# User's guide

# IXB, IXC

CANopen®

DSP 410 - Device profile for inclinometers


lika

Smart encoders & actuators

# General 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 Lika device and 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. |
|  | 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 provide the most complete information the operator needs to correctly and safely install and operate the **IXB and IXC series inclinometers with CANopen interface**.

Inclinometers are instruments designed to measure the levelling and pitch and roll angles for motion control or safety purposes.
IXB and IXC series inclinometers are equipped with CANopen interface (while IXA series inclinometers are equipped with analogue interface).
CANopen inclinometers are available with either 1-axis or 2-axis operation. IXB series inclinometers can be programmed for 1-axis or 2-axis operation; IXC1 inclinometers are 1-axis sensors; IXC2 inclinometers are 2-axis sensors.
The measurement range is up to ±360 deg in 1-axis models and from ±5 deg up to ±60 deg in 2-axis models.

To make it easier to read the text, this guide can be divided into two main sections.
In the first section general information concerning the safety, the mechanical installation and the electrical connection as well as tips for setting up and running properly and efficiently the unit are provided.
While in the second section, entitled **CANopen Interface**, both general and specific information is given on the CANopen interface. In this section the interface features and the CANopen 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 following 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 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, truckbased 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 includes 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 encoders/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 modules. |
| **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 SDO** | The Client SDO initiates the SDO communication by means of reading or writing to the Object Dictionary of the Server device. |
| **Client/Server communication** | In a Client/Server communication the Client initiates the communication with the Server. It is always a point-to-point communication. |
| **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 requires 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 that the acceptance filter is opened. |
| **D-sub connector** | Standardized connectors. Most common in use is the 9-pin Dsub 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 at anytime without notification. |
| **EDS checker** | Software tool that checks the conformity of electronic data sheets. The CANopen EDS checker is available for download on CiA's website. |
| **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 (heartbeat). |
| **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 bytes of data have to be transmitted. |
| **Flying Master** | In safety-critical applications, it may be required that a missing NMT Master is replaced 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 Sub-index. |
| **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 is 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 bytes including one control byte, three multiplexer bytes (containing the 24-bit Index and Sub-index), 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 may also 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 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, they 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 bytes. |
| **Remote transmission request (RTR)** | Bit in the arbitration field indicating whether 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 mesh 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 CANopen communication services 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 infinite. |
| **Server SDO** | The Server SDO receives the SDO messages from the corresponding SDO Client and responses each SDO message or 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 parts;
- device must be used only for the purpose appropriate to its design: use for purposes other than those for which it has been designed could result in serious personal and/or the environment damage;
- high current, voltage and moving mechanical parts can cause serious or fatal injury;
- warning ! Do not use in explosive or flammable areas;
- failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of design, manufacture, and intended use of the equipment;
- Lika Electronic assumes no liability for the customer's failure to comply with these requirements.

## 1.2 Electrical safety

- Turn OFF power supply before connecting the device;
- connect according to explanation in the "Electrical connections" section;
- wires of output signals which are not used must be cut at different lengths and insulated singularly;
- 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 shield and/or the connector housing and/or the frame to ground. Make sure that ground is not affected by noise. The connection point to ground can be situated both on the device side and on user's side. The best solution to minimize the interference must be carried out by the user.

### 1.3 Mechanical safety

- Install the device following strictly the information in the "Mechanical installation" section;
- mechanical installation has to be carried out with stationary mechanical parts;
- do not disassemble the unit;
- do not tool the unit;
- delicate electronic equipment: handle with care; do not subject the device and the shaft to knocks or shocks;
- respect the environmental characteristics of the product;
- always comply with the mounting positions indicated in the "Mechanical installation" section according to the model to be installed (1-axis or 2-axis inclinometer).

# 2 Identification

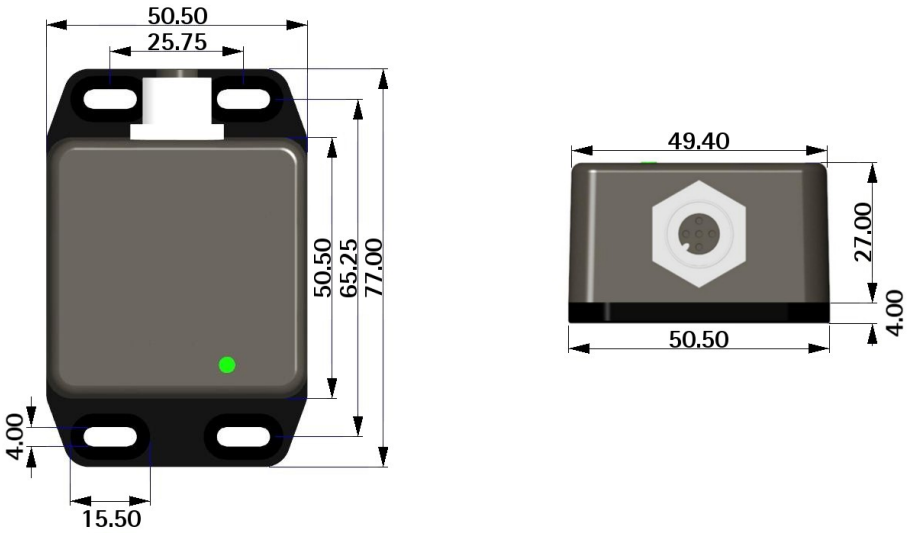
Device can be identified through the **order code** and the **serial number** printed on the label applied to its body. Information is listed in the delivery document too. Please always quote the order code and the serial number when reaching Lika Electronic for purchasing spare parts or needing assistance. 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 absolutely in stop.

## 3.1 Overall dimensions (Figure 1)




(values are expressed in mm)

**Figure 1 - IXB and IXC inclinometers – Overall dimensions**

### 3.2 Installation (Figure 2 and Figure 3)

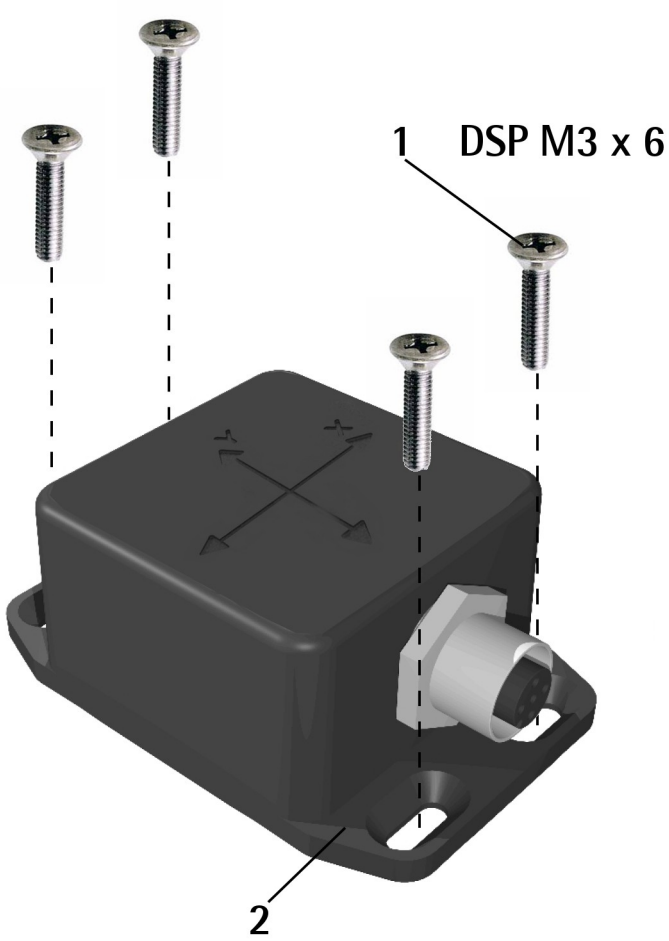



**Figure 2 - Installation**

Install the inclinometer as shown in Figure 2 and Figure 3:

- tighten the flange **2** using four M3 x 6 min. DSP screws **1**;
- max. tightening torque: **1.1 Nm**;
- 1-axis inclinometers must be mounted on a vertical plane; 0° position is achieved by mounting the connector on the left side (see Figure 3);
- 2-axis inclinometers must be mounted on an horizontal plane (see Figure 3);
- 2-axis inclinometers can be mounted also upside down.

**WARNING**
Always comply with the mounting positions indicated in Figure 3 according to the model to be installed or the selected operational mode (1-axis inclinometer or 2-axis inclinometer).

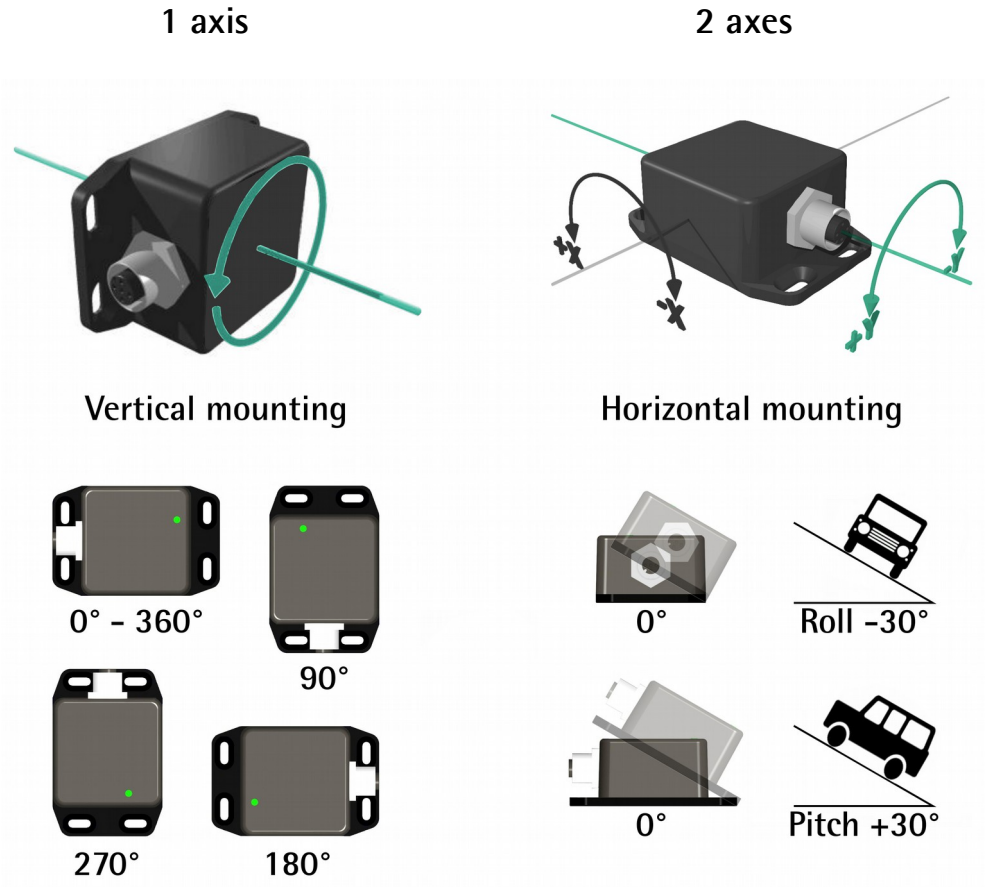
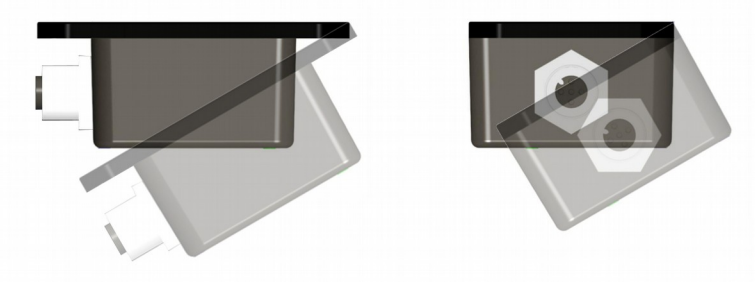



**Figure 3 - Mounting positions**

**NOTE**
2-axis inclinometers can be mounted also upside down as shown in the Figure.



# 4 Electrical connections

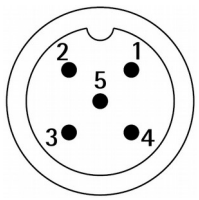
**WARNING**
Electrical connections must be carried out by qualified personnel only, with power supply disconnected and mechanical parts absolutely in stop.

