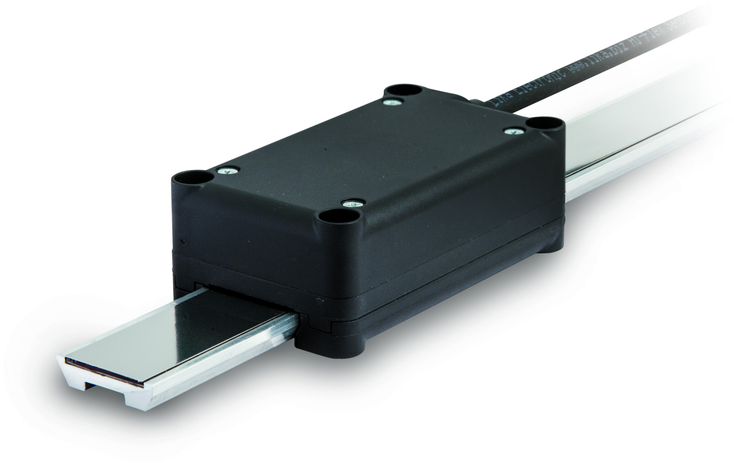
Smart encoders & actuators

User's guide

SMAG + MTAG



DS 406 profile for encoders

- SMAG absolute linear encoder
- MTAG guided profile, insensitive to dust and liquids
- Measuring length between 115 mm and 570 mm
- Programmable resolution down to 0.005 mm
- CANopen interface in compliance with DS 301 and DS 406 profiles

Suitable for the following models:

- SMAG -CB-5-...

Table of Contents

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 for submitting your comments, suggestions and criticisms.

Table of Contents

Subject index

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 **WARNING**, 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 **NOTE**, 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.
ℹ	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 **EXAMPLE** 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 linear encoders with guided profile of the **SMAG series equipped with CANopen interface**.
The SMAG sensors are engineered to measure linear displacements in industrial machines and automation lines.
The measuring system consists of a magnetic tape mounted on a rigid profile and a magnetic sensor. The tape has a coded sequence of poles; the distance between two poles is referred to as "pole pitch". The sensor moves along the magnetic tape (or, on the contrary, the magnetic tape is moved under the sensor), thus the sensor detects the displacement and issues an output signal equivalent to the one generated by an absolute encoder or an optical linear scale.
The sensor must be compulsorily paired with the **MTAG magnetic tape**. The measuring length is from 115 mm (4.528") to 570 mm (22.441"), see the order code.

For technical specifications please refer to the product datasheet.

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

In the first section (from chapter 1 to chapter 5) 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 SMAG linear encoder + MTAG are provided.

In the second section, entitled **CANopen Interface (DS 406)** (section 6), 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.

Application layer	The application layer is the communication entity of the OSI (Open System Interface) reference model. It provides communication services to the application program.
Application objects	Application objects are signals and parameters of the application program visible at the application layer API (application programming interface).
Application profile	Application profiles define all communication objects and application objects in all devices that the network consists of.
Asynchronous PDO	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.
Boot-up message	CANopen communication service transmitted whenever a node enters the **Pre-operational** state after initialization.
Bus	Topology of a communication network, where all nodes are reached by passive links, which allows transmission in both directions.
Bus analyser	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).
Bus arbitration	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 CMSA/CD (Carrier Sense Multiple Access/Collision Detection) with AMP (Arbitration on Message Priority). This allows bus arbitration without destruction of messages.
Bus length	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.
Bus off state	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.
CAN	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.
CAN protocol controller	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.
CAN transceiver	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.
CANopen	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.
CANopen application layer	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).
CANopen Manager	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.
CANopen Safety	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.
Certification	Official compliance test of components or devices to a specific standard. CiA officially certifies CANopen devices.
CiA DR 303	Draft recommendation for CANopen cabling and connector pin assignments, coding of prefixes and SI unit as well as LED usage.
CiA DS 102	Draft standard for high-speed transmission according to ISO 11898-2 using 9-pin D-sub connectors.
CiA DS 301	The CANopen application layer and communication profile specification covers the functionality of CANopen NMT (network management) Slave devices.
CiA DS 401	The CANopen device profile for generic I/O modules covers the definition of digital and analogue input and output devices.
CiA DS 404	The CANopen device profile for measuring devices and closed-

	loop controllers supports also multi-channel devices.
CiA DS 406	The CANopen device profile for encoders defines the communication of rotating as well as linear sensors.
CiA DSP 302	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.
CiA DSP 304	The CANopen safety protocol specification is approved by German authorities and is compliant to SIL class 3 applications.
CiA DSP 305	The Layer Setting Services (LSS) specify how to set node-ID and transmission rate via the CANopen network.
CiA DSP 306	This draft standard proposal defines format and content of Electronic Data Sheets (EDS) to be used in configuration tools.
CiA DSP 308	The CANopen framework for maritime applications defines redundancy of networks including swapping mechanism for SDOs and PDOs.
CiA DSP 309	Set of gateway specifications for CANopen to Ethernet-based networks (e.g. Modbus TCP(IP)).
CiA DSP 402	The CANopen device profile for drives and motion controllers defines the interface to frequency inverters, servo controllers as well as stepper motors.
CiA DSP 405	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.
CiA DSP 407	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.
CiA DSP 408	The CANopen device profile for hydraulic controllers and proportional valves is compliant to the bus-independent VDMA device profile.
CiA DSP 410	The CANopen device profile for inclinometer supports 16-bit as well as 32-bit sensors.
CiA DSP 412	The CANopen device profiles for medical equipment specify the interfaces for x-ray collimators, x-ray generators, stands and tables.
CiA DSP 413	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.
CiA DSP 414	The CANopen device profile for weaving machines specifies

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

Conformance test plan	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.
Conformance test tool	A conformance test tool is the implementation of a conformance test plan.
Consumer	In CAN networks a receiver of messages is called a consumer meaning the acceptance filter is opened.
D-sub connector	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.
Data link layer	Second layer in the OSI reference model providing basic communication services. The CAN data link layer defines data, remote, error, and overload frames.
Data type	Object attribute in CANopen defining the format, e.g. UNSIGNED8, INTEGER16, BOOLEAN, etc.
Default value	Object attribute in CANopen defining the pre-setting of not user-configured objects after power-on or application reset.
Device profile	A device profile defines the device-specific communication services including the configuration services in all details.
Draft Recommendation (DR)	This kind of recommendation is not fixed, but it is published. CiA's draft recommendations are not changed within one year.
Draft Standard (DS)	This kind of standard is not fixed, but it is published. CiA's draft standards are not changed within one year.
Draft Standard Proposal (DSP)	This kind of standard is a proposal, but it is published. CiA's draft standard proposals may be changed anytime without notification.
EDS checker	Software tool that checks the conformity of electronic data sheets. The CANopen EDS checker is available on CiA's web-site to be downloaded.
EDS generator	Software tool that generates CANopen electronic data sheets.
Electronic Data Sheet (EDS)	Electronic data sheets describe the functionality of a device in a standardized manner.
Emergency message	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.
EN 50325-4	CENELEC standard defining the CANopen application layer (version 4.0).
Entry category	Object attribute in CANopen defining whether this object is mandatory or optional.
Error code	CANopen specifies standardized error codes transmitted in emergency messages.

