4000h	Temperature – generic error
4100h	Ambient temperature – generic
4200h	Device temperature – generic
5000h	CANopen device hardware – generic error
5530h	<b>Flash memory error</b>
6000h	CANopen device software – generic error
6100h	Internal software – generic
6200h	User software – generic
6300h	Data set – generic
7000h	Additional modules – generic error
8000h	Monitoring – generic error
8100h	Communication – generic

<b>8110h</b>	CAN overrun (objects lost)
<b>8120h</b>	CAN in error passive mode
<b>8130h</b>	Life guard error or heartbeat error
<b>8140h</b>	Recovered from bus off
<b>8150h</b>	CAN-ID collision
<b>8200h</b>	Protocol error – generic
<b>8210h</b>	PDO not processed due to length error
<b>8220h</b>	PDO length exceeded
<b>8230h</b>	DAM MPDO not processed, destination object not available
<b>8240h</b>	Unexpected SYNC data length
<b>8250h</b>	RPDO timeout
<b>9000h</b>	External error – generic error
<b>F000h</b>	Additional functions – generic error
<b>FF00h</b>	Device specific – generic error

## 6.11 Node guarding protocol

This protocol is used to detect remote error in the network. Each NMT Slave uses one remote COB for the Node Guarding protocol. This protocol implements the provided initiated Error Control services.



S: the state of the NMT Slave

- 4: **STOPPED**
- 5: **OPERATIONAL**
- 127: **PRE-OPERATIONAL**

t: Toggle bit. The value of this bit must alternate between two consecutive responses from the NMT Slave. The value of the Toggle bit of the first response

after the Node Guarding protocol becomes active is 0. The Toggle bit in the Node Guarding protocol is only reset to 0 when reset\_communication is passed (no other change of the state resets the Toggle bit). If a response is received with the same value of the Toggle bit as in the preceding response then the new response is handled as if it was not received.

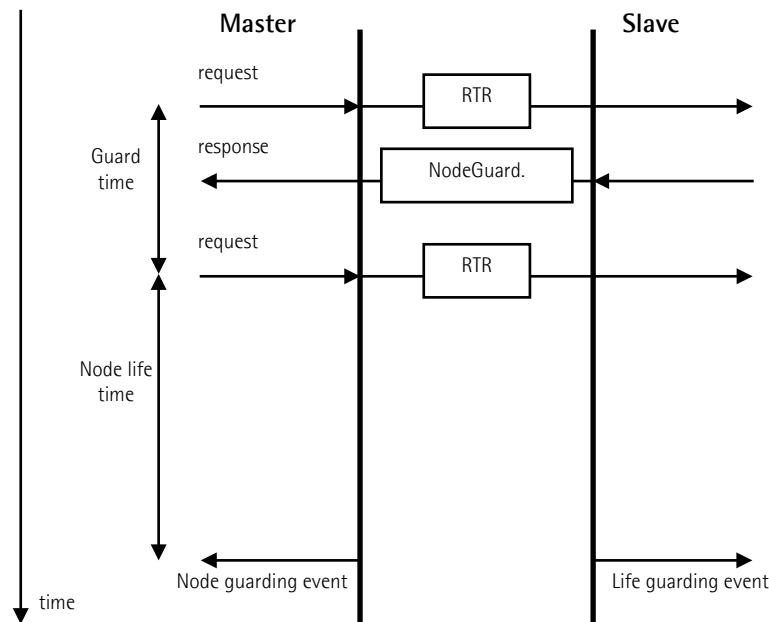
The NMT Master polls each NMT Slave at regular time intervals. This time-interval is called the guard time (see the **100C-00 Guard time** object) and may be different for each NMT Slave. The response of the NMT Slave contains the state of that NMT Slave. The **node life time** is given by the **100C-00 Guard time** object value multiplied by the **100D-00 Life time factor** object value. The node life time can be different for each NMT Slave. If the NMT Slave has not been polled during its life time, a remote node error is indicated through the 'Life Guarding Event' service.

A remote node error is indicated through the 'Node guarding event' service if:

- the remote transmit request is not confirmed within the node life time;
- the reported NMT Slave state does not match the expected state.

If it has been indicated that a remote error has occurred and the errors in the guarding protocol have disappeared, it will be indicated that the remote error has been resolved through the 'Node Guarding Event' and 'Life Guarding Event' services.