## 4.1 Ground connection

Minimize noise by connecting the shield or the connector housing to ground. Make sure that ground is not affected by noise. The connection point to ground can be situated both on the device side and on user's side. The best solution to minimize the interference must be carried out by the user. You are advised to provide the ground connection as close as possible to the inclinometer.

## 4.2 M12 5-pin connector

M12 5-pin connector pin-out complies with CANopen® standard. Therefore you can use standard CAN cables commercially available.



M12 male
(frontal side)

| Description | M12 5-pin |
|---|---|
| Optional CAN Shield | 1 |
| +7Vdc +40Vdc supply voltage | 2 |
| 0Vdc supply voltage | 3 |
| CAN HIGH | 4 |
| CAN LOW | 5 |

### 4.3 Diagnostic LED (Figure 4)

A two-colour LED located in the top of the inclinometer's enclosure is meant to show visually the operating or fault status of both the CANopen interface and the device. The **green** LED is used as **RUN LED** indicator, while the **red** LED is used as **ERROR LED** indicator. The LED complies with CiA DR-303-3 specifications. The meaning of the LED configuration is explained in the following tables.

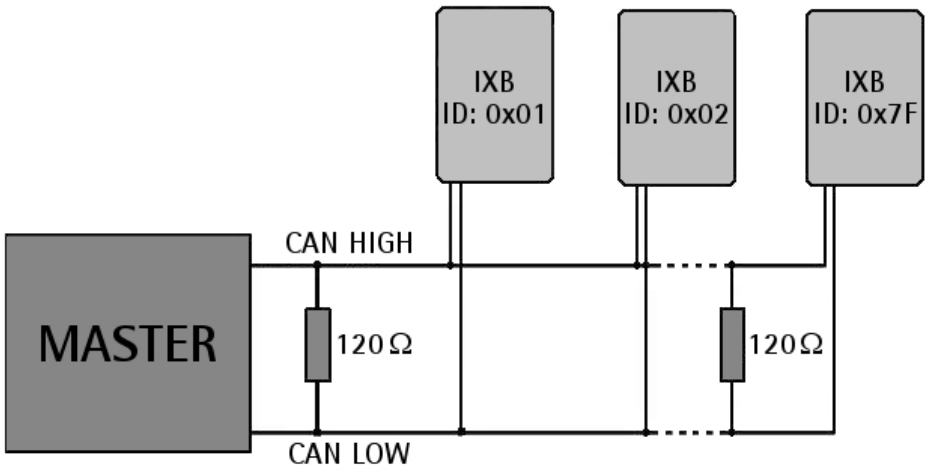
**Figure 4: Diagnostic LED**

| GREEN RUN LED | Description |
|---|---|
| OFF | The device is switched off |
| Single flash | The device is in **Stopped** state (see on page 30) |
| Blinking | The device is in **Pre-Operational** state (see on page 30) |
| ON | The device is in **Operational** state (see on page 30) |

| RED ERROR LED | Description |
|---|---|
| OFF | The device is in work condition, no error is active |
| Single flash | Max. number of warning errors |
| Double flash | Node guarding error (see on page 75) |
| ON | The CAN controller is switched off |

### 4.4 Bus termination resistor

IXB and IXC CANopen inclinometers are not equipped with internal bus termination resistor. The user must connect a **120 Ω terminator** between CAN HIGH and CAN LOW outputs if the inclinometer is either the first or the last device in the network, as shown -for instance- in the following Figure.




### 4.5 Setting the baud rate

The baud rate (transmission rate) can be set via software only. See the **2001-00 Baud rate** object on page 44.

### 4.6 Setting the node address

The node address (NODE ID) can be set via software only. See the **2000-00 Node-ID** object on page 43.

# 5 CANopen® interface

Lika inclinometers are always Slave devices in compliance with the "Device profile for inclinometer".
They comply with "CiA Draft Standard 301 4.2.0", that specifies the CANopen application layer and communication profile. This specification includes the data types, encoding rules and dictionary objects as well as the CANopen communication services network management (services and protocols). It also specifies the CANopen communication profile, e.g. the physical layer, the predefined communication object identifier connection set, and the content of the Emergency, Timestamp, and Sync communication objects. Thus IXB and IXC inclinometers are fully compliant with the specifications provided by the CiA DSP 410 device profile for 1-axis and 2-axis inclinometers with a maximum resolution of 16 bits.
For any omitted information, refer to the "CiA Draft Standard 301 4.2.0" and the "CiA 410 Device profile for inclinometer 1.3.0" documents available at the address **www.can-cia.org**.

## 5.1 EDS file

CANopen® inclinometers are supplied with their own EDS file **Lika_IXB_IXCx_CB_Vx_xxx.eds** (see at **www.lika.biz > OEM & custom products**). Vx shows the release version of the EDS file.
EDS file has to be installed in the CANopen® Master device.
**Lika_IXB_IXC1_CB_Vx_1axis.eds:** for 1-axis (IXC1) or 1-axis work mode (IXB) inclinometers.
**Lika_IXB_IXC2_CB_Vx_2axes.eds:** for 2-axis (IXC2) or 2-axis work mode (IXB) inclinometers.

### 5.2 Operational states

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

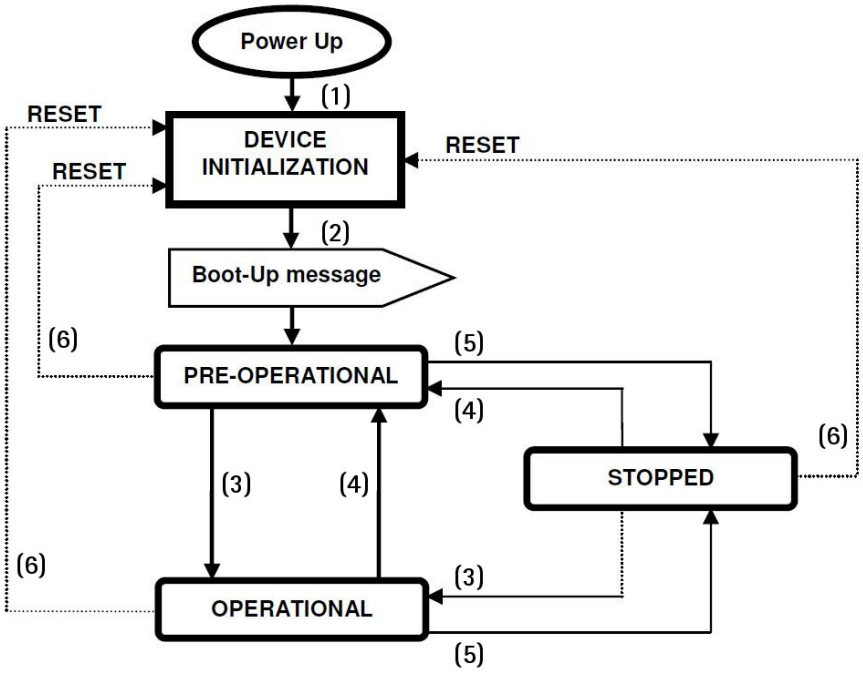



**Figure 5: Operational states**

| | |
|---|---|
| (1) | Power up |
| (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** |

#### 5.2.1 Boot-up message

When the inclinometer is switched on, the boot-up procedure is initialized. The device sends a *boot-up message* having the following frame structure, according to CANopen DS 301 v4.x:

| COB ID | Byte 0 |
|---|---|
| 700h + NODE ID | 00h |

**Table 1 –Boot-up message structure**

### 5.2.2 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 EEPROM, sends a boot-up message and then switches automatically to **Pre-Operational** state.

### 5.2.3 Pre-Operational state

In this state the communication between the Master and the Slave is possible using SDO messages. They allow working parameters to be set. The Slave cannot send PDO messages.
To switch the Slave device to the **Operational** state the Master must send a **Start remote node** command using an NMT message.

### 5.2.4 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" and is allowed to send process data using PDO messages. The "Object dictionary" can be accessed 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.

### 5.2.5 Stopped state

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

## 5.3 COB identifiers and communication objects

CANopen communications occur via CAN-frames. A CAN-frame or Communication Object is a command sent to/from the device. Its 11-bit identifier (called COB ID) is divided into the 4-bit function code and the 7-bit Node ID. At any hardware or software reset, COB IDs and the Node ID are loaded from the device's "Object dictionary" (see the "5.6 Object dictionary" section on page 34).

| Message | Direction | COB ID | | Object description |
|---|---|---|---|---|
| **NMT** | RX | 00h | | NMT services: operational, pre-operational, stop, reset |
| **SYNC** | RX | 80h | | Sync object |
| **EMCY** | TX | 80h | + Node ID | Emergency object |
| **TPDO1** | TX | 180h | + Node ID | Pitch, Roll and Internal Temperature |
| **SDO (Client → Server)** | RX | 600h | + Node ID | Access to the "Object dictionary", see on page 34 |
| **SDO (Server → Client)** | TX | 580h | + Node ID | Reply to SDO request |

**Table 2 - COB identifiers**

**NOTE**
The type of COB (TX or RX) is viewed from the Slave device.

**NOTE**
The Node ID default value is 0Ah (see on page 43).

### 5.4 Network management objects (NMT)

When the initialization is completed, the device enters the **Pre-Operational** state (see on page 30). The maximum time for the sensor initialization to be performed is 300 ms. A malfunction may be caused if any data frame is sent before the end of the initialization process: in this case a hardware reset of the sensor is required.
In order to start getting data, the inclinometer must be set to **Operational** state (see on page 30). The Figure 5 on page 29 shows the NMT state machine of a generic CANopen device.

NMT commands are used to change the machine state (e.g. to start and stop the device), as well as to detect remote device boot-ups and error conditions. The NMT frame structure used for NMT commands is shown below:

| COB ID | Byte0 | Byte1 |
|---|---|---|
| 0000h | COMMAND CODE | 00h ( broadcast ) or Node ID ( specific node ) |

**Table 3 – NMT frame structure**

Byte 0 must be replaced by one of the following command codes:

| COMMAND CODE | Description |
|---|---|
| 01h | Start remote node → Enter the node OPERATIONAL state |
| 02h | Stop remote node →Enter the node STOPPED state |
| 80h | Enter the node PRE–OPERATIONAL state |
| 81h | Reset Node |
| 82h | Reset Communications |

**Table 4 – Command codes**

NMT command frames can be broadcast to all network nodes or sent to a specific node. The address must be written at Byte 1: 00hex for the broadcast transmission, Node ID for a specific node transmission.

### 5.5 Transmit Process Data Object (TPDO1) frame organization

The Process Data Object protocol is used to process real-time data and send information about current inclinations on both axes. TPDO1 frame organization is reported below. TPDO1 frames have different structures depending on the operational mode. The frame structure for **dual axis mode** is shown below:

| COB ID | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
|---|---|---|---|---|---|---|---|---|
| 180h + NODE ID | Longitudinal inclination value (LSB first) | | Lateral inclination value (LSB first) | | Internal temperature of the device | Not used | | |

**Table 5 - TPDO1 frame structure for 2-axis mode**

Received data are in hexadecimal format and must be converted depending on the set resolution, stored on the **6000-00 Resolution** index of the "Object dictionary" (see on page 55).