Error control message	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 **Pre-operational** state after initialization, and it is transmitted if remotely requested by the NMT Master (node guarding) or periodically by the device (heart-beat).
Event driven	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.
Event timer	The event timer is assigned in CANopen to one PDO. It defines the frequency of transmission.
Expedited SDO	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.
Flying Master	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.
Form error	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.
Function code	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).
Galvanic isolation	Galvanic isolation in CAN networks is performed by optocouplers or transformers placed between CAN controller and CAN transceiver chip.
Gateway	Device with at least two network interfaces transforming all seven OSI (open system interconnection) protocol layers, e.g. CANopen-to-Ethernet gateway.
Heartbeat	CANopen uses heartbeat message to indicate that a node is still alive. This message is transmitted periodically.
Heartbeat consumer time	The heartbeat consumer time defines the time when a node is regarded as no longer alive due to a missing heartbeat message.
Heartbeat producer time	The heartbeat producer time defines the transmission frequency of a heartbeat message.
Identifier	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.

Index	16-bit address to access the CANopen dictionary; for array and records the address is extended by an 8-bit Subindex.
Inhibit timer	Object in CANopen for PDOs and Emergency messages that forbids for the specified time (inhibit time) a transmission of this communication object.
Initialization state	NMT Slave state in CANopen that is reached automatically after power-on and communication or application reset.
Interface profile	CANopen profile that describes just the interface and not the application behaviour of device, e.g. gateway and bridge devices.
ISO 11898-1	International standard defining the CAN data link layer including LLC, MAC and PLS sub-layers.
ISO 11898-2	International standard defining the CAN high-speed MAU.
Life guarding	Method in CANopen to detect that the NMT Master does not guard the NMT Slave any more. This not recommended for new systems designs.
Line topology	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.
Master	Communication or application entity that is allowed to control a specific function. In networks this is for example the initialization of a communication service.
Multiplexed PDO (MPDO)	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.
Network length	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.
Network management	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.
Network variables	Network variables are used in programmable CANopen devices to be mapped into PDOs after programming the device.
NMT	Network management in CANopen.
NMT Master	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.
NMT Slave	The NMT Slaves receive the NMT message, which contains commands for the NMT state machine implemented in CANopen devices.

NMT state machine	The NMT state machines support different states and the highest prior CAN message transmitted controls the transition to the states by the NMT Master.
Node guarding	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.
Node ID	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.
Object Dictionary	Heart of each CANopen device containing all communication and application objects.
Operational state	In the NMT **Operational** state all CANopen communication services are available.
PDO mapping	In PDOs, there may be mapped up to 64 objects. The PDO mapping is described in the PDO mapping parameters.
Pin assignment	Definition of the use of connector pins.
Pre-defined connection set	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.
Pre-operational state	In the NMT **Pre-operational** state no CANopen PDO communication is allowed.
Process Data Object (PDO)	Communication object defined by the PDO communication parameter and PDO mapping parameter objects. It is an unconfirmed communication service without protocol overhead.
Producer	In CAN networks a transmitter of messages is called a producer.
Protocol	Formal set of conventions and rules for the exchange of information between nodes, including the specification of frame administration, frame transfer and physical layer.
Receiver	A CAN node is called receiver or consumer, if it is not transmitter and the bus is not idle.
Redundant networks	In some safety-critical applications (e.g. maritime systems), redundant networks may be required that provide swapping capability in case of detected communication failures.
Remote frame	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.
Remote transmission request (RTR)	Bit in the arbitration field indicating if the frame is a remote frame (recessive value) or a data frame (dominant value).
Repeater	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.
Reset application	This NMT command resets all objects in CANopen devices to the default values or the permanently stored configured values.
Reset communication	This NMT command resets only the communication objects in CANopen devices to the default values or the permanently stored configured values.
RPDO	The Receive Process Data Object (RPDO) is a communication object that is received by a CANopen device.
SDO block transfer	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.
SDO Manager	The SDO Manager handles the dynamic establishment of SDO connections. It resides on the very same node as the NMT Master.
Segmented SDO	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.
Server SDO	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).
Service Data Object (SDO)	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.
SI unit	International system of units for physical values as specified in ISO 1000:1983.
Stopped state	NMT state in which only NMT messages are performed and under some conditions error control messages are transmitted.
Sub-index	8-bit sub-address to access the sub-objects of arrays and records.
Suspend transmission	CAN controllers in error passive mode have to wait additional 8 bit-times before the next data or remote frame may be transmitted.
SYNC message	Dedicated CANopen message forcing the receiving nodes to

	sample the inputs mapped into synchronous TPDOs. Receiving this message causes the node to set the outputs to values received in the previous synchronous RPDO.
Termination resistor	In CAN high-speed networks with bus topology, both ends are terminated with resistors in order to suppress reflections.
TIME message	Standardized message in CANopen containing the time as a 6-byte value given as ms after midnight and days after 1st January 1984.
TPDO	The Transmit Process Data Object (TPDO) is a communication object that is transmitted by a CANopen device.
Transmission type	CANopen object defining the scheduling of a PDO.
Value definition	Detailed description of the value range in CANopen profiles.
Value range	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 devices;
- 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 the power supply before connecting the device;
- connect according to explanation in the "4 - Electrical installation" section on page 25;
- in compliance with 2014/30/EU norm on electromagnetic compatibility, following precautions must be taken:
  - before handling and installing the equipment, 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 cable shield 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 unit;
- do not stretch the cable; do not pull or carry by cable; do not use the cable as a handle.

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 power supply disconnected and stationary mechanical devices;
- do not disassemble the unit;
- do not tool the unit;
- delicate electronic equipment: handle with care; do not subject the unit to knocks or shocks;
- protect the unit against acid solutions or chemicals that may damage it;
- respect the environmental characteristics declared by manufacturer;
- we suggest installing the unit providing protection means against waste, especially swarf as turnings, chips, or filings; should this not be possible, please make sure that adequate cleaning measures (as for instance brushes, scrapers, jets of compressed air, etc.) are in place in order to prevent the sensor and the magnetic scale from jamming.

2 - Identification

Device can be identified through the **order code** and the **serial number** printed on the label applied to its body. 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 must be carried out by qualified personnel only, with power supply disconnected and mechanical parts compulsorily in stop.