At system boot the "Node guarding protocol" is disabled; this protocol is enabled automatically as soon as the Master device sends an RTR message (Remote Transmission Request) the first time.



**100C-00 Guard time:** interval between two RTR messages.

**Node life time:** maximum time available for the encoder to receive an RTR message.



**Node life time** = 100C-00 Guard time \* 100D-00 Life time factor

"Node guarding" is enabled if **Node life time** ≠ 0.

If the Slave does not receive an RTR message before the **Node life time** has expired, it warns activating a "Life Guarding Event". Furthermore 1001-00 Error register and 1003 Predefined error field objects are updated and an error message is sent.

To reset the error send a **Reset node** command.

## 7 – Setting-up

Here following are some examples of transmission between Master and Slave devices.

A generic "ID" value is used to indicate the unit address; the address of the Master is always 0. All values are expressed in hexadecimal notation.

### 7.1 Setting the **Operational**, **Pre-operational** state

NMT message

Master → Slave

**Operational:**

**Pre-operational:**

COB-ID	Cmd	Node
000	01	ID
000	80	ID

### 7.2 Reading the value of the measuring step

#### 6501-00 Measuring step

Master → Slave encoder

COB-ID	Cmd	Index	Sub	Process data
601	40	01 65	00	- - - -

Slave encoder → Master

COB-ID	Cmd	Index	Sub	Process data
581	43	01 65	00	88 13 00 00
				Low ... High

→ 0000 1388h = 5 000 nm = 5 µm (Data bytes as an example)

### 7.3 Setting the value of the measuring step

#### 6005-01 Measuring step setting (e.g. 1 mm = 1000000 nm = 000F 4240h)

Master → Slave (Set request)

COB.ID	Cmd	Index	Sub	Process data
600+ID	23	05 60	01	40 42 0F 00

Slave → Master (Set confirmation)

COB.ID	Cmd	Index	Sub	Process data
580+ID	60	05 60	01	00 00 00 00

### 7.4 Setting the operating parameters

#### 6000-00 Operating parameters

(Scaling function control: 1 = enabled, Measuring direction: 0 = standard, Limit switch min. / Limit switch max.: 0 = disabled)

Master → Slave (Set request)

COB.ID	Cmd	Index	Sub	Process data
600+ID	2B	00 60	00	04 00 - -

Slave → Master (Set confirmation)

COB.ID	Cmd	Index	Sub	Process data			
580+ID	60	00	60	00	00	-	-

## 7.5 Setting the Preset value

**6003-00 Preset value** (preset = 1000 = 03E8h)

Master → Slave (Set request)

COB.ID	Cmd	Index	Sub	Process data			
600+ID	23	03	60	00	E8	03	00

Slave → Master (Set confirmation)

COB.ID	Cmd	Index	Sub	Process data			
580+ID	60	03	60	00	00	00	00

## 7.6 Setting the SYNC counter

**1801 TPD02 parameters sub 2** (n = 5 = 05h)

Master → Slave (Set request)

COB.ID	Cmd	Index	Sub	Process data			
600+ID	2F	01	18	02	05	-	-

Slave → Master (Set confirmation)

COB.ID	Cmd	Index	Sub	Process data			
580+ID	60	01	18	02	00	-	-

## 7.7 Disabling the SYNC mode

**1801 TPD02 parameters sub 1**

Read COB-ID used by PDO2:

Master → Slave (Req request)

COB.ID	Cmd	Index	Sub	Process data			
600+ID	40	01	18	01	-	-	-

Slave → Master (Req reply)

COB.ID	Cmd	Index	Sub	Process data			
580+ID	43	01	18	01	B0	B1	B2

COB-ID used by PDO2 = ( (B3<<24) | (B2<<16) | (B1<<8) | B0 )

set to 1 the most significant bit:

B3 |= 0x80;

Set new COB-ID used by PDO2:

Master → Slave (Set request)

COB.ID	Cmd	Index	Sub	Process data			
600+ID	23	01	18	01	B0	B1	B2

Slave → Master (Set confirmation)

COB.ID	Cmd	Index		Sub	Process data			
580+ID	60	01	18	01	00	00	00	00

## 7.8 Enabling the Cyclic mode

Set the cyclic time **6200-00 Cyclic timer** (100 ms = 64h)

Master → Slave (Set request)

COB.ID	Cmd	Index		Sub	Process data			
600+ID	2B	00	62	00	64	00	-	-

Slave → Master (Set confirmation)