Measured inclination values are also stored on the "Object dictionary" according to CiA DSP 410: the longitudinal value (X axis) can be found at **6010-00 Measured X axis value** index, the lateral value (Y axis) can be found at **6020-00 Measured Y axis value** index. The registers store the last measured angle values in a two's complement fixed-point 16-bit number. As the inclination values, also the device internal temperature (stored on the **5000-00 Device internal temperature [°C]** object) is transmitted, in a two's complement 8-bit number.

**EXAMPLE**

X-axis value at **6010-00 Measured X axis value** index is "F3B1hex". For the two's complement conversion, the number is equal to -3151 dec. The resolution stored on the **6000-00 Resolution** object and expressed in thousandths of a degree is 0Ah = 10 dec, therefore the resolution is 0.01 degrees. The current angle is then calculated as -3151/100 = -31.51 degrees.

The frame structure for **single axis mode** is similar, but only one field is used for the inclination value.

| COB ID | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
|---|---|---|---|---|---|---|---|---|
| 180h + NODE ID | Inclination value (LSB first) | | Internal temperature of the device | Not used | | | | |

**Table 6 - TPDO1 frame structure for 1-axis mode**

According to CiA DSP 410, the inclination value is mapped at **6010-00 Measured X axis value** index, that is in the field used for X-axis in the dual axis mode. This register stores the last measured angle value in a two's complement fixed-point 16-bit number. Together with the inclination value, the device internal temperature (**5000-00 Device internal temperature [°C]**) is also transmitted as a two's complement 8-bit number.

## 5.6 Object dictionary

The "Object dictionary" contains all settings concerning the device and the communication procedures.

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

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

or

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

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

Unsigned8 / Integer8 data type:

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

Unsigned16 / Interger16 data type:

| Process data bytes | |
|---|---|
| byte 4 | byte 5 |
| bit 7 ... 0 | bit 15 ... 8 |
| LSByte | MSByte |

Unsigned32 data type:

| Process data bytes | | | |
|---|---|---|---|
| byte 4 | byte 5 | byte 6 | byte 7 |
| bit 7 ... 0 | bit 15 ... 8 | bit 23 ... 16 | bit 31 ... 24 |
| LSByte | ... | ... | MSByte |

### 5.6.1 Standard objects (DS 301)

#### 1000-00 Device type

[Unsigned32, ro]

Default = 0001 019Ah = 1-axis inclinometer with max. resolution of 16 bits according to CiA device profile for inclinometers (CiA DSP 410)
0002 019Ah = 2-axis inclinometer with max. resolution of 16 bits according to CiA device profile for inclinometers (CiA DSP 410)

Refer also to **4001-00 Sensor operational mode 2 axes / 1 axis** object on page 52.

#### 1001-00 Error register

[Unsigned8, ro]

Should an error occur, the bit 0 of this object is set to "1". For complete information on the **1001-00 Error register** field structure please refer to the "6.1 Error register" section on page 70.

Default = 00h

#### 1002-00 Manufacturer error register

[Unsigned32, ro]

The **1002-00 Manufacturer error register** shows the recent state of all detectable errors. For complete information on the **1002-00 Manufacturer error register** field structure please refer to the "6.2 Manufacturer error register" section on page 70.

Default = 0000 0000h

#### 1003 Predefined error field

This object contains the last four errors which generated an emergency message. For complete information on the **1003 Predefined error field** structure please refer to the "6.3 Pre-Defined error field" section on page 71.

- **00 Number of current errors** [Unsigned32, rw]
  If no error is present the value of sub-index 00h is 00h and a read access to sub-index 01h is responded with an SDO abort message (abort code: **0800 0024h** or **0800 0000h**).
  Write 00h to delete the error history.
- **01 Last error occurred** [Unsigned32, ro]
- **02 ... 04 Previous errors occurred** [Unsigned32, ro]
- **05 Oldest error occurred** [Unsigned32, ro]

### 1005-00 COB_ID SYNC message

[Unsigned32, rw]
For complete information on the synchronous transmissions and the SYNC frames please refer to the "5.10.1 Synchronous transmissions and SYNC frames" section on page 65.
Follow the saving procedure described in the "5.7.4 SAVE ALL procedure and reset commands" section on page 64 to validate modifications.
Default = 0000 0080h (min. = 1, max. = 2047)

### 100A-00 Manufacturer software version

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

### 100C-00 Guard time

[Unsigned16, rw]
It contains the Guard time expressed in msec (milliseconds).
The **100C-00 Guard time** object is used in the "Node guarding protocol" controlled by the Master. For more details see the "6.5.2 Node guarding and Life guarding" section on page 75.
Default = 0000h (min. = 0, max. = 65535)

### 100D-00 Life time factor

[Unsigned8, rw]
The **100D-00 Life time factor** object is used in the "Node guarding protocol" controlled by the Master. For more details see the "6.5.2 Node guarding and Life guarding" section on page 75.
Default = 00h (min. = 0, max. = 255)

## 1010-01 Store parameters

[Unsigned32, rw]
Use this object to save all parameters on non-volatile memory.
Write "**save**" (ASCII code in hexadecimal format) in the data bytes:

Master → Inclinometer

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

Inclinometer → Master (confirmation)

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

For complete information on the saving procedure please refer to the "5.7.4 SAVE ALL procedure and reset commands" section on page 64.

## 1011-01 Restore default parameters

[Unsigned32, rw]
This object allows the operator to restore all parameters to default values (default values are set at the factory by Lika Electronic engineers to allow the operator to run the device for standard operation in a safe mode).
Write "**load**" (ASCII code in hexadecimal format) in the data bytes and then issue a **Reset node** command:

Master → Inclinometer

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

Inclinometer → Master (confirmation)

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

Master → Inclinometer (**Reset node**)

| COB ID | Cmd | Slave ID |
|---|---|---|
| 000 | 81 | ID |

Inclinometer → Master (Boot-up)

| COB ID | Cmd |
|---|---|
| 700+ID | 00 |

For complete information on the restoring procedure please refer to the "5.7.5 Restoring all parameters" section on page 64.

**NOTE**
Save default values 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). For complete information on the emergency frames please refer to the "6.4 Emergency frames (EMCY)" section on page 72.
Default = 0000 0080h + Node ID

### 1015-00 Inhibit time EMCY

[Unsigned16, rw]
Inhibit time of the emergency messages (EMCY) expressed in multiples of 100 µs. When set to 0, this function is disabled. For complete information on the emergency frames please refer to the "6.4 Emergency frames (EMCY)" section on page 72.
Default = 0000h (min. = 0, max. = 65535)

### 1017-00 Producer heartbeat time

[Unsigned16, rw]
The producer heartbeat time indicates the configured cycle time of the heartbeat. The value is expressed in multiples of 1 ms. When set to 0, this function is disabled. The *heartbeat* is a failure monitoring mechanism that is managed by the CAN Slave (IXB/IXC device). For complete information on the heartbeat please refer to the "6.5.1 Heartbeat" section on page 75.
Default = 0000h (min. = 0, max. = 65535)

## 1018 Identity object

- **01 Vendor-ID** [Unsigned32, ro]
  Default = 0000 012Eh
- **02 Product code** [Unsigned32, ro]
  Default = 0041 4333h (ASCII code: AC3)
- **03 Revision number** [Unsigned32, ro]
  Default = 0000 0041h (ASCII code: A)
- **04 Serial number** [Unsigned32, ro]
  Default = device dependent

## 1200 SDO1 Server parameter

This parameter is meant to describe the SDOs (Service Data Objects) used in the CANopen device. SDO messages are used to set and read values from the "Object dictionary" of the inclinometer. These parameters are described in the "Object dictionary" section, see on page 34. For complete information on SDOs please refer to the "5.7 SDO commands" section on page 60.

- **01 COB-ID Client > Server** [Unsigned32, ro]
  Default = 0000 0600h + Node ID
- **02 COB-ID Server > Client** [Unsigned32, ro]
  Default = 0000 0580h + Node ID

| Bit(s) | Value | Description |
|---|---|---|
| **31 (MSB) valid** | 0 | SDO exists / is valid |
| | 1 | SDO does not exist / is not valid |
| **30 dyn** | 0 | Value is assigned statically |
| | 1 | Value is assigned dynamically |
| **29 frame** | 0 | 11-bit CAN-ID valid (CAN base frame) |
| | 1 | 29-bit CAN-ID valid (CAN extended frame) |
| **28 ... 11 29-bit CAN-ID** | X | 29-bit CAN-ID of the CAN extended frame |
| **10 ... 0 (LSB) 11-bit CAN-ID** | X | 11-bit CAN-ID of the CAN base frame |

## 1800 Transmit PDO1 communication parameters

This object contains the communication parameters for the PDOs the CANopen device is able to transmit. IXB/IXC inclinometer implements different types of transmission, which can be set at this index **1800 Transmit PDO1 communication parameters** sub-index 02h. Values from 0 dec to 240 dec (0h

to F0h) are used for synchronous transmissions, 253 dec for the RTR transmission, and 254 dec for asynchronous types. For complete information on PDOs please refer to the "5.10 TPDO1 transmission type" section on page 65.

- **01 COB-ID of TPDO1** [Unsigned32, ro]

| Bit number | Value | Meaning |
|---|---|---|
| **31 (MSB) valid** | 0 | PDO exists / is valid |
| | 1 | PDO does not exist / is not valid |
| **30 RTR** | 0 | RTR allowed on this PDO |
| | 1 | no RTR allowed on this PDO |
| **29 frame** | 0 | 11-bit CAN-ID valid (CAN base frame) |
| | 1 | 29-bit CAN-ID valid (CAN extended frame) |
| **28 ... 11 29-bit CAN-ID** | X | 29-bit CAN-ID of the CAN extended frame |
| **10 ... 0 (LSB) 11-bit CAN-ID** | X | 11-bit CAN-ID of the CAN base frame |

Default = 0000 0180h + Node ID

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

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

Default = FEh (asynchronous transmission, manufacturer specific)

- **03 Inhibit time** [Unsigned16, rw]
  Minimum interval for PDO transmission if the transmission type is set to FEh or FFh. The value is expressed in multiples of 100 µs. The value 0 disables the inhibit time. The value is changed while the PDO exists (bit

31 of sub-index 01h is set to 0). For more information please refer to the "5.10.4 Inhibit timer" section on page 67.
Default = 0000h (min. 0, max. 65535)

- **04** [Unsigned8, rw]
  Reserved

- **05 Event timer** [Unsigned16, rw]
  This bit contains the event timer. It is the maximum interval for PDO transmission if the transmission type is set to FEh or FFh. The value is expressed in multiples of 1 ms. The value 0 disables the event timer. For more information please refer to the "5.10.3.2 Transmission on event timer" section on page 67.
  Default = 0000h (min. 0, max. 65535)

### 1A00 Transmit PDO1 mapping parameter

This object contains the mapped position value of the inclinometer according to the DSP 410 device profile specifications.

- **01 Inclination value X-axis parameters** [Unsigned16, ro]
  The first part refers to the register index where the last measured value on X-axis is stored (**6010-00 Measured X axis value**);
  The second part is the value resolution stored on the **6000-00 Resolution** index.
  Default = 6010 0010h

- **02 Inclination value Y-axis parameters** [Unsigned16, ro]
  The first part refers to the register index where the last measured value on Y-axis is stored (**6020-00 Measured Y axis value**);
  The second part is the value resolution stored on the **6000-00 Resolution** index.
  Default = 6020 0010h

**NOTE**
This is not available for IXB model when the single axis mode is set (see the **4001-00 Sensor operational mode 2 axes / 1 axis** object) and for IXC1 model.

- **03 Device internal temperature** [Unsigned16, ro]
  The first part refers to the register index where the last measured temperature value expressed in °C is stored (**5000-00 Device internal temperature [°C]** object);

The second part is the surveillance stored on the **5001 Surveillance of the device's internal temperature** index.
Default = 5000 0008h

### 5.6.2 Manufacturer specific objects

#### 2000-00 Node-ID

[Unsigned8, rw]
This object defines the node identifier of the device. Allowed node addresses are between 1 and 127. The default value is 0Ah (= 10 dec).