3.1 Overall dimensions (Figure 1)

(values expressed in mm)

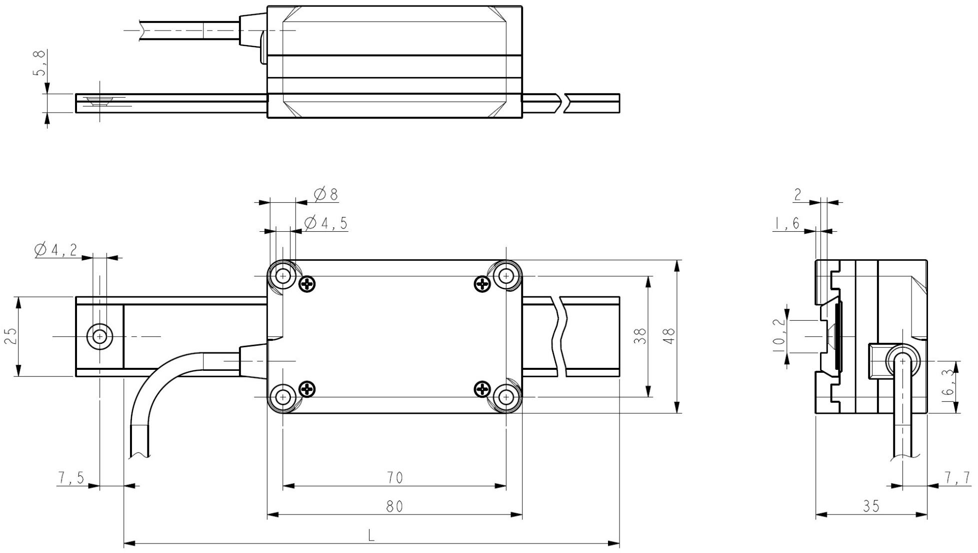


Figure 1

L = 195 mm / 7.677" ÷ 650 mm / 25.59" (see the order code of the MTAG profile)

3.2 Mounting instructions

Install and use the product in conformance with the protection level it has been designed to.
Protect the unit against knocks, frictions and solvents and avoid temperatures over the allowed range.
Make sure the unit is mounted where hard or sharp objects (e.g. turnings, chips or filings) do not come into contact with the magnetic tape and the bottom of the sensor head. If these conditions cannot be avoided provide adequate cleaning measures (as for instance brushes, scrapers, jets of compressed air, etc.) in order to prevent the sensor and the magnetic scale from jamming.

The profile of the magnetic tape must be inserted in the track at the bottom of the sensor enclosure; the shape of the track is univocal in order to prevent mounting errors.

Two mounting options are allowed:

1. you can fasten the profile of the magnetic tape to a fixed support and move the sensor;
2. or, on the contrary, you can fasten the sensor to a fixed support and move the profile of the magnetic tape.

It is the customer's duty to evaluate the type of installation that is most suitable for his application. Anyway both the sensor and the profile (according to your choice) must be able to slide freely and accomplish a smooth travel.

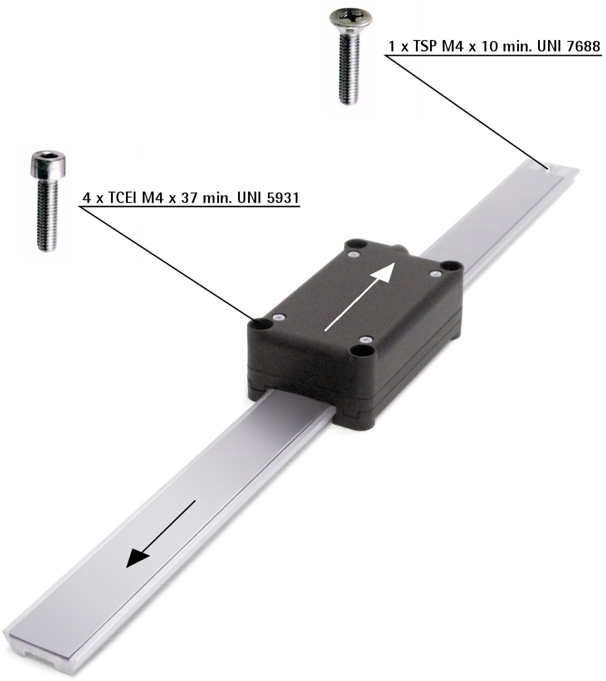


Figure 2

3.3 Sensor (Figure 2)

The sensor can be secured:

1. to a sliding carriage;
2. to an actuator;
3. to a mobile support;
4. or to any fixed support.

Use **four TCEI M4 UNI 5931 type cylinder head screws** inserted from the top in the holes provided at the four angles of the enclosure. The screws must be 37 mm min. long (see Figure 1). **The maximum tightening torque is 1.2 Nm.** We suggest adding a few drops of threadlocker of medium bond strength.

The fixing support will be placed at the bottom of the sensor case. Make sure that the cable does not block or hamper the movement of the sensor.

3.4 Profile of the magnetic tape (Figure 2)

It is mandatory to pair the sensor with the **MTAG type magnetic tape**.
The profile of the magnetic tape can be secured:

1. to a sliding device driven manually, pneumatically or by a motor: in this first case the tape will move back and forth while the sensor will be mounted in a fixed position;
2. or to a fixed support: in this case, the tape will be mounted in a fixed position while the sensor will move back and forth.

The profile must be installed so that the active magnetic surface (black side) is turned towards the active surface of the sensor (electronic card).
Use **one TSP M4 UNI 7688 type countersunk head screw** to secure the profile of the magnetic tape; insert the screw in the hole provided at one end of the profile. The screw is 10 mm min. long. **The maximum tightening torque is 2.5 Nm**.
The length of the profile ranges between 195 mm / 7.677" and 650 mm / 25.59".

WARNING
Do not expose the sensor and the magnetic scale to magnetic fields. Avoid any possible interference of the unit with magnetic fields.

3.5 Measuring length (Figure 1)

The **maximum tape length L** is between 195 mm / 7.677"" and 650 mm / 25.59" (for further information refer the order code in the product datasheet). As the sensor area has always to be fully within the limits of the tape magnetic surface, then the **maximum measuring length ML** is the maximum tape length minus the sensor length (see the Figure 1) = L – 80 mm / 3.149" (115 mm / 4.528" ÷ 570 mm / 22.441").

3.6 Standard counting direction (Figure 2)

The positive counting direction (count up information) is achieved when the sensor moves on the tape according to the white arrow shown in Figure 2; or, on the contrary, when the tape moves according to the black arrow shown in Figure 2. For further information see the bit 3 **Code sequence** in the **6000-00 Operating parameters** object, see on page 47).