COB.ID	Cmd	Index		Sub	Process data			
580+ID	60	00	62	00	00	00	-	-

Read COB-ID used by PDO1:

Master → Slave (Req request)

COB.ID	Cmd	Index		Sub	Process data			
600+ID	40	00	18	01	-	-	-	-

Slave → Master (Req reply)

COB.ID	Cmd	Index		Sub	Process data			
580+ID	43	00	18	01	B0	B1	B2	B3

COB-ID used by PDO1 = ( (B3<<24) | (B2<<16) | (B1<<8) | B0 )

set to 0 the most significant bit:

B3 &= 0x7F;

Set new COB-ID used by PDO1:

Master → Slave (Set request)

COB.ID	Cmd	Index		Sub	Process data			
600+ID	23	00	18	01	B0	B1	B2	B3

Slave → Master (Set confirmation)

COB.ID	Cmd	Index		Sub	Process data			
580+ID	60	00	18	01	00	00	00	00



### NOTE

To save new parameters execute the store parameters function (see the **1010-01 Store parameters** object).

When the power is turned off or in case of **Reset node** or **Reset communication** commands, parameters not saved are lost!

## 8 – Default parameters list

Default values are expressed in hexadecimal notation.

Parameters list	Default value		
1000-00 Device type	0008 0196		
1001-00 Error register	00		
1003 Predefined error field	-		
1005-00 COB-ID SYNC message	0000 0080		
1008-00 Manufacturer Device name	IF55LIN_CB*		
1009-00 Manufacturer Hardware version	-		
100A-00 Manufacturer Software version	-		
100C-00 Guard time	0000		
100D-00 Life time factor	00		
1014-00 COB-ID EMCY	0000 0080+NODE-ID		
1015-00 Inhibit time EMCY	0000		
1018 Identity object, sub 1	0000 012E		
1018 Identity object, sub 2	0000 000B		
1018 Identity object, sub 3	0001 0001		
1800 TPDO1 parameters, sub 1	4000 0180+NODE-ID		
1800 TPDO1 parameters, sub 2	FE		
1801 TPDO2 parameters, sub 1	4000 0280+NODE-ID		
1801 TPDO2 parameters, sub 2	01		
1802 TPDO3 parameters, sub 1	C000 0380+NODE-ID		
1802 TPDO3 parameters, sub 2	01		
1A00-01 TPDO1 mapping parameter, sub 1	6004 0020		
1A01-01 TPDO2 mapping parameter, sub 1	6004 0020		
1A02-01 TPDO3 mapping parameter, sub 1	6008 0040		
2104-00 Limit switch min	0000 0010		
2105-00 Limit switch max	0001 FFF0		
2200-01 Code Type (BIN/GRAY)	00		
2200-02 SSI Protocol	00		
2200-03 Number of SSI clocks	20		
2200-04 Physical Total Resolution [bits]	1E		
2200-05 Physical pulse measure [nm]	0000 1388		
3000-00 Baud rate	05		
3001-00 Node-ID	01		
6000-00 Operating parameters	0000		
Scaling function control	0		
Measuring direction	0		
Limit switch min.	0		
Limit switch max.	0		
6001-00 Total measuring range	0002 0000		
6002-00 Programmable total resolution	0002 0000		
6003-00 Preset value	0000 0000		
6005-01 Measuring step setting	0000 1388		
6200-00 Cyclic timer	0000		

6500-00 Operating status	0000		
6504-00 Supported alarms	0000		
6506-00 Supported warnings	0000		
6507-00 Profile and software version	0301 0101		
6508-00 Operating time	FFFF FFFF		
6509-00 Offset value	0000 0000		
650A-01 Manufacturer offset value, sub 1	0000 0000		
650B-00 Serial number	FFFF FFFF		

\* Text value

This page intentionally left blank

Document release	Release date	Description	HW	SW	EDS file version
1.0	01.09.2015	1st issue	1.0	1.0	V1
1.1	03.10.2019	New firmware, new EDS files, bypass function added and related parameters updated, setting range updated in some parameters, new POWER SUPPLY DIP switch	1.0	1.1	V2



Dispose separately

**lika**

**Lika Electronic**

Via S. Lorenzo, 25 • 36010 Carrè (VI) • Italy

Tel. +39 0445 806600

Fax +39 0445 806699



info@lika.biz • www.lika.biz