Smart encoders & actuators

User's guide

IXM1
IXM2

Version RS-485

- 1-axis and 2-axis inclinometers
- MODBUS RTU interface (RS-485)
- Resolution down to 0.01°
- Robust die cast housing with IP67 protection rate
- Operating temperature -40°C +85°C

Suitable for the following models:

- IXM1-MB...
- IXM2-MB...

Table of Contents

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 **IXM series inclinometers with MODBUS RTU (RS-485) interface**.

Inclinometers are devices designed to measure the levelling and pitch and roll angles for motion control or safety purposes.
IXM series inclinometers are equipped with MODBUS RTU interface (while IXB and IXC series inclinometers are equipped with CANopen interface and IXA series inclinometers are equipped with analogue interface).
CANopen inclinometers are available with either 1-axis or 2-axis operation. IXM1 inclinometers are 1-axis sensors; IXM2 inclinometers are 2-axis sensors.
The measurement range is 0-360 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 **MODBUS Interface**, both general and specific information is given on the MODBUS RTU interface. In this section the interface features and the MODBUS registers implemented in the unit are fully described.

Glossary of MODBUS terms

MODBUS, 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 MODBUS interface. They are listed in alphabetical order.

Address field	It contains the Slave address.
Application Process	The Application Process is the task on the Application Layer.
Application protocol	MODBUS is an application protocol or messaging structure that defines rules for organizing and interpreting data independent of the data transmission medium.
ASCII transmission mode	When devices are setup to communicate on a MODBUS serial line using ASCII (American Standard Code for Information Interchange) mode, each 8-bit byte in a message is sent as two ASCII characters. This mode is used when the physical communication link or the capabilities of the device does not allow the conformance with RTU mode requirements regarding timers management.
Bus	A bus is a communication medium connecting several nodes. Data can be transferred via serial or parallel circuits, that is, via electrical conductors or fibre optic.
Client	A Client is any network device that sends data requests to servers. MODBUS follows the Client/Server model. MODBUS Masters are referred to as Clients, while MODBUS Slaves are Servers.
Cyclic Redundancy Check (CRC)	Error-checking technique in which the frame recipient calculates a remainder by dividing frame contents by a prime binary divisor and compares the calculated remainder to a value stored in the frame by the sending node.
Data encoding	MODBUS uses a 'big-Endian' representation for addresses and data items. This means that when a numerical quantity larger than a single byte is transmitted, the most significant byte is sent first.
Exception code	Code to be returned by Slaves in the event of problems. All exceptions are signalled by adding 0x80 to the function code of the request.
Exception response	MODBUS operates according to the common client/server (Master/Slave) model: the Client (Master) sends a request telegram (service request) to the Server (Slave), and the Server replies with a response telegram. If the Server cannot process a request, it will instead return a error function code (exception response) that is the original function code plus 80H (i.e. with its most significant bit set to 1).
Function code	MODBUS is a request/reply protocol and offers services

	specified by function codes. The function code is sent from a Client to the Server and indicates which kind of action the Server must perform. MODBUS function codes are elements of MODBUS request/reply PDUs. The function code field of a MODBUS data unit is coded in one byte. Valid codes are in the range of 1 ... 255 decimal (the range 128 – 255 is reserved and used for exception responses). Function code "0" is not valid. Lika devices only implement public function codes.
Holding register	In the MODBUS data model, a Holding register is the output data. A Holding register has a 16-bit quantity, is alterable by an application program, and allows either read-write or read-only access.
IEEE 1588	This standard defines a protocol enabling synchronisation of clocks in distributed networked devices (e.g. connected via Ethernet).
Input register	In the MODBUS data model, an Input register is the input data. An Input register has a 16-bit quantity, is provided by an I/O system, and allows read-only access.
LRC Checking	In ASCII mode, messages include an error–checking field that is based on a Longitudinal Redundancy Checking (LRC) calculation that is performed on the message contents, exclusive of the beginning 'colon' and terminating CRLF pair characters. It is applied regardless of any parity checking method used for the individual characters of the message.
Master	A Master is any network device that sends data requests to Slaves.
Message	The MODBUS messaging service provides a Client/Server communication between devices connected on the network. The Client / Server model is based on four types of messages: • MODBUS Request • MODBUS Confirmation • MODBUS Indication • MODBUS Response The MODBUS messaging services are used for information exchange.
MODBUS Confirmation	A MODBUS Confirmation is the Response Message received on the Client side.
MODBUS Indication	A MODBUS Indication is the Request message received on the Server side.
MODBUS Request	A MODBUS Request is the message sent on the network by the Client to initiate a transaction.
MODBUS Response	A MODBUS Response is the Response message sent by the Server.
Network	Network is a group of computers on a single physical network segment.