The NODE ID is effective only after the *SAVE ALL* procedure (see the "5.7.4 SAVE ALL procedure and reset commands" section on page 64) and the node reset. After the reset, the COB IDs are recomputed according to the pre-defined connection set objects.

The **2000-00 Node-ID** object is not restored by the *Restore all parameters* command set at **1011-01 Restore default parameters** and has to be changed manually.

To change the Node ID value you have to:
- set the **2000-00 Node-ID** object;
- send a **Reset node** command;
- save the parameter.

Default = 0Ah (min. 01, max. 7F)

Master → Inclinometer

| COB ID | Cmd | Index | | Sub | Data byte |
|---|---|---|---|---|---|
| 600+ID | 2F | 00 | 20 | 00 | new Node ID |

Inclinometer → Master (confirmation)

| COB ID | Cmd | Index | | Sub | Data byte |
|---|---|---|---|---|---|
| 580+ID | 60 | 00 | 20 | 00 | 00 |

Master → Inclinometer (**Reset node**)

| COB ID | Cmd | Slave ID |
|---|---|---|
| 000 | 81 | old ID |

Inclinometer → 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, the parameters not saved are lost.

## 2001-00 Baud rate

[Unsigned16, rw]
This object is meant to set the baud rate (transmission rate).
IXB/IXC inclinometer is designed to support different CAN baud rates, referring to the CANopen Draft Standard 301. The supported baud rates are reported in the following table. The exact value must be written in hexadecimal format next to this index **2001-00 Baud rate**. The default value is 500 kbit/s.

Supported baud rates expressed in kbit/s are:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 10[1] | 20 | 50 | 125 | 250 | **500** | 800 | 1000 |

**Table 7 - Supported baud rates**

The baud rate is effective only after the *SAVE ALL* procedure (see the "5.7.4 SAVE ALL procedure and reset commands" section on page 64) and the node reset. After the reset, the COB IDs are recomputed according to the pre-defined connection set objects.

The **2001-00 Baud rate** object is not restored by the *Restore all parameters* command set through the **1011-01 Restore default parameters** object and has to be changed manually.

To change the baud rate value you have to:

- set the **2001-00 Baud rate** object;
- send a **Reset node** command;
- save the parameter;
- set the Master to the new baud rate.

Default = 01F4h (min. 000A, max. 03E8)

Master → Inclinometer

| COB ID | Cmd | Index | | Sub | Data byte |
|---|---|---|---|---|---|
| 600+ID | 2F | 01 | 20 | 00 | see table |

Inclinometer → Master (confirmation)

| COB ID | Cmd | Index | | Sub | Data byte |
|---|---|---|---|---|---|
| 580+ID | 60 | 01 | 20 | 00 | 00 |

Master → Inclinometer (**Reset node**)

| COB ID | Cmd | Slave ID |
|---|---|---|
| 000 | 81 | ID |

1 Please check that your CAN bus supports a baud rate = 10kbps before setting it on IXB/IXC.

Set the Master device to the new baud rate:

Inclinometer → 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, the parameters not saved are lost.

### 3000-00 Filtered samples

[Unsigned16, rw]
Default = 1000 (min. 1, max. 1000)

When vibrations affect the application, noise reduction is necessary in order to get stable measure and correct data. IXB/IXC inclinometers are equipped with two filters that reduce the environmental noise effects.
The first filter is an active 2nd order analogue filter, with a cut-off frequency of 20 Hz (fixed by hardware layout); the second is a software filter, that implements a moving average on acquired data.
The average is programmable by the user, he can set the number of averaged samples next to this index **3000-00 Filtered samples**. The default value (03E8h) corresponds to the maximum number (1000 samples). If set to 01h, the filter is deactivated. The value is valid for both one-axis and two-axis modes. The *SAVE ALL* procedure (see the "5.7.4 SAVE ALL procedure and reset commands" section on page 64) is necessary to validate modifications.
The optimal number of samples to be averaged depends on the application and is a trade-off between measure stability and sensor response time: the higher the number of averaged samples, the lower the response time yet the measure is more stable.
The filter frequency response is strictly related to the number of averaged samples and the sample rate, which is **fixed to 550 samples per second**. Sample rate is guaranteed in normal work conditions, when transmission commands (TPDO1) and SAVE ALL commands are not too frequent.
The formula of filter frequency response is reported below.

$$H[f] = \frac{\sin(\pi \times f \times M)}{M \times \sin(\pi \times f)}$$

**Figure 6: Filter frequency response**

The plot of the filter frequency response is a sync graph in the normalized frequency domain (see the Figure below).

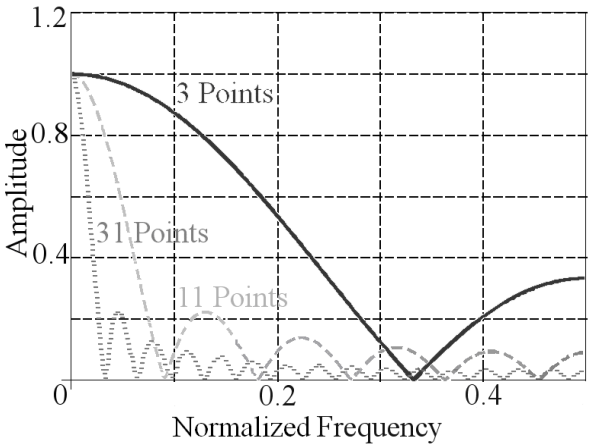



**Figure 7: Software filter response in frequency domain**

The normalized frequency domain runs from 0 to 0.5 rad/sample. For a normalized frequency of 0 rad/sample, the frequency response H[f] is equal to 1. To convert a specific vibration/noise frequency (expressed in Hz) into a normalized frequency, the formula reported below is used:

$$f_{NORM} = \frac{f\left[\frac{cycles}{\sec}\right]}{S_R\left[\frac{Samples}{\sec}\right]} = \left[\frac{cycles}{Samples}\right] = \left[\frac{radians}{Samples}\right]$$

$f\left[\frac{cycles}{\sec}\right]$ → Noise frequency expressed in Hz

$S_R\left[\frac{Samples}{\sec}\right]$ → Sampling rate (constant at 550 samples/s for IXB/IXC)

→ Normalized frequency

The filter frequency response can be plotted for different numbers of filtered samples.

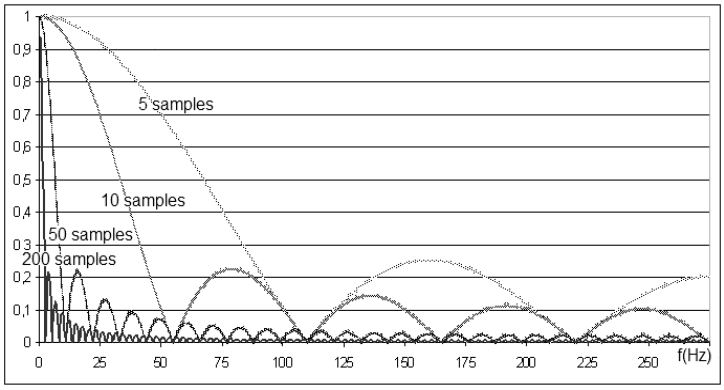



**Figure 8: Filter frequency response with different numbers of filtered samples**

In Figure 8, the Y-axis indicates the response gain, while the X-axis shows the frequency in Hz. Different noise frequencies can be cut by correctly choosing a specific number of samples. The number of filtered samples must be set as a trade-off between response stability and response time. The higher the filter, the shorter the time to wait for a correct measure. The graph below reports the response time, calculated as 5**τ**, versus the number of filtered samples.

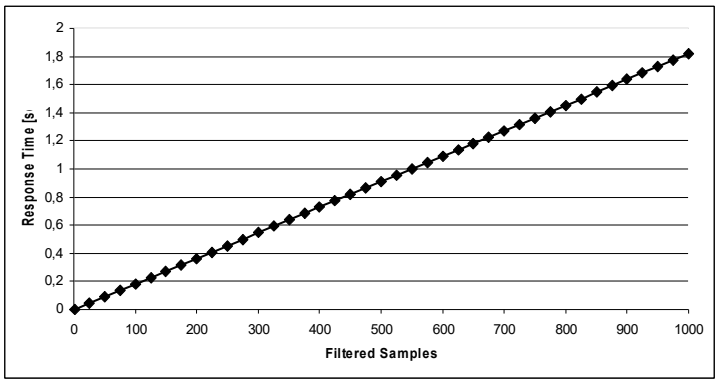



**Figure 9: Response time vs filtered samples**

The response time (or damping time) is the ratio between the number of averaged samples and the sampling time. The sampling time is the inverse ratio of the sampling rate.

$\#samples = \frac{Td}{Ts}$, where *Td* is the damping time and *Ts* is the sampling time (fixed at 1.82 ms).

Referring to the graph reported in Figure 9, if a damping time of 1 second is required, the number of average samples to be set is: 1000/1.82 = 549 samples. The graph below shows the step response in the time domain.

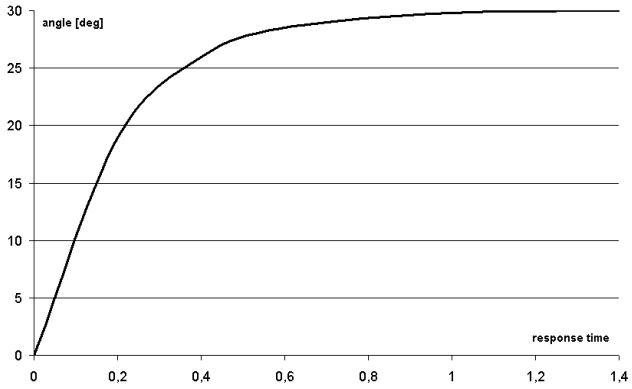



**Figure 10: Step response in time domain with 550 samples filter**

**EXAMPLE**

The following example shows a simple procedure useful to set the filter properly, depending on the final application.

- Selection of the maximum number of filtered samples. Supposing that 50 ms is the maximum acceptable response time, the maximum number of samples is equal to 50 ms / 1.82 ms = 27 samples. The filter must be set to less than 27 samples.
- Now let us suppose that the main noise frequency due to vibration is 50 Hz. At this frequency the characteristic H[f] is null for multiples of 11 samples. Following the condition found at point 1, quantities of 11 or 22 samples are suggested: the user can choose between 11 samples for a better response time or 22 samples for a better band-pass filter.
- Some practical tests and a comparison to check the theoretical calculations are recommended.

## 3001 TPDO1 transmission at inclination change

- **01 Enable/Disable (1/0) TX at inclination change** [Unsigned8, rw]
  Default = 0 (min. 0, max. 1)

- **02 Minimum inclination change for X axis** [Unsigned16, rw]
  Default = 100 (min., max. resolution dependent)

- **03 Minimum inclination change for Y axis** [Unsigned16, rw]
  Default = 100 (min., max. resolution dependent)

The transmission at inclination change is enabled by setting to "01h" the value at index **3001 TPDO1 transmission at inclination change** sub-index 01h. TPDO1 is transmitted as the inclination changes of at least the value stored on the register **3001 TPDO1 transmission at inclination change** sub-index 02h for X axis and sub-index 03h for Y axis, depending on the resolution set next to the **6000-00 Resolution** index. TPDO1 transmission at inclination change occurs only if the node is in **Operational** state. In order to prevent CAN bus flooding, the inhibit timer is suggested to be activated (**1800 Transmit PDO1 communication parameters** sub-index 03h) and the event timer deactivated (**1800 Transmit PDO1 communication parameters** sub-index 05h).

**EXAMPLE**

If the resolution is set to 1 degree (**6000-00 Resolution** object value =1000 dec = 3E8h) and the minimum inclination value at index **3001 TPDO1 transmission at inclination change** sub-index 02h is set to "07h", TPDO1s are transmitted every time the changes in the X axis inclination are greater than 7 degrees.