4 - Electrical installation

4.1 Electrical connections

WARNING
Electrical connections must be carried out by qualified personnel only, with power supply disconnected and mechanical parts compulsorily in stop.

WARNING
To avoid conflicts between the nodes, before installing the unit in the CANopen network please always check the Node-ID and configure a new one, if required. If you need to install several units, please check the Node-ID singularly and connect them one by one.

For any information on the mechanical and the electrical characteristics of the device please refer to the technical catalogue.

Description	T6 cable
+10Vdc +30Vdc Power supply voltage	Brown
0Vdc Power supply voltage	White
CAN_L (BUS IN)	Green
CAN_H (BUS IN)	Yellow
CAN_L (BUS OUT)	Grey
CAN_H (BUS OUT)	Pink

4.1.1 T6 cable specifications

Model	: T6 Li-YCY type cable
Wires	: 3 x 2 x 0.25 mm^2 twisted pair cable
Jacket	: flame retardant PVC
Shield	: tinned copper wire braid
External diameter	: 5.7 mm (2.244")
Min. bending radius	: Ø x 10
Operating temperature	: -20°C +75°C (-4°F +167°F)
Impedance	: ≤ 79 Ω/km

NOTE
The provided cable is not a CANopen certified cable, anyway it is fitted with bus input and bus output to avoid installing stubs in the CANopen network.

4.2 Ground connection

To minimize noise connect properly the cable shield to ground on user's side. 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 interference must be carried out by the user. You are advised to provide the ground connection as close as possible to the device.

4.3 Baud rate

The transmission rate can be set only in a software mode.

The baud rate is set next to the **3000-00 Baud rate** object in the "Object Dictionary", it can be modified via SDO messages.

The default baud rate is 500 Kbit/s.

The allowed baud rate values are: 20 Kbit/s, 50 Kbit/s, 100 Kbit/s, 125 Kbit/s, 250 Kbit/s, **500 Kbit/s**, 800 Kbit/s, 1000 Kbit/s.

4.4 Node-ID

The node number can be set only in a software mode.

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

The node number is set next to the **3001-00 Node-ID** object in the "Object Dictionary", it can be modified via SDO messages.

Default node ID is 1.

WARNING

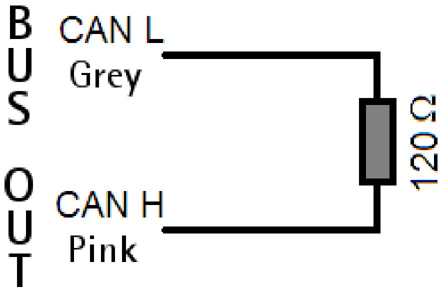
To avoid conflicts between the nodes, before installing the unit in the CANopen network please always check the Node-ID and configure a new one, if required. If you need to install several units, please check the Node-ID singularly and connect them one by one.

The boot-up message sent at initialization also contains the node-ID of the unit. See on page 33.

Once the communication has been activated, a different baud rate and node number can be set (see the **3000-00 Baud rate** and **3001-00 Node-ID** objects). After having set new values, transmit a **Reset node** command and then save the parameters (see the **1010-01 Store parameters** object).

4.5 Bus termination

This unit is not equipped with a bus termination resistor. If the unit has to be installed at the ends of the transmission line (either the first or the last device of the network), the bus termination resistor is required and must be provided outside the device



(120Ω bus termination resistor between CAN High and CAN Low outputs) as shown in the scheme.

4.6 Diagnostic LEDs

This unit is not equipped with diagnostic LEDs.

5 - Quick reference

Using the default settings provided by the manufacturer, you can switch on the device and read immediately the position value.

Follow the instructions below to:

- read the measuring step (physical resolution) 6501-00 Measuring step;
- set a custom cyclic time 6200-00 Cyclic time ≠ 0;
- set the Operational mode;
- read the current position of the encoder (cyclic and/or synchronous mode).

Default Baud rate and Node-ID are:
Baud rate = 500 Kbit/s
Node-ID = 1

Read the measuring step (physical resolution) 6501-00 Measuring step

Master → SMAG

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

SMAG → 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

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

Master → SMAG (Set request)

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

SMAG → Master (Set confirmation)

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

Set the Operational mode

Master → SMAG

COB-ID	Cmd	Node
000	01	01

Read position every 100 ms

SMAG → 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 any 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

Let's assume that we are using the physical resolution.
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 more information please refer to the **6004-00 Position value** object on page 53.

NOTE

For further examples please refer to the "7 – Set up" section on page 62.

6 - CANopen® interface (DS 406)

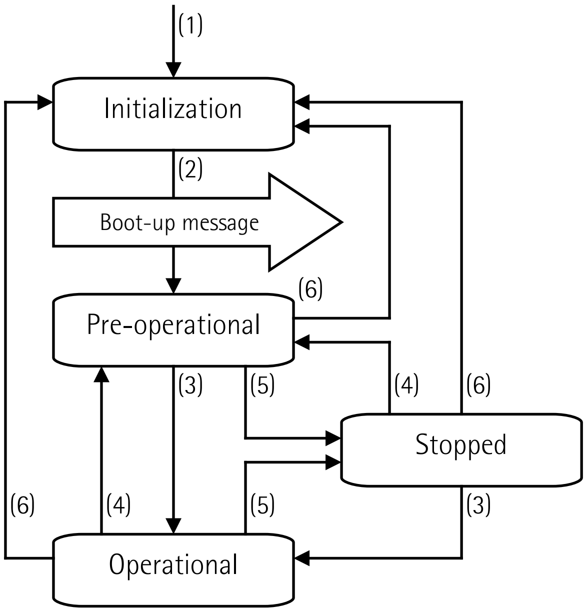
CANopen® devices from Lika Electronic are Slave units and comply with the "Device profile for encoders", Class 2.
For any omitted information on the CANopen protocol, refer to the "CiA Draft Standard 301" and "CiA Draft Standard 406" documents available at the address **www.can-cia.org**.

6.1 EDS file

CANopen® devices are supplied with their own EDS file **Lika_SMAG_DS406_Vx.eds**. When you need to download the file please refer to the address **www.lika.biz > LINEAR ENCODERS > ABSOLUTE MAGNETIC SENSORS (LINECOD)**).
EDS file must be installed in the CANopen® Master device.
Vx is intended to indicate the file version.
EDS file is available in both English (**Lika_SMAG_DS406_Vx_en.eds**) and Italian (**Lika_SMAG_DS406_Vx_it.eds**) language versions.

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: **Start remote node**
(4)	NMT message: **Enter pre-operational**
(5)	NMT message: **Stop remote node**

(6)	NMT message: **Reset node** or **Reset communication** 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. 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. Slave cannot send PDO messages.
To switch the Slave device to **Operational** state the Master must send a **Start remote node** command using an NMT message.