PDU	The Protocol Data Unit (PDU) is the MODBUS function code and data field. It is packed together with the Address Field and the CRC (or LRC) to form the Modbus Serial Line PDU. The MODBUS protocol defines three PDUs. They are: • MODBUS Request PDU, mb_req_pdu • MODBUS Response PDU, mb_rsp_pdu • MODBUS Exception Response PDU, mb_excep_rsp_pdu
Read Holding Registers (03, 0003hex)	This function code is used to READ the contents of a contiguous block of holding registers in a remote device; in other words, it allows to read the values set ina group of work parameters placed in order.
Read Input Register (04, 0004hex)	This function code is used to READ from 1 to 125 contiguous input registers in a remote device; in other words, it allows to read some result values and state / alarm messages in a remote device.
Register	MODBUS functions operate on memory registers to configure, monitor, and control device I/O.
RTU transmission mode	Remote Terminal Unit. When devices communicate on a MODBUS serial line using the RTU mode, each 8-bit byte in a message contains two 4-bit hexadecimal characters. The main advantage of this mode is that its greater character density allows better data throughput than ASCII mode for the same baud rate. Each message must be transmitted in a continuous stream of characters.
Server	A Server is any program that awaits data requests to be sent to it. Servers do no initiate contacts with Clients, but only respond to them. MODBUS follows the Client/Server model. MODBUS Masters are referred to as clients, while MODBUS Slaves are servers.
Service request	It is the MODBUS Request, i.e. the message sent on the network by the Client to initiate a transaction.
Slave	A Slave is any program that awaits data requests to be sent to it. Slaves do no initiate contacts with Masters, but only respond to them.
Transmission rate	Data transfer rate (in bps).
Write Multiple Registers (16, 0010hex)	This function code is used to WRITE a block of contiguous registers (1 to 123 registers) in a remote device.
Write Single Register (06, 0006hex)	This function code is used to WRITE a single holding register in a remote device.

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 the power supply before connecting the device;
- connect according to the explanation in the "Electrical connections" section on page 18;
- 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 on page 15;
- mechanical installation has to be carried out with stationary mechanical parts;
- do not disassemble the unit;
- delicate electronic equipment: handle with care;
- do not subject the device 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

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

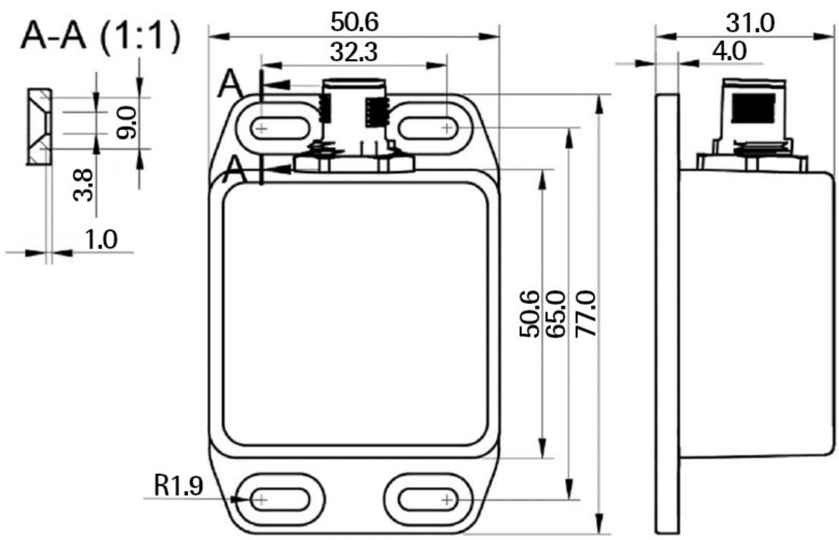
Warning: devices having order code ending with "/Sxxx" may have mechanical and electrical characteristics different from standard and be supplied with additional documentation for special connections (Technical Info).