Transmission at inclination change is also active for single axis [0-360 deg] mode. In this case only the sub-index value for X-axis is active, while the sub-index of the Y-axis is disabled. The maximum available resolution for the single axis mode is expressed in hundredths of a degree. As the resolution is changed, the transmission is deactivated in order to prevent bus overflows, and must be reactivated at the end of the resolution setting.

**EXAMPLE**

The resolution (**6000-00 Resolution** object) is set to 1 degree. The minimum inclination change for X axis is set to "0Ah". So TPDO1s are transmitted only if changes greater than 10 degrees occur. Let us modify the resolution to 0.001 degrees. If the TPDO1 had not been automatically deactivated, transmissions at changes of 0.01 degrees would have happened, leading to the CAN bus overflow. To prevent CAN bus overflows, the setting of inhibit timer is suggested (see the "5.10.4 Inhibit timer" section on page 67).

For more information please also refer to the "5.10 TPDO1 transmission type" section on page 65.

### 3002-00 Single axis data format

[Unsigned8, rw]
This object allows the user to set the single axis data format, so it is only valid for IXB and IXC1 models.
It can be as follows:

- 0 = [0;360]°
- 1 = ±180°

If set to "0", data transmitted via TPDO1 and stored on **6010-00 Measured X axis value** index is in the range 0 ... 360 deg.
If set to "1", the range is between -180 deg and +180 deg.
Refer also to the "Single-axis mode" section on page 57.
Default = 0 (min. 0, max. 1)

**EXAMPLE**
Let us consider the IXC1 model with a 0.01 deg resolution. The **3002-00 Single axis data format** object is set to "0" and the value transmitted by the TPDO1 is 80E0h = 32992, that means 329.92 deg; in relation to the range 0 ... 360 deg, it means a negative inclination of about 30 deg. Set the **3002-00 Single axis data format** object to value "1" and save all data. After the reset the value transmitted by the TPDO1 is F440h = -3008 dec, that means -30.08 deg.

### 4000 Pitch and Roll value range

- **01 X range** [Unsigned16, rw]
  Default = 30 000 (min., max. resolution dependent)

- **02 Y range** [Unsigned16, rw]
  Default = 30 000 (min., max. resolution dependent)

- **03 Enable/Disable (1/0) user range** [Unsigned8, rw]
  Default = 0 (min. 0, max. 1)

This feature is only valid for two-axis mode of measurement on IXB and IXC2 models. Registers at index **4000 Pitch and Roll value range** allow the user to fix the operative ranges for X and Y axes. The values stored on sub-indexes 01h and 02h are unsigned and their resolution is stored on **6000-00 Resolution** index. The feature works only for the dual-axis operational mode and can be enabled at sub-index 03h (write the value 01h). When one of the two set thresholds is exceeded, an *EMCY frame* relative to the axis affected by the error is sent. If disabled, X and Y range values are kept equal to the absolute operative range of ±60 deg.

User-defined ranges are considered symmetric to the sensor relative zero that is obtained by adding the offset value to the absolute zero. The ranges added to the relative zero (initial offset) must be lower than the sensor absolute operative range (±60 deg): if the measured angle value is greater than +60 deg or lower than -60 deg, it is automatically trimmed (see the example below).

**EXAMPLE**

1. Set the offset for X axis (**6013-00 X axis offset value** object) = -45 deg
2. Set the offset for Y axis (**6023-00 Y axis offset value** object) = +15 deg
3. Set the range for X axis = ±30 deg
4. Set the range for Y axis = ±15 deg

**Figure 11: Graph that describes the example: values in red refer to X axis, values in black to Y axis**

In this case the X-axis relative zero is equal to the +45 deg absolute angle. In order to avoid *EMCY frames*, the inclination on X axis must be in the range:

X MAX = [60 deg – 45 deg] = +15 deg (+60 deg absolute angle)
X MIN = -30 deg (+15 deg absolute angle)

The maximum measurable value on X-axis is +15 deg instead of +30 deg because of the actual sensor absolute operative range.
The Y-axis relative zero is equal to the -15 deg absolute Y angle. In order to avoid *EMCY frames*, the inclination on Y axis must be in the range:

Y MAX = +15 deg (0 deg absolute angle)
Y MIN = - 15 deg (-30 deg absolute angle)

X-axis range is asymmetric to the measured zero while Y-axis range is symmetric, as it is included in the absolute range of ±60 deg. Pitch and roll ranges depend on the resolution set next to the **6000-00 Resolution** index: if it is greater than 0.001 deg, ranges can be specified between +/-5 deg and +/-60 deg. If the resolution is set to 0.001 deg, ranges can only be set up to +/-30 deg.
The mentioned feature can be exploited even in order to use the IXB/IXC as an "*ON/OFF*" device: if the event timer is deactivated (index **1800 Transmit PDO1 communication parameters**, sub-index 05h, value=0h) and the device is set in

Operational mode, the CAN Master will receive EMCY frames only when thresholds are exceeded.

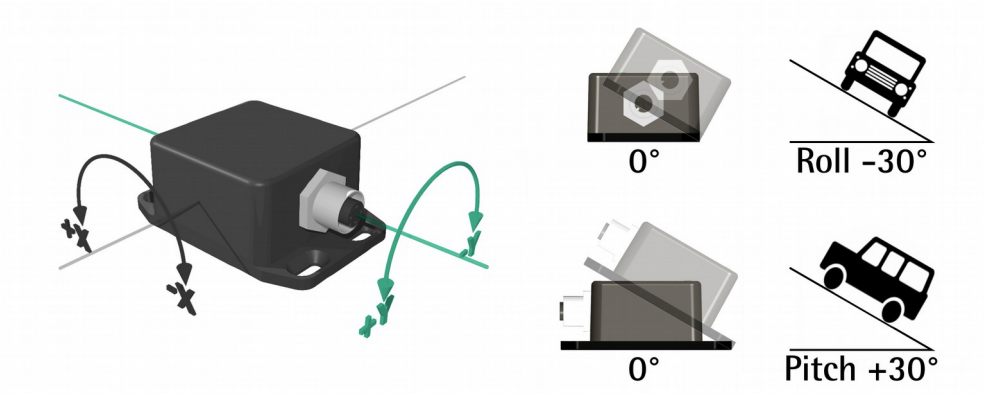
### 4001-00 Sensor operational mode 2 axes / 1 axis

[Unsigned8, rw]
IXB version can be configured for 1-axis (AAh) or 2-axis (00h) operational mode by setting this **4001-00 Sensor operational mode 2 axes / 1 axis** object. IXC1 and IXC2 have fixed operational mode, they are 1-axis mode and 2-axis mode respectively.
IXB and IXC1 are at zero degrees when the connector is kept on the left side, see the Table 8 here below.

| | | | |
|---|---|---|---|
|  |  |  |  |
| 0/360 deg | 90 deg | 180 deg | 270 deg |

**Table 8 – IXB/IXC1 positions for single axis measurements**

In dual-axis mode (IXB and IXC2), please follow the arrows indicated on the top of the device's enclosure: by lowering the inclinometer along either arrow, the measured value increases following the signs indicated on the label.




Default = 00h

**EXAMPLE**
The following example shows the procedure to switch from 2-axis mode to 1-axis mode.

- Write the value "AAh" next to the index **4001-00 Sensor operational mode 2 axes / 1 axis**.
- Send the **1010-01 Store parameters** SDO command.
- Reset the device.

Only after resetting the IXB inclinometer, the device type object value (**1000-00 Device type**) changes from 0002 019Ah (2 axes with max. resolution of 16 bits) to 0001 019Ah (1 axis with max. resolution of 16 bits) according to CiA device profile for inclinometers (CiA DSP 410).
When the device is set for single axis mode, **all the objects related to the Y-axis are disabled and cannot be accessed for reading or writing procedures**, otherwise an *SDO abort command* is triggered (**0602 0000h**).
The **4001-00 Sensor operational mode 2 axes / 1 axis** object is deactivated for IXC1 and IXC2 versions.

## 5000-00 Device internal temperature [°C]

[Integer8, ro]
The device internal temperature is transmitted by the TPDO1 when the node is in **Operational** state (see the **1A00 Transmit PDO1 mapping parameter** object). It is available as a two's complement 8-bit number. For further information please refer also to the "5.5 Transmit Process Data Object (TPDO1) frame organization" section on page 32.

## 5001 Surveillance of the device's internal temperature

- **01 Enable / disable (1/0) temperature surveillance** [Unsigned8, rw]
  Default = 0 (min. 0, max. 1)

- **02 Lower temperature limit [°C]** [Integer8, rw]
  Default = -30 (min. -55, max. +120)

- **03 Upper temperature limit [°C]** [Integer8, rw]
  Default = +75 (min. -55, max. +120)

A surveillance of the device internal temperature can be enabled next to this index **5001 Surveillance of the device's internal temperature** sub-index 01h (value = 01h to enable, value = 00h to disable the surveillance operation). The lower and upper temperature thresholds are set in the sub-indexes 02h and 03h

as two's complement 8-bit numbers. The resolution is expressed in units of Celsius degrees. As soon as the internal temperature exceeds the set range limits, an *EMCY frame* is transmitted and the three error registers are thus updated (refer to the "Errors" section on page 70). The control of the internal temperature is done every second. Note that the two thresholds can range between -55°C and +120°C but the maximum operating range of the sensor is -40°C to +85°C. Default values are -30°C for the lower temperature limit and +75°C for the upper temperature limit.

### 5544-00

Reserved for calibration and debug values.

### 5555-00

Reserved for calibration and debug values.

### 5.6.3 Device profile objects (DSP 410)

#### 6000-00 Resolution

[Unsigned16, rw]
The resolution of the inclinometer is expressed in thousandths of a degree. Available values are:

| Resolution | | Object value | |
|---|---|---|---|
| Entire degree | [1 deg] | 1000 dec | 03E8h |
| Tenths of a degree | [1 deg / 10] | 100 dec | 0064h |
| Hundredths of a degree | [1 deg / 100] | 10 dec | 000Ah |
| Thousandths of a degree | [1 deg / 1000] | 1 dec | 0001h |

**Table 9 - Available resolutions**

Angle data are 16-bit numbers in a two's complement format that limit the measurable range to ±30 degrees when the resolution of thousandths of a degree is selected. The whole angle range of ±60 degrees is measurable with the other three configurations instead. Default resolution is "thousandths of a degree".
Default = 1 (min. 1, max. 1000)

#### 6010-00 Measured X axis value

[Integer16, ro]
For complete information please refer to the "Setting the X-axis objects: 6010-00 Measured X axis value, 6011-00 Inversion of X axis range, 6012-00 X axis preset value and 6013-00 X axis offset value" section on page 56.

#### 6011-00 Inversion of X axis range

[Unsigned8, rw]
For complete information please refer to the "Setting the X-axis objects: 6010-00 Measured X axis value, 6011-00 Inversion of X axis range, 6012-00 X axis preset value and 6013-00 X axis offset value" section on page 56.
Default = 0 (min. 0, max. 1)

#### 6012-00 X axis preset value

[Integer16, rw]
For complete information please refer to the "Setting the X-axis objects: 6010-00 Measured X axis value, 6011-00 Inversion of X axis range, 6012-00 X axis preset value and 6013-00 X axis offset value" section on page 56.
Default = 0

### 6013-00 X axis offset value

[Integer16, rw]
For complete information please refer to the "Setting the X-axis objects: 6010-00 Measured X axis value, 6011-00 Inversion of X axis range, 6012-00 X axis preset value and 6013-00 X axis offset value" section on page 56.
Default = 0

## Setting the X-axis objects: 6010-00 Measured X axis value, 6011-00 Inversion of X axis range, 6012-00 X axis preset value and 6013-00 X axis offset value

**Dual-axis mode**

The measured value for X-axis angle is available at **6010-00 Measured X axis value** index. Value format is a two's complement 16-bit number and depends on the stored resolution. Value sign can be inverted by setting "01h" next to the **6011-00 Inversion of X axis range** object.
**6013-00 X axis offset value** object represents the **offset** value for X axis: it is a two's complement 16-bit number that is added to the absolute measured angle to get the relative angle stored as *Measured X value*:

Measured X angle $_{6010h}$ = $X_{ACQUIRED\ ABSOLUTE\ ANGLE}$ + X Offset $_{6013h}$

The offset can be set directly through an SDO command. It is also automatically modified when the preset value is set. The **preset** value is meant as the measured angle to be obtained at a specific inclination. Value format is a two's complement 16-bit number stored on the **6012-00 X axis preset value** object (for X axis). The preset value is subtracted from the absolute measured angle and the result is stored as offset value on the **6013-00 X axis offset value** index.