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 35) 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 to the **Pre-operational** state the Master must send an **Enter pre-operational** command using an NMT message.

6.2.4 Stopped state

In this state the Slave device is forced to stop the communication with the Master (except the node guarding, if active).
Communication using PDO and SDO messages is not allowed.
To switch the Slave device to either **Pre-operational** or **Operational** state the Master must send the specific commands **Enter pre-operational** or **Start remote node** using an NMT message.

6.3 Communication objects

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

- Network management NMT protocol: NMT protocols are used to issue state machine change commands (i.e. to start and stop the devices), detect remote device boot-ups and 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 (position information).
  - Emergency: error messages are triggered at each error event.
  - Nodeguard: used to request the state of the Slave.

Relation between device states and communication objects:

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

6.3.1 Pre-defined connection set

Master → Slave transmission		
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 (tx or rx) is viewed from the Slave device.

6.4 NMT objects

NMT structure:

COB-ID (11 bits)	
Func. code	Node ID
0000	0

2 CAN Data Bytes	
Command	Slave ID
NMT Func.	Slave ID

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

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

Boot-up message structure:

COB-ID(hex)
700+Node ID

1 CAN Data Byte
00

6.6 PDO objects

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

PDO structure:

IDENTIFIER	
COB-ID(hex)	
F.C.	Node-ID

4 CAN Data Bytes			
Byte 0	Byte 1	Byte 2	Byte 3
$2^7 - 2^0$	$2^{15} - 2^8$	$2^{23} - 2^{16}$	$2^{31} - 2^{24}$
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 time** 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 PDO1 parameters**, sub 1 object).

PDO2 and PDO3 Sync mode: synchronous transmission

The transmission of the position value is managed 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.
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 numbers.

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 PDO2 parameters**, sub 2 object.
For PDO3 refer to the **1802 PDO3 parameters**, sub 2 object.

SYNC transmission mode can be enabled (or disabled) by setting to 0 (or 1) the most significant bit (msb) of COB-ID used by PDO (**1801 PDO2 parameters** / **1802 PDO3 parameters**, sub1 objects).

NOTE
Several transmission modes 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 35).
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).

SDO structure:

IDENTIFIER	
COB-ID(hex)	
F.C.	Node-ID

from 4 to 8 CAN data bytes							
0	1	2	3	4	5	6	7
Com	Index		Sub	Data			
1 byte	LSB	MSB	1 byte	LSB	...	...	MSB

Com	command
Index	parameter index
Sub	parameter sub-index
Data	parameter value (either read in or written to the parameter)

6.7.1 Command

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

Three types of telegram are available:

- Set: used to send the configuration parameters to a device;
- Req: used by the Master to read data from a Slave device;
- Warnings: used by a Slave to send error messages to the Master (e.g. **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	Warning	S → M reply	4 bytes

6.8 Object dictionary

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

Unsigned16 data type:

Data bytes	
byte 4	byte 5
LSByte	MSByte

Unsigned32 data type:

Data bytes			
byte 4	byte 5	byte 6	byte 7
LSByte	...	...	MSByte

Unsigned64 data type:

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

6.8.1 Standard objects (DS 301)

1000-00 Device type

[Unsigned32, ro]
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 “7.10 Emergency objects” section on page 59.

- **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]
Default = 0000 0080h

1008-00 Manufacturer Device name

[String, ro]
It shows the name of the device.
Default = “SMAG-CB”

1009-00 Manufacturer Hardware version

[String, ro]
It shows the hardware version of the device.

100A-00 Manufacturer Software version

[String, ro]
It shows the software version of the device.

100C-00 Guard time

[Unsigned16, rw]
It is intended to set the Guard time expressed in milliseconds.
The **100C-00 Guard time** object is used in the "Node guarding protocol" controlled by the Master. For more details see the "7.11 Node guarding protocol" section on page 60.
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 "7.11 Node guarding protocol" section on page 60.
Default = 00h

1010-01 Store parameters

[Unsigned32, rw]
This object allows to save all parameters in the 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

[Unsig32, rw]

This object is used 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 for sending emergency messages (EMCY).

Please note that, as the node address is set via software, the set value is always retained, unless a software procedure for setting a new address is forced at power on.

Default = 80h+NodeID

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 number** provided by CIA organization [Unsigned32, ro]
  Default = 0000 012Eh
- **02 Product number** [Unsigned32, ro]
  Default = 0000 0009h
- **03 Revision number** [Unsigned32, ro]
  Default = 0001 0001h

1800 PDO1 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 33.
See the **6200-00 Cyclic time** 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
30	0	RTR allowed on this PDO (**not implemented**)
	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 0180h + NodeID (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".
As the node address is set via software, the set value is always retained, unless a software procedure for setting a new address is forced at power on.

- **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 = FEh (cyclic transmission, see hereafter and the **6200-00 Cyclic time** 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: **Value range of parameter exceeded**) is generated.
If the value next to the **6200-00 Cyclic time** object ≠ 0, the PDO message is sent cyclically and the interval between two messages is the time set next to the **6200-00 Cyclic time** object; otherwise, if the value next to the **6200-00 Cyclic time** object = 0, the PDO message is not sent.

1801 PDO2 parameters

PDO2 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 33.

- **01 COB-ID of the PDO2** [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 (**not implemented**)
	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 + NodeID (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".

As the node address is set via software the set value is always retained, unless a software procedure for setting a new address is forced at power on.

- **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)

Position value is transmitted after the set number of SYNC commands.

Interval in SYNC commands must be set next to this **1801 PDO2 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: **Value range of parameter exceeded**) is generated.

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

1802 PDO3 parameters

PDO3 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 33.

- **01 COB-ID of the PDO3** [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 (**not implemented**)
	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 = C000 0380h + NodeID (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".

As the node address is set via software the set value is always retained, unless a software procedure for setting a new address is forced at power on.

- **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).

Position value is transmitted after the set number of SYNC commands.

Interval in SYNC commands must be set next to this **1802 PDO3 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: **Value range of parameter exceeded**) is generated.

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

NOTE

- 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 (**180xh**, sub1 object).
- Cyclic transmission or synchronous transmission can be modified by setting the **180xh** sub 2 object:
  01h: synchronous transmission every SYNC command;
  02h: synchronous transmission every two SYNC commands;
  ...
  FEh: cyclic transmission.

  If **6200-00 Cyclic time** ≠ 0 → "cyclic transmission": cycle time is set next to the 6200h object;
  if **6200-00 Cyclic time** = 0 → the PDO message is not sent.

1A00-01 TPDO1 mapping parameter

[Unsigned32, rw]

This object contains the mapped position value of the transducer according to the DS 406 device profile.

Default = 6004 0020h

1A01-01 TPDO2 mapping parameter

[Unsigned32, rw]

See the **1A00-01 TPDO1 mapping parameter**, sub1 object.