3 Mechanical installation

WARNING
Installation and maintenance operations must be carried out by qualified personnel only, with power supply disconnected. Mechanical components must be in stop.

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

3.1 Overall dimensions (Figure 1)



(values are expressed in mm)

Figure 1 - IXM1 and IXM2 inclinometers – Overall dimensions

3.2 Installation (Figure 2 and Figure 3)

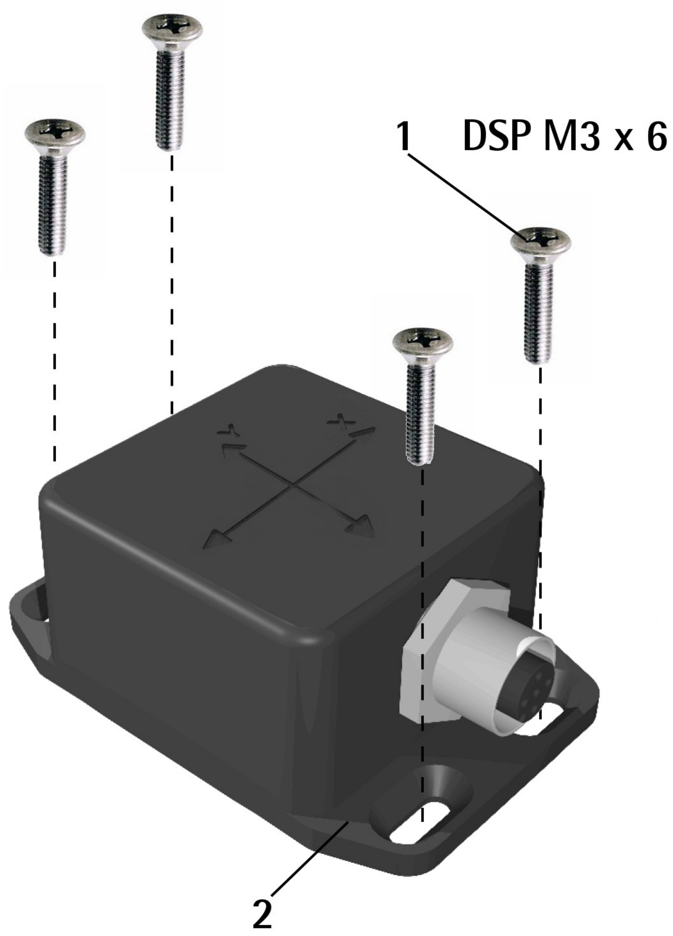


Figure 2 - Installation

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 a horizontal plane (see Figure 3);
- 2-axis inclinometers can be mounted also upside down.

3.3 Orientation of the axis

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).