**EXAMPLE**
Let us consider the inclination of +13 deg on X-axis. In order to set the zero at +7 deg, it is necessary to set the new inclination for the sensor to read. So a preset value of +6 deg must be set.

The preset value can be used even as **auto-zero command (zero setting)**: if the value is set to 00h, the measured X-axis value becomes 00h and the offset is updated with the difference between the preset value (00h) and the current inclination.
It is strongly recommended to clear the offset value stored on **6013-00 X axis offset value** before setting a preset value, otherwise problems using the offset may occur.

**Single-axis mode**

In single-axis mode the inclinometer stores the data on the registers relative to the X-axis at the following indexes: **6010-00 Measured X axis value**, **6012-00 X axis preset value** and **6013-00 X axis offset value**, as explained in the section above.

The inversion object at index **6011-00 Inversion of X axis range** behaves depending on the data format set at index **3002-00 Single axis data format**. If the measure range is [-180;180] deg, the inversion mode behaves as explained in the previous section. Otherwise, if the range is [0;360] deg, the sign inversion reverses the direction of rotation with respect to the X axis, as shown in the following table.

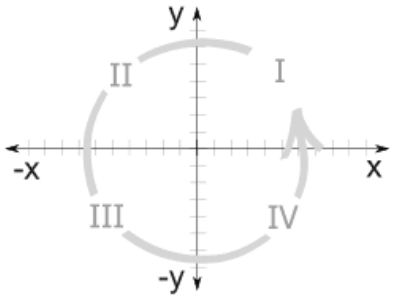
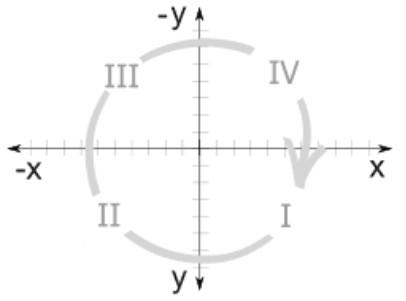
|  |  |
|---|---|
| **Angle inversion disabled** | **Angle inversion enabled** |

**Table 10 - Single axis mode inversion**

**EXAMPLE**

Let us suppose that we read an inclination of 35 deg. The value at index **3002-00 Single axis data format** is 00h (i.e. the range is [0;360] deg). Enable the inversion of X axis by writing 01h next to the **6011-00 Inversion of X axis range** object. Now the inclination is read as 325 deg.

### 6020-00 Measured Y axis value

[Integer16, ro]

For complete information please refer to the "Setting the Y-axis objects: 6020-00 Measured Y axis value, 6021-00 Inversion of Y axis range, 6022-00 Y axis preset value and 6023-00 Y axis offset value" section on page 58.

### 6021-00 Inversion of Y axis range

[Unsigned8, rw]
For complete information please refer to the "Setting the Y-axis objects: 6020-00 Measured Y axis value, 6021-00 Inversion of Y axis range, 6022-00 Y axis preset value and 6023-00 Y axis offset value" section on page 58.
Default = 0 (min. 0, max. 1)

### 6022-00 Y axis preset value

[Integer16, rw]
For complete information please refer to the "Setting the Y-axis objects: 6020-00 Measured Y axis value, 6021-00 Inversion of Y axis range, 6022-00 Y axis preset value and 6023-00 Y axis offset value" section on page 58.
Default = 0

### 6023-00 Y axis offset value

[Integer16, rw]
For complete information please refer to the "Setting the Y-axis objects: 6020-00 Measured Y axis value, 6021-00 Inversion of Y axis range, 6022-00 Y axis preset value and 6023-00 Y axis offset value" section on page 58.
Default = 0

## Setting the Y-axis objects: 6020-00 Measured Y axis value, 6021-00 Inversion of Y axis range, 6022-00 Y axis preset value and 6023-00 Y axis offset value

The measured value for Y-axis angle is available at **6020-00 Measured Y axis value** index. Value format is a two's complement 16-bit number and depends on the stored resolution. Value sign can be inverted by setting "01h" next to the **6021-00 Inversion of Y axis range** object.
**6023-00 Y axis offset value** object represents the **offset** value for Y axis: it is a two's complement 16-bit number that is added to the absolute measured angle to get the relative angle stored as *Measured Y value*:

$$\text{Measured Y angle}_{6020h} = Y_{\text{ACQUIRED ABSOLUTE ANGLE}} + \text{Y Offset}_{6023h}$$

The offset can be set directly through an SDO command. It is also automatically modified when the preset value is set. The **preset** value is meant as the measured angle to be obtained at a specific inclination. Value format is a two's complement 16-bit number stored on the **6022-00 Y axis preset value** object (for Y axis). The preset value is subtracted from the absolute measured angle and the result is stored as offset value on the **6023-00 Y axis offset value** index.

The preset value can be used even as **auto-zero command (zero setting)**: if the value is set to 00h, the measured Y-axis value becomes 00h and the offset is updated with the difference between the preset value (00h) and the current inclination.
It is strongly recommended to clear the offset value stored on **6023-00 Y axis offset value** object before setting a preset value, otherwise problems using the offset may occur.

**NOTE**
Please note that the above-mentioned registers are disabled for the single axis mode of measurement.

### 5.7 SDO commands

SDO commands (Service Data Objects) allow the user to read or modify the registers in the "Object dictionary". COB IDs used for SDO commands are stored at **1200 SDO1 Server parameter** index of the "Object dictionary" (see on page 39). IXB/IXC firmware version 2.x.y supports the segmented data transfer, in compliance with CiA DS 301 specifications.
Due to their asynchronous operation, SDO commands that are too frequently used could affect the inclination measure.

#### 5.7.1 How to read a register

SDO request frames allow the user to read data from the "Object dictionary". The frame structure is shown in the following table.

| COB ID | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
|---|---|---|---|---|---|---|---|---|
| 600h + NODE ID | Data length | Index LOW | Index HIGH | Sub-Index | Not used | | | |

**Table 11 – SDO request data frame**

Byte 0 contains the indication of data length as shown in the table below: if needed "*any size length*" = 40h must be used to avoid request errors.

| Data length | Value |
|---|---|
| Any size length | 40h |
| 1 | 4Fh |
| 2 | 4Bh |
| 4 | 43h |
| STRING | 41h |

**Table 12 – Values for Byte 0 (SDO request frame)**

IXB/IXC sends back a *response SDO frame*, whose structure -see below- is the same as the request frame. Byte 0 shows the current length of data allocated from Byte 4 to Byte 7.

| COB ID | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
|---|---|---|---|---|---|---|---|---|
| 580h + NODE ID | Data length | Index LOW | Index HIGH | Sub-Index | Read DATA (LSB to MSB) | | | |

**Table 13 – SDO response data frame**

**EXAMPLE**
IXB inclinometer can be set for either single-axis or double-axis measure. Let us check the set operational mode. The information to be read is available at **4001-00 Sensor operational mode 2 axes / 1 axis** index. Please set the data length of the request frame to the generic value (40h), COB ID is 600h+Node ID (see Table 2). Node ID is set to the default value 0Ah.
The resulting frame will be as follows:

| COB ID | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 ... Byte 7 |
|---|---|---|---|---|---|
| 60Ah | 40h | 01h | 40h | 00h | - |

**Table 14 – Example of SDO request frame**

The correct SDO response data frame will be as follows:

| COB ID | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 ... Byte 7 |
|---|---|---|---|---|---|
| 58Ah | 4Fh | 01h | 40h | 00h | 00h |

**Table 15 - Example of SDO response frame**

COB ID is 580h+Node ID. Byte 0 = "4Fh" indicates that the frame contains 1 byte of data. Bytes 1, 2 and 3 contain index and sub-index. Data is equal to "00h", this means that the sensor is set to 2-axis operational mode.

### 5.7.2 How to set a register

*SDO download request frames* are used to write a parameter into the "Object dictionary". Object index and sub-index must be specified at Bytes 1, 2 and 3. Data to write can be 1 byte, 2 byte, 3 byte or 4 byte long. The first byte (Byte 4 in Table 16) is the least significant byte. Firmware version 2.0.0 supports the segmented data transfer in compliance with CiA DS 301 specifications.
The frame structure is shown in the following table.

| COB ID | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
|---|---|---|---|---|---|---|---|---|
| 600h + NODE ID | Data length | Index LOW | Index HIGH | Sub-Index | DATA TO WRITE (LSB to MSB) | | | |

**Table 16 – SDO download request data frame**

Byte 0 contains the data length indication, so the user must choose one of the values listed in Table 17:

| Data length | Value |
|---|---|
| Any length | 22h |
| 1 byte | 2Fh |
| 2 bytes | 2Bh |
| 4 bytes | 23h |

**Table 17 - Values for Byte 0 (SDO download request frame)**

An *SDO download response frame* is sent back by IXB/IXC inclinometer to confirm that the request has succeeded. The structure is shown in the following table.

| COB ID | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
|---|---|---|---|---|---|---|---|---|
| 580h + NODE ID | 60h | Index LOW | Index HIGH | Sub-Index | 00h | 00h | 00h | 00h |

**Table 18 - SDO download response frame**

**EXAMPLE**
Let us set the IXB model (Node ID = 0Ah) to 2-axis mode and limit the X range to 10 degrees.

- Send an SDO command to program the **4001-00 Sensor operational mode 2 axes / 1 axis** index, and set the value 00h (= 2-axis mode).

| COB ID | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
|---|---|---|---|---|---|---|---|---|
| 60A | 22 | 01 | 40 | 00 | 00 | - | - | - |

If the request is correct, the following answer is sent back:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 58A | 60 | 01 | 40 | 00 | 00 | 00 | 00 | 00 |

- Now set the X-measuring range at **4000 Pitch and Roll value range** index. Enable the user range first (sub-index 03h):

| COB ID | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
|---|---|---|---|---|---|---|---|---|
| 60A | 22 | 00 | 40 | 03 | 01 | - | - | - |

If the request is correct, the following answer is received:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 58A | 60 | 00 | 40 | 03 | 00 | 00 | 00 | 00 |

- Set the X-range to 10 degrees. Pay attention: the value depends on the resolution currently set. For a resolution of 0.01 degrees (**6000-00 Resolution** index, value = 10 = Ah), the value to write is 1000 = 3E8h.

| COB ID | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
|---|---|---|---|---|---|---|---|---|
| 60A | 22 | 00 | 40 | 01 | E8 | 3 | - | - |

If the request is correct, the following answer is sent back:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 58A | 60 | 00 | 40 | 01 | 00 | 00 | 00 | 00 |

- To confirm the modifications, save the set parameters on the EEPROM as explained in the "5.7.4 SAVE ALL procedure and reset commands" section on page 64 and reset the device.

| COB ID | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
|---|---|---|---|---|---|---|---|---|
| 60A | 22 | 10 | 10 | 01 | 73 | 61 | 76 | 65 |

### 5.7.3 SDO abort codes

If an SDO request or an *SDO download request frame* fails, IXB/IXC inclinometer sends back an *SDO abort message* reporting the error sources. The SDO abort frame structure is reported below.

| COB ID | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
|---|---|---|---|---|---|---|---|---|
| 600h + NODE ID | 80h | Index LOW | Index HIGH | Sub-Index | SDO abort code (LSB to MSB) | | | |

**Table 19 - SDO abort frame**

For complete information please refer to the "SDO abort transfer protocol" section in the "CiA Draft Standard 301" document available at the address www.can-cia.org.