Default = 6004 0020h

1A02-01 TPDO3 mapping parameter

[Unsigned32, rw]

See the **1A00-01 TPDO1 mapping parameter**, sub1 object.

Default = 6008 0040h

6.8.2 Objects specific to the manufacturer

2104-00 Limit switch min

[Unsigned32, rw]

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

If the encoder position is greater than the value set in this object, 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, 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 intended to set the highest software limit switch (+) in the travel.

If the encoder position is less than the value set in this object, 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, 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

3000-00 Baud rate

[Unsigned8, rw]

This object is meant to set the baud rate (transmission rate) according to the following table:

Data byte	Baud rate
00h	20 Kbit/s
01h	50 Kbit/s
02h	100 Kbit/s
03h	125 Kbit/s
04h	250 Kbit/s
05h	**500 Kbit/s (default)**
06h	800 Kbit/s
07h	1000 Kbit/s

To change the baud rate value you have to:

- set the **3000-00 Baud rate** object;
- send a **Reset node** command (or a **Reset communication** command);
- save the parameter (see the **1010-01 Store parameters** object);
- set the Master to the new baud rate too.

Default = 05h

Master → Slave

COB-ID	Cmd	Index		Sub	Data byte
600+ID	2F	00	30	00	see the table

Slave → Master (confirmation)

COB-ID	Cmd	Index		Sub	Data byte
580+ID	60	00	30	00	00

Master → Slave (**Reset node**)

COB-ID	Cmd	Slave ID
000	81	ID

Set the Master device to the new baud rate:

Slave → Master (Boot-up with new baud rate)

COB-ID	Cmd
700+ID	00

NOTE

To save the new Baud rate value execute the store parameters function (see the **1010-01 Store parameters** object).
When the power is turned off, parameters not saved will be lost.

3001-00 Node-ID

[Unsigned8, rw]
This object is meant to set the node identifier (node ID) of the device. The node addresses are allowed between 1 and 127. The default value is 1.

To change the Node-ID value you have to:

- set the **3001-00 Node-ID** object;
- send a **Reset node** command;
- save the parameter (see the **1010-01 Store parameters** object).

Default = 01h

Master → Slave

COB-ID	Cmd	Index		Sub	Data byte
600+ID	2F	01	30	00	new Node-ID

Slave → Master (confirmation)

COB-ID	Cmd	Index		Sub	Data byte
580+ID	60	01	30	00	00

Master → Slave (**Reset node**)

COB-ID	Cmd	Slave ID
000	81	old ID

Slave → Master (Boot-up with new Node-ID)

COB-ID	Cmd
700+ID	00

NOTE

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

When the power is turned off, parameters not saved will be lost.

6.8.3 Device profile objects (DS 406)

6000-00 Operating parameters

[Unsigned16, rw]

Bit	Function	bit = 0	bit = 1
0 - 1	not used		
2	**Scaling function control**	disabled	enabled
3	**Code sequence**	standard	inverted
4 ... 11	not used		
12	**Limit switch min.**	disabled	enabled
13	**Limit switch max.**	disabled	enabled
14 - 15	not used		

Default = 0000h

Scaling function control

When this option is disabled (bit 2 **Scaling function control** = 0), the device uses the physical resolution values (see the **6501-00 Measuring step** object and the default value of the **6001-00 Total measuring range** / **6002-00 Total measuring range** objects). On the contrary, if it is enabled (bit 2 **Scaling function control** = 1), the user is allowed to enter the custom values in the **6001-00 Total measuring range** / **6002-00 Total measuring range** and **6005-01 Measuring step setting** objects.

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 55.

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

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

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 Total measuring range** 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).

Code sequence

This is intended to set if the count is increasing (count up information) either when the encoder moves in the standard direction or when the encoder moves in reverse of the standard direction. When **Code sequence** = 0, the position value increases when the device moves in the standard direction; on the contrary, when **Code sequence** = 1, the position value increases when the device moves in reverse of the standard direction. For more information on the standard and inverted counting direction please refer to the section "3.6 Standard counting direction (Figure 2)" on page 24).

To know what type of **Scaling function control** is currently enabled, you can read the bit 3 **Code sequence** of the **6500-00 Operating status** object, see on page 55.

WARNING

Every time you change the **Code sequence**, 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).

Limit switch min.
Limit switch max.

It allows 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 44.

6001-00 Total measuring range

[Unsig32, rw]

WARNING

This object can be changed only if the bit 2 **Scaling function control** in the **6000-00 Operating parameters** object is set to "=1"; otherwise the user cannot change the value in this object and the system will use the maximum number of physical information (i.e. the default value of this **6001-00 Total measuring range** object) to calculate the encoder position.

NOTE

Please note that this object is exactly the same as the **6002-00 Total measuring range** object. When you change the value next to this **6001-00 Total measuring range** object you also change the value next to the **6002-00 Total measuring range** object; and vice versa.

It sets the number of distinguishable steps over the total measuring range. In other words, it sets the maximum number of information the encoder will provide for the travel it has to measure. The value has to be expressed in

number of information (pulses). It has to be comprised between 1 and 2^{17} = 131,072.
Default = 0002 0000h

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

$$\text{Max. number of information} = \frac{\text{Max. profile length}}{\text{Resolution}}$$

As you can see in the order code, the resolution is 5 µm (0.005 mm), i.e. one measuring step is 5 µm. The length of the MTAG magnetic tape is between 195 mm / 7.677" and 650 mm / 25.59".
Thus it will be as in the following examples.

EXAMPLE
SMAG-CB-5-... encoder
MTAG-650-... magnetic tape
Physical resolution: 0.005 mm (see the order code)

$$\text{Max. number of information} = \frac{\text{Max. profile length}}{\text{Resolution}} = \frac{650}{0.005} = 130{,}000$$

Now you have to "round up" the result to the next highest power of 2, that is: 130,000 ≈ 131,072 = 2^{17}. Thus the max. number of information will be 20000h. If you need a custom measuring length, you need to enable the **Scaling function control** and then set a value equal to or lower than 2^{17} = 131,072 = 20000h 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^{17} - 1, i.e. 131,071.

...	**131069**	**131070**	**131071**	**0**	**1**	**2**	**...**

EXAMPLE
SMAG-CB-5-... encoder
MTAG-240-... magnetic scale
Physical resolution: 0.005 mm (see the order code)

$$\text{Max. number of information} = \frac{\text{Max. profile length}}{\text{Resolution}} = \frac{240}{0.005} = 48{,}000$$

Now you have to "round up" the result to the next highest power of 2, that is: 48,000 $\approx$ 65,536 = 2^{16}. Thus the max. number of information will be 10000h. You can either retain the physical resolution (20000h) or set a custom resolution, for instance the one resulting from the max. number of information (10000h).
If you need this custom measuring length, you need to enable the **Scaling function control** and then set a value equal to or lower than 2^{16} = 65,536 = 10000h 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^{16} - 1, i.e. 65,535.