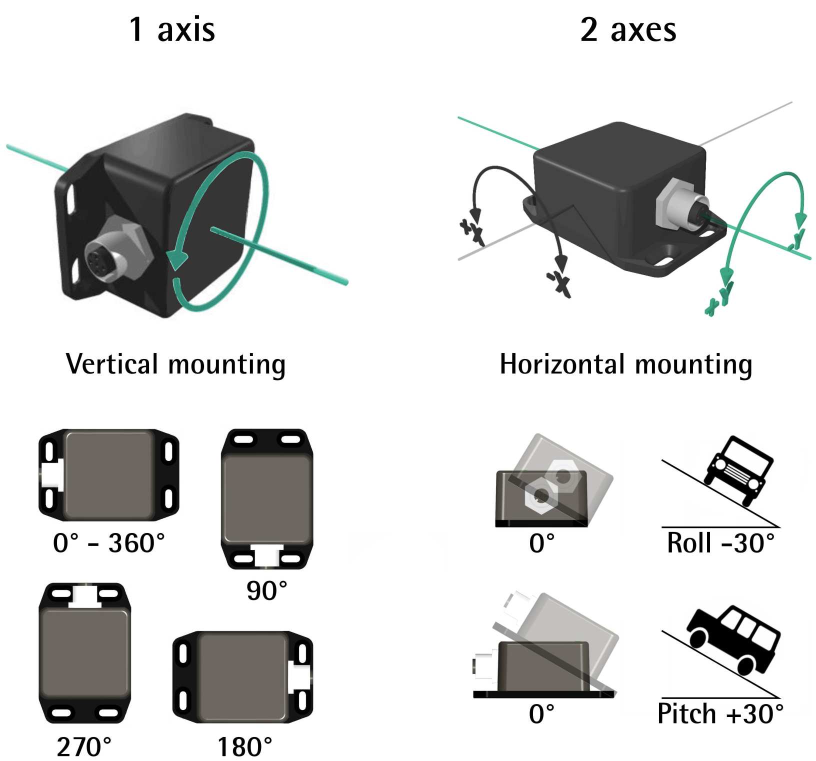
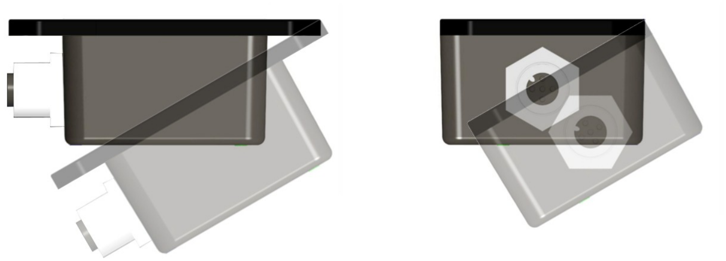


Figure 3 - Mounting positions

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. Shaft and mechanical components must be in stop.

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

4.1 Cable output (IXM...L...)

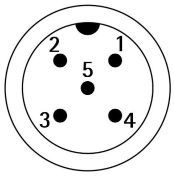
Wiring colours	Description
White	+10Vdc +30Vdc power supply voltage
Green [1]	0Vdc power supply voltage
Yellow	Modbus A (RS-485+) (D1) Transceiver terminal 1, V1 Voltage (V1 > V0 for binary 1 [OFF] state)
Brown	Modbus B (RS-485-) (D0) Transceiver terminal 0, V0 Voltage (V0 > V1 for binary 0 [ON] state)
Shield	Cable shield (connect to ground)

1 0Vdc of the RS-485 serial connection too.

IXM inclinometer cable has a typical outer diameter of 4.5 mm and it is composed of four 0.14 mm^2 conductors (size 26 AWG) with a braided screen of tin copper wire. The cable weight is approx. 40 g/m.

4.2 M12 5-pin connector (IXM...M...)

M12 5-pin
male connector
A coding
(frontal view)