The transmitted data can be one of the error codes listed in the following table.

| SDO abort code | Bytes 4 ... 7 | | | | Type of failure |
|---|---|---|---|---|---|
| 0504 0001h | 01h | 00h | 04h | 05h | SDO command not valid or unknown |
| 0601 0002h | 02h | 00h | 01h | 06h | Attempt to write a read-only object |
| 0602 0000h | 00h | 00h | 02h | 06h | Object does not exist in the "Object dictionary" |
| 0604 0043h | 43h | 00h | 04h | 06h | General incompatibility reason |
| 0607 0010h | 10h | 00h | 07h | 06h | Data type does not match, length of service does not match |
| 0609 0011h | 11h | 00h | 09h | 06h | Sub-index does not exist |
| 0609 0030h | 30h | 00h | 09h | 06h | Parameter invalid value (download only) |
| 0609 0031h | 31h | 00h | 09h | 06h | Value of written parameter too high |
| 0609 0032h | 32h | 00h | 09h | 06h | Value of written parameter too low |
| 0800 0000h | 00h | 00h | 00h | 08h | General Error |
| 0800 0020h | 20h | 00h | 00h | 08h | Data cannot be transferred or stored to/in the application |
| 0800 0024h | 24h | 00h | 00h | 08h | No data available |

**Table 20 – SDO error codes**

### 5.7.4 SAVE ALL procedure and reset commands

Refer to the **1010-01 Store parameters** object on page 37.
When a register of the "Object dictionary" is changed, a *SAVE ALL* procedure is needed to store data on non-volatile memory (EEPROM). The word "save" must be written in the **1010-01 Store parameters** index from the least to the most significant byte (the resulting word is "73617665h"). Modifications are thus valid only after a hardware or software reset of the device.

The software reset commands listed in the "5.4 Network management objects (NMT)" section on page 31 are:

- Node reset (NMT command code 81h): this command resets the whole node and it is recommended for every saving procedure.
- Communication reset (NMT command code 82h): only the communication parameters are reset, so objects related to transmission and measure settings are not reset.

### 5.7.5 Restoring all parameters

Refer to the **1011-01 Restore default parameters** object on page 37.
The default configuration of the "Object dictionary" can be restored by writing the word "load" in ASCII format (= "6C6F6164h") in the **1011-01 Restore default parameters** object. **2000-00 Node-ID**, **2001-00 Baud rate** and **4001-00 Sensor operational mode 2 axes / 1 axis** objects will not be affected by this command, as they must be set manually.

### 5.8 Node ID and supported Baud Rates

For complete information please refer to the following objects:

- 2000-00 Node-ID on page 43;
- 2001-00 Baud rate on page 44.

### 5.9 Sensor operational mode and device type

For complete information please refer to the following objects:

- 4001-00 Sensor operational mode 2 axes / 1 axis on page 52;
- 1000-00 Device type on page 35.

### 5.10 TPDO1 transmission type

Refer to the 1800 Transmit PDO1 communication parameters object on page 39.
IXB/IXC inclinometer implements different types of transmission, which can be set at index 1800 Transmit PDO1 communication parameters sub-index 02h. Values from 0 dec to 240 dec (0h to F0h) are used for synchronous transmissions, 253 dec for the RTR transmission, and 254 dec for asynchronous transmissions.

#### 5.10.1 Synchronous transmissions and SYNC frames

The synchronous transmission occurs through SYNC frames, they are sent by the Master using the frame structure shown in the following table.

| COB ID |
|---|
| Sync COB ID object |

**Table 21 - SYNC frame structure**

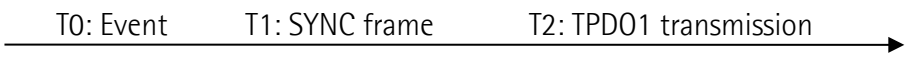
The default value of the Sync COB ID object is "80h"; it is stored at index 1005-00 COB_ID SYNC message of the "Object dictionary". When more than one sensor is connected to the CAN bus, each sensor must have a different Sync COB ID to correctly synchronize the communication. Available values for Sync COB ID objects are in the range [1;2047] dec: low values are suggested in order to assign a higher priority to the message on the bus. If different devices have the same Sync COB ID object, the SDO abort code 0609 0030h will be received. After any modification, the SAVE ALL procedure is required for validation (see on page 64).

Besides the COB ID definition, the transmission must be programmed at index **1800 Transmit PDO1 communication parameters** sub-index 02h. The stored value (from 1 to 240 dec) is the number of SYNC frames that are received by the inclinometer before sending back the TPDO1 frame.

**EXAMPLE**
Let us suppose that the inclinometer is connected to the bus and its Sync COB ID object is "2Bh"; the transmission type is set to "5h". This means that the Master must send the frame "2Bh" 5 times, in order to get the TPDO1 frame back from the specific inclinometer.

The transmission type can be set to 0 dec for an *acyclic synchronous transmission*. The IXB/IXC is configured to send the TPDO1 frame after an event occurrence only (e.g. event timer, transmission at inclination change, etc.) and the SYNC frame.

**EXAMPLE**
Consider the following time line:




T0, T1 and T2 are progressive moments. Only after an event occurrence and the receipt of a SYNC frame, the TPDO1 frame will be transmitted.

### 5.10.2 RTR: Remote Transmit Request

RTR is a request that the NMT Master performs directly to a specific CANbus node. Lika inclinometers implements RTRs for TPDO1 transmissions and Life guarding / Node guarding (refer to the "6.5.2 Node guarding and Life guarding" section on page 75).
If in operational mode and on receipt of an RTR, the IXB/IXC replies sending back the object requested by the RTR COB ID. In case of a TPDO1 request, the COB ID that is sent by the Master is as follows:

| COB ID |
|---|
| 180h + Node ID |

**Table 22 – RTR frame structure for a TPDO1 transmission**

An RTR can be sent at any time by the Master: IXB/IXC will answer even if other transmission types are set. If you want the IXB/IXC to answer to RTRs only, index **1800 Transmit PDO1 communication parameters** sub-index 2 must be set to 253 dec.

RTRs are the less recommended type of transmission in a CAN network.

### 5.10.3 Asynchronous transmissions

Asynchronous transmissions are triggered by an event occurrence, i.e. an inclination change or an event timer. This type of transmission is enabled by setting 254h (=FEh) in the index **1800 Transmit PDO1 communication parameters** sub-index 02h.

#### 5.10.3.1 Transmission at inclination change

For complete information on the transmission at inclination change please refer to the **3001 TPDO1 transmission at inclination change** object on page 48.

#### 5.10.3.2 Transmission on event timer

A TPDO1 can be temporally forced by an event timer and transmitted whether the measured inclination changes or not. The value stored at index **1800 Transmit PDO1 communication parameters** sub-index 05h represents the gap between two TPDO1 transmissions, values are in the range [1;65535] ms, with the fixed resolution of 1 ms. Default time is 0 ms, which means that the timer is deactivated and the TPDO1 is transmitted only at inclination changes. A minimum value of 50 ms is suggested in order to get correct data (please note that the inclinometer has an internal hardware low-pass filter with cut-frequency = 20 Hz).

### 5.10.4 Inhibit timer

In order to prevent CAN bus flooding due to continuous and frequent transmissions, an inhibit timer can be set. The inhibit timer is configured at index **1800 Transmit PDO1 communication parameters** sub-index 03h of the "Object dictionary". Available values are multiple of 100 µs in the range [0; 65535]. The value indicates the period during which no TPDO1 will be transmitted. The default value is "0" (inhibit timer disabled).
The inhibit timer setting can be done only if TPDO1 transmission is deactivated, by writing "8000 0180h + Node ID" (CiA DS 301 specifies that Bit 31 of the COB ID must be set to "1") in the index **1800 Transmit PDO1 communication parameters** sub-index 01h. As the inhibit timer is set, the TPDO1 transmission can be re-activated.
If transmission at inclination change is used (see the "5.10.3.1 Transmission at inclination change" section on page 67), an inhibit timer greater than 0 ms is suggested in order to prevent CAN bus saturations.

### 5.11 Transmit PDO1 mapping parameter

For complete information please refer to the **1A00 Transmit PDO1 mapping parameter** object on page 41.

### 5.12 Programmable digital filter

For complete information please refer to the **3000-00 Filtered samples** object on page 45.

### 5.13 Single axis data format

For complete information please refer to the **3002-00 Single axis data format** object on page 50.

### 5.14 Pitch and Roll Value Range in dual axis mode

For complete information please refer to the **4000 Pitch and Roll value range** object on page 50.

### 5.15 Internal temperature and surveillance

For complete information on the internal temperature please refer to the **5000-00 Device internal temperature [°C]** object on page 53.
For complete information on the temperature surveillance please refer to the **5001 Surveillance of the device's internal temperature** object on page 53.

### 5.16 Resolution

For complete information please refer to the **6000-00 Resolution** object on page 55.

### 5.17 Setting the X axis

For complete information on configuring the X axis please refer to the following objects:

- **6010-00 Measured X axis value** on page 55;
- **6011-00 Inversion of X axis range** on page 55;
- **6012-00 X axis preset value** on page 55;
- **6013-00 X axis offset value** on page 56.

### 5.18 Setting the Y axis

For complete information on configuring the Y axis please refer to the following objects:

- 6020-00 Measured Y axis value on page 57;
- 6021-00 Inversion of Y axis range on page 58;
- 6022-00 Y axis preset value on page 58;
- 6023-00 Y axis offset value on page 58.

# 6 Errors

CANopen manages error conditions through the transmission of Emergency frames and the update of specific registers in the "Object dictionary".

## 6.1 Error register

As soon as an error occurs, the **1001-00 Error register** is updated. The register is 1-byte long and each bit is associated to a specific error, concerning the work conditions, the communication or the internal status. The register can be accessed only for reading: depending on the error source, one of the bits in the error register is set to "1".

The bit scheme below shows the error register format:

| ERROR REGISTER FIELD STRUCTURE | | | | | | | |
|---|---|---|---|---|---|---|---|
| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| Hardware Error | Not Used | Profile Specific Error | Communication Error | Temperature Error | Not Used | | At least one active error |

**Table 23 - Error register format**

The 8 bits are activated when the following conditions occur:

1. Bit 0: every time an error condition occurs.
2. Bit 3: when the temperature exceeds the thresholds stored on the **5001 Surveillance of the device's internal temperature** register.
3. Bit 4: when a communication error occurs and the Communication Error Field changes (see the "6.2 Manufacturer error register" section below).
4. Bit 5: when either the user-defined range stored on the **4000 Pitch and Roll value range** register or the absolute sensor range are exceeded.
5. Bit 7: when a hardware error occurs either in the EEPROM CRC32 control or in the sensor self-test procedures performed at any device reset.

## 6.2 Manufacturer error register

The **1002-00 Manufacturer error register** shows the recent state of all detectable errors. It reports information about the communication and the device operation. Each bit refers to a specific error that is active if it is set to "1". The last 16 bits are also sent to the manufacturer specific part of the EMCY object shown in the "6.4 Emergency frames (EMCY)" section on page 72.