...	**65533**	**65534**	**65535**	**0**	**1**	**2**	**...**

NOTE
Please note that the total hardware measuring range (i.e. the total physical resolution) is always 2^{17} = 131,072 (0002 0000h = default value), regardless of the profile length. So possible values are less than or equal to the total hardware measuring range (131,072).

WARNING
Every time you change the value in this object 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).

6002-00 Total measuring range

[Unsigned32, rw]

WARNING
This object can be changed only if the bit 2 **Scaling function control** in the **6000-00 Operating parameters** object is set to "=1"; otherwise the user cannot change the value in this object and the system will use the maximum number of physical information (i.e. the default value of this **6002-00 Total measuring range** object) to calculate the encoder position.

NOTE
Please note that this object is exactly the same as the **6001-00 Total measuring range** object. When you change the value next to this **6002-00 Total measuring range** object you also change the value next to the **6001-00 Total measuring range** object; and vice versa.

For any information on setting this **6002-00 Total measuring range** object, please refer to the **6001-00 Total measuring range** object.

Default = 0002 0000h

WARNING

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

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 tape. The chosen physical position will get the value set next to this object and all 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.

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	00

Encoder → Master (Set confirmation)

COB-ID	Cmd	Index		Sub	Process data			
580+ID	60	03	60	00	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 the default value (physical resolution) of the **6001-00 Total measuring range** / **6002-00 Total measuring range** objects - 1 (= 2^{17} - 1 = 131,071).
- 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 the value set in the **6001-00 Total measuring range** / **6002-00 Total measuring range** objects -1.

WARNING

6003-00 Preset value must not be greater than the total measuring range (see the **6001-00 Total measuring range** / **6002-00 Total measuring range** objects).
Every time you enable the scaling function (see bit 2 **Scaling function control** in the **6000-00 Operating parameters** object) and/or change the scaling values (see the **6001-00 Total measuring range** / **6002-00 Total measuring range** 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).

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 PDO1 parameters and 1801 PDO2 parameters objects (see on page 39). 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 any 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 the 6000-00 Operating parameters object is enabled = 1).

EXAMPLE 1

Scaling function control = 0

6501-00 Measuring step = 0000 1388h = 5000 nm = 0.005 mm

6004-00 Position value = 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

EXAMPLE 2

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 6501-00 Measuring step object.

This object is used to set a custom resolution (otherwise referred to as measuring step), expressed in nanometres [nm]. Eg.: 1 mm = 1 000 000 nm = 000F 4240h.

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 measuring step which is 5 µm (see the **6501-00 Measuring step** object).
Default = 0000 1388h

WARNING
The value in the **6005-01 Measuring step setting** object must be greater than or equal to the value set in the **6501-00 Measuring step** object.
Every time you change the value in this object 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).

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 **1802 PDO3 parameters** object (see on page 42). See also the **6004-00 Position value** object.

6200-00 Cyclic time

[Unsigned16, rw]
Cyclic timer value is used in asynchronous transmission mode (**Transmission Type** = FEh) to set the interval between two following PDO transmissions during a cyclic communication.
If the value next to the **6200-00 Cyclic time** object ≠ 0, the PDO message is sent cyclically and the interval between two messages is the time set next to the **6200-00 Cyclic time** object; otherwise, if the value next to the **6200-00 Cyclic time** object = 0, the PDO message is not sent.
Value is expressed in milliseconds. See on pages 33 and 39.
Default = 0000h

6500-00 Operating status

[Unsigned16, ro]

Bit	Function	bit = 0	bit = 1
0 - 1	not used		
2	**Scaling**	Disabled	Enabled
3	**Code sequence**	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 status**	**Stop** / **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 disabled or enabled. If the value "=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 47.

Code sequence

It shows the value that is currently set through the bit 3 **Code sequence** in the **6000-00 Operating parameters** object. If the bit is "=0" the output encoder position value has been set to increase when the sensor moves in the standard direction; if the bit is "=1" instead the output encoder position value has been set to increase 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 47.

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 status

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 30.
bit 15 = 0: **Stopped** or **Pre-operational** state;
bit 15 = 1: **Operational** state.

6501-00 Measuring step

[Unsigned32, ro]
This object is intended to show the physical measuring step of the encoder, i.e. its physical resolution expressed in nanometres [nm]. The physical resolution of the SMAG linear encoder is 5 µm = 5000 nm (0000 1388h). If you want to set a custom resolution see the **6005-01 Measuring step setting** object.

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]
Default = 0000h (no alarms supported)

6506-00 Supported warnings

[Unsigned16, ro]
Default = 0000h (no warnings supported)

6507-00 Profile and software version

[Unsigned32, ro]
It shows the profile and software version.
Profile version for encoders = 3.1
Software version = 1.1
Default = 0301 0101h

6508-00 Operating time

[Unsigned32, ro]
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 38).
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 51.
Default = 0000 0000h

650A-01 Manufacturer offset value

[Integer32, ro]
Default = 0000 0000h (not used)

650B-00 Serial number

[Unsigned32, ro]
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 follow the list and the meaning of the SDO abort code indicated by CANopen but not necessarily supported by the manufacturer. For complete information please refer to the "SDO abort transfer protocol" section in "CiA Draft Standard 301" document available at address 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

7.10 Emergency objects

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

EMCY structure:

IDENTIFIER
COB-ID(hex)
See the 1014-00 COB-ID EMCY object

CAN Data Byte			
0	1	2	3...7
Error code		Sub of error register	Specific code
LSB	MSB	01	00 ... 00

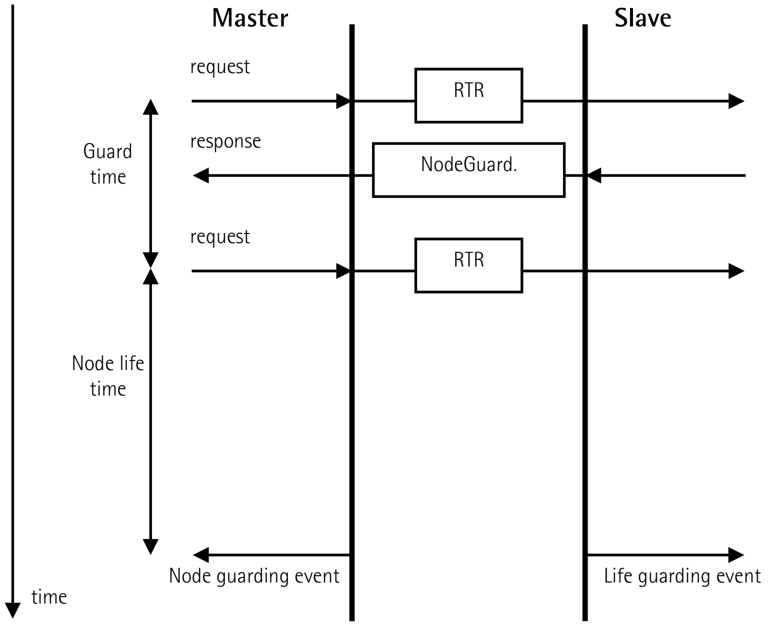
Available error codes indicated by CANopen but not necessarily supported by the manufacturer:

Error code	Description
0000h	Error reset or no error
1000h	Generic error, **Node guarding error**
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	**Flash memory error**
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
8110h	CAN overrun (objects lost)
8120h	CAN in error passive mode
8130h	Life guard error or heartbeat error
8140h	Recovered from bus off

8150h	CAN-ID collision
8200h	Protocol error - generic
8210h	PDO not processed due to length error
8220h	PDO length exceeded
8230h	DAM MPDO not processed, destination object not available
8240h	Unexpected SYNC data length
8250h	RPDO timeout
9000h	External error – generic error
F000h	Additional functions – generic error
FF00h	Device specific – generic error

7.11 Node guarding protocol

At system boot the "Node guarding protocol" is disabled; this protocol is enabled automatically as soon as the Master device sends a 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 a 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 a 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 – Set up

Here following are some examples of transmission between a Master and a Slave device.
A generic "ID" value is used to indicate the unit address; the Master address is always 0. All values are expressed in hexadecimal notation.

Set Operational, Pre-operational status

NMT message Master → Slave

	COB-ID	Cmd	Node
Operational:	000	01	ID
Pre-operational:	000	80	ID

Setting the 6005-01 Measuring step setting (eg. 1 mm = 1 000 000 nm = 000F 4240h)

Master → Slave (Set request)

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

Slave → Master (Set confirmation)

COB.ID	Cmd	Index		Sub	Data bytes			
580+ID	60	05	60	01	00	00	00	00

Setting the 6000-00 Operating parameters

(**Scaling function control**: enabled, **Code sequence**: standard, **Limit switch min.** / **Limit switch max.**: disabled)

Master → Slave (Set request)

COB.ID	Cmd	Index		Sub	Data bytes			
600+ID	2B	00	60	00	04	00	-	-

Slave → Master (Set confirmation)

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

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

Master → Slave (Set request)

COB.ID	Cmd	Index		Sub	Data bytes			
600+ID	23	03	60	00	E8	03	00	00

Slave → Master (Set confirmation)

COB.ID	Cmd	Index		Sub	Data bytes			
580+ID	60	03	60	00	00	00	00	00

Setting the SYNC counter 1801 PDO2 parameters sub 2 (n = 5 = 05h)

Master → Slave (Set request)

COB.ID	Cmd	Index		Sub	Data bytes			
600+ID	2F	01	18	02	05	-	-	-

Slave → Master (Set confirmation)

COB.ID	Cmd	Index		Sub	Data bytes			
580+ID	60	01	18	02	00	-	-	-

Disabling the SYNC mode 1801 PDO2 parameters sub 1

Read COB-ID used by PDO2:

Master → Slave (Req request)

COB.ID	Cmd	Index		Sub	Data bytes			
600+ID	40	01	18	01	-	-	-	-

Slave → Master (Req reply)

COB.ID	Cmd	Index		Sub	Data bytes			
580+ID	43	01	18	01	B0	B1	B2	B3

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	Data bytes			
600+ID	23	01	18	01	B0	B1	B2	B3

Slave → Master (Set confirmation)

COB.ID	Cmd	Index		Sub	Data bytes			
580+ID	60	01	18	01	00	00	00	00

Enabling the Cyclic mode

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

Master → Slave (Set request)

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

Slave → Master (Set confirmation)

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

Read COB-ID used by PDO1:

Master → Slave (Req request)

COB.ID	Cmd	Index		Sub	Data bytes			
600+ID	40	00	18	01	-	-	-	-

Slave → Master (Req reply)

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

Slave → Master (Set confirmation)

COB.ID	Cmd	Index		Sub	Data bytes			
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** and **Restore node** 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	0x00080196		
1001-00 Error register	0x00		
1003 Predefined error field	-		
1005-00 COB-ID SYNC message	0x00000080		
1008-00 Manufacturer Device name	SMAG-CB*		
1009-00 Manufacturer Hardware version	-		
100A-00 Manufacturer Software version	-		
100C-00 Guard time	0x0000		
100D-00 Life time factor	0x00		
1014-00 COB-ID EMCY	NODEID+0x00000080		
1015-00 Inhibit time EMCY	0x0000		
1018 Identity object			
1800 PDO1 parameters, sub 1	NODEID+0x40000180		
1800 PDO1 parameters, sub 2	0xFE		
1801 PDO2 parameters, sub 1	NODEID+0x40000280		
1801 PDO2 parameters, sub 2	0x01		
1802 PDO3 parameters, sub 1	NODEID+0xC0000380		
1802 PDO3 parameters, sub 2	0x01		
1A00-01 TPDO1 mapping parameter, sub 1	0x60040020		
1A01-01 TPDO2 mapping parameter, sub 1	0x60040020		
1A02-01 TPDO3 mapping parameter, sub 1	0x60080040		
2104-00 Limit switch min	0x00000010		
2105-00 Limit switch max	0x0001FFF0		
3000-00 Baud rate	0x05		
3001-00 Node-ID	0x01		
6000-00 Operating parameters	0x0000		
6001-00 Total measuring range	0x00020000		
6002-00 Total measuring range	0x00020000		
6003-00 Preset value	0x00000000		
6005-01 Measuring step setting	0x00001388		
6200-00 Cyclic time	0x0000		
6501-00 Measuring step	0x00001388		
6502-00 Number of distinguishable revolutions	0x0001		
6504-00 Supported alarms	0x0000		
6506-00 Supported warnings	0x0000		
6507-00 Profile and software version	0x03010101		
6508-00 Operating time	0xFFFFFFFF		
6509-00 Offset value	0x00000000		
650A-01 Manufacturer offset value, sub 1	0x00000000		
650B-00 Serial number	0xFFFFFFFF		

* Text value

This page intentionally left blank

This page intentionally left blank

Document release	Release date	Description	HW	SW	Installation file version
1.0	25.02.2013	First issue	1.0	1.1	V1
1.1	20.05.2015	General review, translation into Italian	1.0	1.1	V1
1.2	13.02.2017	General review, "3 - Mechanical installation" section updated	1.0	1.1	V1

Dispose separately

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

Tel. +39 0445 806600
Fax +39 0445 806699

info@lika.biz • www.lika.biz