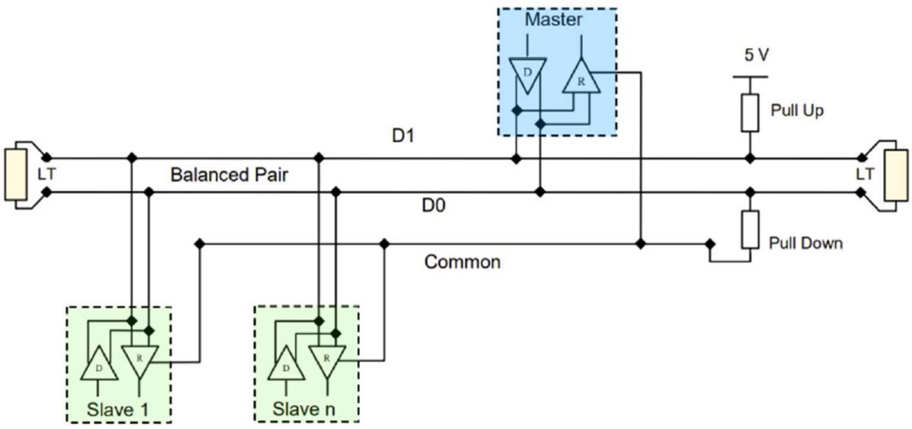


Pin	Description
1	Not used
2	+10Vdc +30Vdc power supply voltage
3 [1]	0Vdc power supply voltage
4	Modbus A (RS-485+) (D1) Transceiver terminal 1, V1 Voltage (V1 > V0 for binary 1 [OFF] state)
5	Modbus B (RS-485-) (D0) Transceiver terminal 0, V0 Voltage (V0 > V1 for binary 0 [ON] state)

1 0Vdc of the RS-485 serial connection too.

4.3 Wiring scheme

General wiring scheme with Line Termination resistance (LT) on both the ends of RS-485 BUS plus line polarization.



The "Common" circuit (Signal and optional Power Supply Common) must be connected directly to the protective ground.
This connection has to be done preferably in a single point of the bus. Generally, choose a point which is close to the Master device.
To minimize reflections from the RS-485 BUS cable ends, place an LT (Line Termination) close to each one of the two ends. The line must be terminated at both ends, since the propagation is bi-directional. Never use more than two LT or place any LT on a derivation cable.
Each LT must be connected between the two conductors of the balanced line: D0 and D1. LT may be a 150 Ω 0.5W resistor or a capacitor of 1nF 10V (min) in series with a 120 Ω (0.25 W) resistor. This second option is a better choice when the BUS lines are polarized. For more details on polarization line, please refers to "MODBUS over serial line specification and implementation guide V1.02".

4.4. Supply voltage and current consumption

Admitted supply voltage is in the range of +10Vdc +30Vdc. Do not supply the device in AC mode.
The current consumption is typically 6.50 mA with power supply voltage ranging from +10Vdc to +30Vdc (measured angle reading cycle of the inclinometer is 250 ms).

4.5. Supported standards and protocols

The IXM Modbus inclinometer supports the following standards and protocols:

- MODBUS over Serial Line Specification and Implementation Guide V1.02
- MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b3

Additional device-specific registers are available allowing specific features and functionality for IXM inclinometers.

4.6 Operating modes

Modbus IXM inclinometers requires polling method to read data from devices. The inclinometers are factory programmed to operate as single-axis or dual-axis tilt sensors. The end-user can modify several registers to adjust the digital filter, offset, preset, inversion, auto-zero, and many other parameters accessing the inclinometers Modbus registers via the Modbus interface. All the user-programmable parameters can be stored in the non volatile memory (NVM) of the inclinometer. After being stored on the NVM, register values are retained also when the power supply of the inclinometer is turned off. When the inclinometers are powered up the configuration parameters will be loaded from the NVM in order to initialize the device.

4.7 Line termination

For complete information refer to the "4.3 Wiring scheme" section on page 19.

4.8 Setting the baud rate

For complete information refer to the "5.2 Setting the baud rate" section on page 23.

4.9 Setting the parity

For complete information refer to the "5.3 Setting the parity" section on page 23.

4.10 Setting the node address

For complete information refer to the "5.4 Setting the node address" section on page 23.

5 Quick reference

5.1 Getting started

The following instructions allow the operator to set up the device for standard operation in a quick and safe mode.