The manufacturer error register structure is shown in the table below:

| MANUFACTURER ERROR REGISTER FIELD STRUCTURE | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit31...Bit16 | Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| **Not Used** | **Communication Error Field** | | | | | | | | **Device Error Field** | | | | | | | |

| Communication Error Field | | | | | | | |
|---|---|---|---|---|---|---|---|
| Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| Guarding Error | Not Used | | | | Receive queue overrun | CAN BUS-OFF state reached | CAN WARNING limit reached |

| Device Error Field | | | | | | | |
|---|---|---|---|---|---|---|---|
| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| HW error (sensor self test) | Not used | | CRC32 memory error | Temperature error | Sensor Error Y - Axis | Sensor Error X - Axis | Not used |

**Table 24 – Structure of the manufacturer error register**

**EXAMPLE**
If the sensor position exceeds the longitudinal or lateral inclination thresholds stored on the **4000 Pitch and Roll value range** index of the "Object dictionary" (see on page 50), either Bit 1 or Bit 2 are set to "1" depending on the affected axis.

### 6.3 Pre-Defined error field

The **1003 Predefined error field** object stores the last 5 occurred error conditions. Sub-index 00h stores the number of occurred errors, which are chronologically saved on the following objects: the latest error is stored on sub-index 01h, the oldest one on sub-index 05h. As a new error condition occurs, if the register is full, the oldest error at sub-index 05h is deleted and the new error is added in sub-index 01h. The **1003 Predefined error field** object has the structure reported in the following table.

| PRE-DEFINED ERROR FIELD STRUCTURE | | |
|---|---|---|
| Bit 31 ... Bit 16 Additional Information Field | | Bit 15 ... Bit 0<br>Error Code (see Table 26) |
| Communication Error Field | Device Error Field | |

**Table 25 - Pre-defined error field structure**

The 16 most significant bits are the Communication and Device Error fields of the **1002-00 Manufacturer error register**. The 16 least significant bits represent an error code having the following description:

| Error Code | Error description |
|---|---|
| 0000h | Error reset or no more error present |
| 1030h | Generic error |
| 4200h | Device temperature error |
| 5000h | Self-test error or CRC memory error |
| 5010h | Sensor error on X axis |
| 5020h | Sensor error on Y axis |
| 8110h | Receive/Transmit buffer overflow |
| 8120h | CAN warning limit reached |
| 8130h | Node guard event occurred |
| 8140h | Recover from Bus-off |

**Table 26 - Pre-defined error codes**

### 6.4 Emergency frames (EMCY)

When an error occurs, an emergency message (*EMCY frame*) is triggered and sent with high priority to the bus. As soon as the emergency condition is recovered, an *Error reset* message is sent by the device.
As shown in the table below, EMCY frames are structured with the COB ID = 80h (specified at the **1014-00 COB-ID EMCY** index) + Node ID, followed by the Emergency object that contains complete information on the occurred error. The frame organization is shown in the table below.

| | Emergency Object | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| COB ID | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
| 80h + NODE ID | Emergency error code | | Error register | Manufacturer specific error field | | | | |
| | | | | Communication error | Device error | 00h | 00h | 00h |

**Table 27 - Emergency object frame structure**

**1015-00 Inhibit time EMCY** register allows the user to set an inhibit time between two consecutive EMCY frames. In the set period, the Client will not receive any EMCY frame. Bus overflow is thus limited when the device works close to error conditions. The 16-bit value is expressed in multiples of 100 µs. For further information please refer to page 38.

**EXAMPLE**

Let us set the device (Node ID = 0Ah) in 2-axis mode and limit the X range to 10 degrees. Then let us check the EMCY frame transmitted when the inclination of 10 degrees is exceeded, and the subsequent *Error reset* message transmitted when the inclination is below 10 degrees again.

- Send an SDO command to change the value of the **4001-00 Sensor operational mode 2 axes / 1 axis** object and set the value 00h (2-axis mode).

| COB ID | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
|---|---|---|---|---|---|---|---|---|
| 60A | 22 | 01 | 40 | 00 | 00 | - | - | - |

If the request is correct, the following answer is sent back:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 58A | 60 | 01 | 40 | 00 | 00 | 00 | 00 | 00 |

Now set the X-measuring range next to the **4000 Pitch and Roll value range** index.

- Enable the user range (see sub-index 03h):

| COB ID | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
|---|---|---|---|---|---|---|---|---|
| 60A | 22 | 00 | 40 | 03 | 01 | - | - | - |

If the request is correct, the following answer is sent back:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 58A | 60 | 00 | 40 | 03 | 00 | 00 | 00 | 00 |

- Set the X range to 10 degrees. Pay attention: the value depends on the currently set resolution. For a resolution of 0.01 degrees (index **6000-00 Resolution**, value = 10 = Ah), the correct value is 1000 = 3E8h.

| COB ID | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
|---|---|---|---|---|---|---|---|---|
| 60A | 22 | 00 | 40 | 01 | E8 | 3 | - | - |

If the request is correct, the following answer is sent back:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 58A | 60 | 00 | 40 | 01 | 00 | 00 | 00 | 00 |

- Save the set parameters on the EEPROM as explained in the "5.7.4 SAVE ALL procedure and reset commands" section on page 64 and reset the device.

| COB ID | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
|---|---|---|---|---|---|---|---|---|
| 60A | 22 | 10 | 10 | 01 | 73 | 61 | 76 | 65 |

- Set the device to **Operational** state.

| COB ID | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
|---|---|---|---|---|---|---|---|---|
| 00 | 01 | 0A | - | - | - | - | - | - |

- Finally tilt the inclinometer over 10 degrees on the X-axis. The following Emergency frame will transmitted:

| 8A | 10 | 50 | 21 | 00 | 02 | 00 | 00 | 00 |
|---|---|---|---|---|---|---|---|---|

The meaning of the message is as follows:

- "8A" is the COB ID for the EMCY frame.
- "10" is the low part of the error code, while "50" is the high part. The resulting error code is "**5010h**" as shown in Table 26.
- "21" is the value of the **1001-00 Error register**, where Bit 5 and Bit 0 are active, meaning that an error occurred and it is a Profile specific error.
- "00" is the Communication error of the **1002-00 Manufacturer error register**. Its value means that no communication error occurred.
- "02" is the Device error of the **1002-00 Manufacturer error register**. Bit 1 is active, meaning that the error occurred on X-axis.
- Now let us recover the unit from the emergency condition and tilt back the inclinometer below 10 degrees. An *Error reset* message will be received: the frame is an EMCY with updated error register values.

### 6.5 Failure monitoring: Heartbeat and Node guarding / Life guarding

When the inclinometer is configured for asynchronous transmissions, e.g. for transmission at inclination change (**3001 TPDO1 transmission at inclination change**, see on page 48), the transmission is not recurring, so the node cannot be periodically controlled. The CANopen network uses two different protocols to monitor the node state: the *heartbeat* and the *node guard / life guard protocol*. One protocol rules out the other: if both are active, the *heartbeat* outweighs and the *node guarding* is deactivated.

### 6.5.1 Heartbeat

The *heartbeat* is a failure monitoring mechanism that is managed by the CAN Slave (inclinometer). If active, the node sends cyclically a heartbeat message, which contains information about the state of the inclination sensor. The transmission of the *heartbeat frame* can be enabled by setting a value greater than zero next to **1017-00 Producer heartbeat time** index. The value represents the interval between two heartbeat transmissions and is expressed in milliseconds. Values lower than 50 ms are automatically set to 50 ms. The default value is zero, i.e. no heartbeat transmission is set. The *heartbeat* message has the following frame structure:

| COB ID | Byte 0 | Corresponding state |
|---|---|---|
| 700h + NODE ID | 00h | Boot up |
| | 04h | Stop condition |
| | 05h | Run condition |
| | 7Fh | Pre-operational condition |

**Table 28 - Heartbeat frame structure**

### 6.5.2 Node guarding and Life guarding

The *Node guarding* is the monitoring of one or several nodes interfaced to the CANopen network through cyclic RTR frames (see the "5.10.2 RTR: Remote Transmit Request" section on page 66). As the CAN Master sends an *RTR message frame* to the node to be monitored, the requested node answers providing its state and a toggle bit. The toggle bit is toggled after every node guarding request. If the status/toggle bit does not match the expected status/toggle bit, or no response is provided to the Master, a Slave error is triggered.

This mechanism can be even used to detect Master failures. In this case two parameters are used: the **100C-00 Guard time** object and the **100D-00 Life time factor** object. The **100C-00 Guard time** parameter specifies the interval between two state requests from the Master. The **100D-00 Life time factor** defines the time multiplier after which the connection with the Master is assumed as interrupted. This time is defined as the *node lifetime*:

NODE LIFETIME = **100C-00 Guard time** * **100D-00 Life time factor**

If the node does not receive any guarding request from the Master within the lifetime, a Master failure is assumed and the device sends an *EMCY frame*. The node enters the **Pre-Operational** state.

# 7 Default parameters list

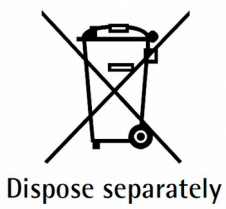
| Parameters list | Default value | | |
|---|---|---|---|
| 1000-00 Device type | 0001 019Ah = 1-axis inclinometer<br>0002 019Ah = 2-axis inclinometer | | |
| 1001-00 Error register | 00h | | |
| 1002-00 Manufacturer error register | 0000 0000h | | |
| 1005-00 COB_ID SYNC message | 0000 0080h | | |
| 100C-00 Guard time | 0000h | | |
| 100D-00 Life time factor | 00h | | |
| 1014-00 COB-ID EMCY | 0000 0080h+ NodeID | | |
| 1015-00 Inhibit time EMCY | 0000h | | |
| 1017-00 Producer heartbeat time | 0000h | | |
| 1018 Identity object<br>1018-01<br>1018-02<br>1018-03<br>1018-04 | <br>0000 012Eh<br>0041 4333h<br>0000 0041h<br>- | | |
| 1200 SDO1 Server parameter<br>1200-01<br><br>1200-02 | <br>0000 0600h+ NodeID<br>0000 0580h+ NodeID | | |
| 1800 Transmit PDO1 communication parameters<br>1800-01<br><br>1800-02<br>1800-03<br>1800-04<br>1800-05 | <br>0000 0180h+ NodeID<br>FEh<br>0000h<br>-<br>0000h | | |
| 1A00 Transmit PDO1 mapping parameter<br>1A00-01<br>1A00-02<br>1A00-03 | <br>6010 0010h<br>6020 0010h<br>5000 0008h | | |
| 2000-00 Node-ID | 0Ah | | |
| 2001-00 Baud rate | 01F4h | | |
| 3000-00 Filtered samples | 03E8h | | |
| 3001 TPDO1 transmission at inclination change<br>3001-01 | <br>00h | | |

| | | | |
|---|---|---|---|
| 3001-02<br>3001-03 | 0064h<br>0064h | | |
| 3002-00 Single axis data format | 00h | | |
| 4000 Pitch and Roll value range<br>4000-01<br>4000-02<br>4000-03 | <br>7530h<br>7530h<br>00h | | |
| 4001-00 Sensor operational mode 2 axes / 1 axis | 00h | | |
| 5001 Surveillance of the device's internal temperature<br>5001-01<br>5001-02<br>5001-03 | <br>00h<br>DDh<br>4Bh | | |
| 6000-00 Resolution | 0001h | | |
| 6011-00 Inversion of X axis range | 00h | | |
| 6012-00 X axis preset value | 0000h | | |
| 6013-00 X axis offset value | 0000h | | |
| 6021-00 Inversion of Y axis range | 00h | | |
| 6022-00 Y axis preset value | 0000h | | |
| 6023-00 Y axis offset value | 0000h | | |

This page intentionally left blank

This page intentionally left blank

| Document release | Release date | Description | HW | SW | Interface |
|---|---|---|---|---|---|
| 1.0 | 13.06.2016 | First issue | 1 | 1.0 | - |
| 1.1 | 12.10.2016 | Mounting instructions updated | 1 | 1.0 | - |


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

Tel. +39 0445 806600
Fax +39 0445 806699

info@lika.biz • www.lika.biz