- Mechanically install the device, see on page 15 ff;
- execute the electrical connections, see on page 18 ff;
- configure the Master device communication parameters in order to meet the IXM factory configuration:
  - Baud rate = 19,200 bits per second;
  - 8 data bits;
  - Even parity bit;
  - 1 stop bit;
  - no flow control;
- switch +10Vdc ÷ +30Vdc power supply on;
- if required, set the the baud rate according to the Modbus network settings; refer to the **Baud rate [000A hex]** register; see the "5.2 Setting the baud rate" section on page 23;
- if required, set the parity bit and the stop bit according to the Modbus network settings; refer to the **Parity [000B hex]** register and to the **Stop bits [000C hex]** register; see the "5.3 Setting the parity" section on page 23; when the parity is enabled to be Even or Odd, the number of stop bits is forced to 1 bit; when the parity check is disabled, the number of stop bits can be configured to 1 or 2 bits ("Modbus over serial line specification and implementation guide V1.02" protocol requires 2 stop bits when the parity check is disabled);
- if required, set the node address; the default value set by Lika Electronic at factory set-up is "100" (0x64 hex); refer to the **Node address [000D hex]** register; see the "5.4 Setting the node address" section on page 23; it has to be unique in the network; the use of two identical node addresses in the same Modbus network causes communication errors, a wrong behaviour of the nodes, and the impossibility of communication among the nodes;
- save the IXM configuration on the non volatile memory by means of the **Store all [0032 hex]** register;
- restart the IXM inclinometer by switching the power supply OFF and then ON again;
- poll the inclinometer and read the values by means of the "Read Holding Registers" command.

5.2 Setting the baud rate

The baud rate set at factory is "19,200 bits per second" (value 4). If you need to change it, set a value between "1" and "5" in the **Baud rate [000A hex]** Modbus register.

Value (hex)	Value (dec)	Baud rate
0001	1	2,400 bit/s
0002	2	4,800 bit/s
0003	3	9,600 bit/s
0004	**4**	**19,200 bit/s (default)**
0005	5	38,400 bit/s

5.3 Setting the parity

The factory parity setting is "Even" (value 2). If you neeed to change it, set a value between "1" and "3" in the **Parity [000B hex]** Modbus register.

Value (hex)	Value (dec)	Parity check type
0001	1	NONE
0002	**2**	**EVEN**
0003	3	ODD

5.4 Setting the node address

The factory node address is "100" (0x64 hex). If you need to change it, set a value between "1" and "247" in the **Node address [000D hex]** Modbus register. Then send the **Store all [0032 hex]** register command to save the parameters on the non volatile memory. Address "0" is reserved for broadcast messages and shall not be used by any mode in the Modbus network.

5.5 Store All

To store the configuration of the inclinometer in the NVM, write the ASCII value corresponding to "ST" (0x5354 hex, 21,332 dec) in the **Store all [0032 hex]** register. The new value is effective after the inclinometer restarts (set the **ResetDev [0034 hex]** register frame or switch the power supply OFF and then ON).

5.6 Measured values

Once the inclinometer is configured to operate in the Modbus network it is possible to read the measured angles by accessing the Modbus table of the device. Use the Read Holding Registers (CMD code 0x03 hex) request frames.

5.6.1 Single-axis inclinometers

The measured angle is available reading the registers **180 angle [0003 hex]** and **360 angle [0004 hex]** in the Modbus table. Both registers store the angle value but in different data format.

180 angle [0003 hex] register value is expressed in the ±180 degree format, the number is a 2'complement 16-bit signed integer fixed-point value with a resolution of 0.01 degrees.

360 angle [0004 hex] register value is expressed in the 0…360 degree format, the number is a 16-bit unsigned integer value with a resolution of 0.01 degrees.

EXAMPLE

We want to read the angle measured by the inclinometer having node address 100 and then convert the values.

The measured angle is -33.17 deg (±180 degree format) or 326.83 deg (0…360 degree format).

NADS 100; RD from RG.0003 to RG.0004	0x64 0x03 0x00 0x03 0x00 0x02 0x3D 0xFE
NADS 100 RESP	0x64 0x03 0x04 0xF3 0x0B 0x7F 0xAB 0xEC 0x3C
180 angle [0003 hex] value	0xF30B → 2'complement 16-bit integer → -3317 resolution is 0.01 deg → -3317 / 100 = -33.17 deg
360 angle [0004 hex] value	0x7FAB → unsigned 16-bit integer → 32683 resolution is 0.01 deg → 32683 / 100 = 326.83 deg

5.6.2 Dual-axis inclinometers

The measured angles are available by reading the **X angle [0001 hex]** register for X axis and the **Y angle [0002 hex]** register for Y axis.

Both angles are expressed as 2'complement 16-bit signed integer fixed-point value with a resolution of 0.01 degrees.

EXAMPLE

We want to read the angle measured by the inclinometer having node address 72 and then convert the values.

The measured angles are -32.52 deg for X axis and +10.69 deg for Y axis.

NADS 72; RD from RG.0001 to RG.0002	0x48 0x03 0x00 0x01 0x00 0x02 0x9B 0x92
NADS 72 RESP	0x48 0x03 0x04 0xF3 0x4C 0x04 0x2D 0xEC 0x3C
X angle [0001 hex] value	0xF34C → 2'complement 16-bit integer → -3252 resolution is 0.01 deg → -3252 / 100 = -32.52 deg
Y angle [0002 hex] value	value 0x042D → 2'complement 16-bit integer → +1069 resolution is 0.01 deg → +1069 / 100 = +10.69 deg

6 MODBUS® interface

Lika inclinometers are Slave devices and implement the MODBUS application protocol (level 7 of OSI model) and the "Modbus over Serial Line" protocol (levels 1 & 2 of OSI model).
For any further information or omitted specifications please refer to "Modbus Application Protocol Specification V1.1b3" and "Modbus over Serial Line. Specification and Implementation Guide V1.02" available at www.modbus.org.

6.1 MODBUS Master / Slaves protocol principle

The Modbus Serial Line protocol is a Master – Slaves protocol. One only Master (at the same time) is connected to the bus and one or several (247 maximum number) Slave nodes are also connected to the same serial bus. A Modbus communication is always initiated by the Master. The Slave nodes will never transmit data without receiving a request from the Master node. The Slave nodes will never communicate with each other. The Master node initiates only one Modbus transaction at a time.

The Master node issues a Modbus request to the Slave nodes in two modes:

- **UNICAST mode**: in that mode the Master addresses an individual Slave. After receiving and processing the request, the Slave returns a message (a "reply") to the Master. In that mode, a Modbus transaction consists of two messages: a request from the Master and a reply from the Slave. Each Slave must have a unique address (from 1 to 247) so that it can be addressed independently from other nodes. Lika devices only implement commands in "unicast" mode.
- **BROADCAST mode**: in that mode the Master can send a request to all Slaves at the same time. No response is returned to "broadcast" requests sent by the Master. The "broadcast" requests are necessarily writing commands. The address 0 is reserved to identify a "broadcast" exchange. Lika devices do not implement commands in "broadcast" mode.

NOTE
Lika devices do not implement commands in "broadcast" mode.

6.2 MODBUS frame description

The Modbus application protocol defines a simple Protocol Data Unit (PDU) independent of the underlying communication layers:

Function code	Data

MODBUS PDU

The mapping of Modbus protocol on a specific bus or network introduces some additional fields on the Protocol Data Unit. The client that initiates a Modbus transaction builds the Modbus PDU, and then adds fields in order to build the appropriate communication PDU.

MODBUS SERIAL LINE PDU

Address field	Function code	Data	CRC (or LRC)

MODBUS PDU

- **ADDRESS FIELD**: on Modbus Serial Line the address field only contains the Slave address. As previously stated (see the "5.4 Setting the node address" section on page 23), the valid Slave node addresses are in the range of 0 – 247 decimal. The individual Slave devices are assigned addresses in the range of 1 – 247. A Master addresses a Slave by placing the Slave address in the **ADDRESS FIELD** of the message. When the Slave returns its response, it places its own address in the response **ADDRESS FIELD** to let the Master know which Slave is responding.
- **FUNCTION CODE**: the function code indicates to the Server what kind of action to perform. The function code can be followed by a **DATA** field that contains request and response parameters. For any further information on the implemented function codes refer to the "6.5 Function codes" section on page 30.
- **DATA**: the **DATA** field of messages contains the bytes for additional information and transmission specifications that the server uses to take the action defined by the **FUNCTION CODE**. This can include items such as discrete and register addresses, the quantity of items to be handled, and the count of actual data bytes in the field. The structure of the **DATA** field depends on each **FUNCTION CODE** (refer to the "6.5 Function codes" section on page 30).
- **CRC (Cyclical Redundancy Checking)**: error checking field is the result of a "Redundancy Checking" calculation that is performed on the message contents. This is intended to check whether transmission has been performed properly. The CRC field is two bytes, containing 16-bit

binary value. The CRC value is calculated by the transmitting device, which appends the CRC to the message. The device that receives recalculates a CRC during receipt of the message and compares the calculated value to the actual value it received in the CRC field. If the two values are not equal, an error results.

The Modbus protocol defines three PDUs. They are:

- **Modbus Request PDU**;
- **Modbus Response PDU**;
- **Modbus Exception Response PDU**.

The **Modbus Request PDU** is defined as {function_code, request_data}, where:
function_code = Modbus function code [1 byte];
request_data = this field is function code dependent and usually contains information such as variable references, variable counts, data offsets, sub-function, etc. [n bytes].

The **Modbus Response PDU** is defined as {function_code, response_data}, where:
function_code = Modbus function code [1 byte];
response_data = this field is function code dependent and usually contains information such as variable references, variable counts, data offsets, sub-function, etc. [n bytes].

The **Modbus Exception Response PDU** is defined as {exception-function_code, exception_code}, where:
exception-function_code = Modbus function code + 0080 hex [1 byte];
exception_code = Modbus Exception code, refer to the table "Modbus Exception Codes" in the section 7 of the document "Modbus Application Protocol Specification V1.1b".

6.3 Transmission modes

Two different serial transmission modes are defined in the Modbus serial protocol: the **RTU** (**Remote Terminal Unit**) **mode** and the **ASCII mode**. The transmission mode defines the bit contents of message fields transmitted serially on the line. It determines how information is packed into the message fields and decoded. The transmission mode and the serial port parameters must be the same for all devices on a Modbus Serial Line. All devices must implement the RTU mode, while the ASCII mode is an option. Lika devices only implement RTU transmission mode, as described in the following section.

6.3.1 RTU transmission mode

When devices communicate on a Modbus serial line using the RTU (Remote Terminal Unit) mode, each 8-bit byte in a message contains two 4-bit hexadecimal characters. Each message must be transmitted in a continuous stream of characters. Synchronization between the messages exchanged by the transmitting device and the receiving device is achieved by placing an interval of at least 3.5 character times between successive messages, this is called "silent interval". In this way a Modbus message is placed by the transmitting device into a frame that has a known beginning and ending point. This allows devices that receive a new frame to begin at the start of the message and to know when the message is completed. So when the receiving device does not receive a message for an interval of 4 character times, it considers the previous message as completed and the next byte will be the first of a new message, i.e. an address.
When baud rate = 9,600 bit/s the "silent interval" is 4 ms.
When baud rate = 19,200 bit/s the "silent interval" is 2 ms.

The format (11 bits) for each byte in RTU mode is as follows:

Coding system: 8-bit binary
Bits per Byte: 1 start bit;
8 data bits, least significant bit (lsb) sent first;
1 bit for parity completion (= Even);
1 stop bit.

Modbus protocol uses a "big-Endian" representation for addresses and data items. This means that when a numerical quantity greater than a single byte is transmitted, the most significant byte (MSB) is sent first.
Each character or byte is sent in this order (left to right):

lsb (Least Significant Bit) … msb (Most Significant Bit)

Start	1	2	3	4	5	6	7	8	Parity*	Stop

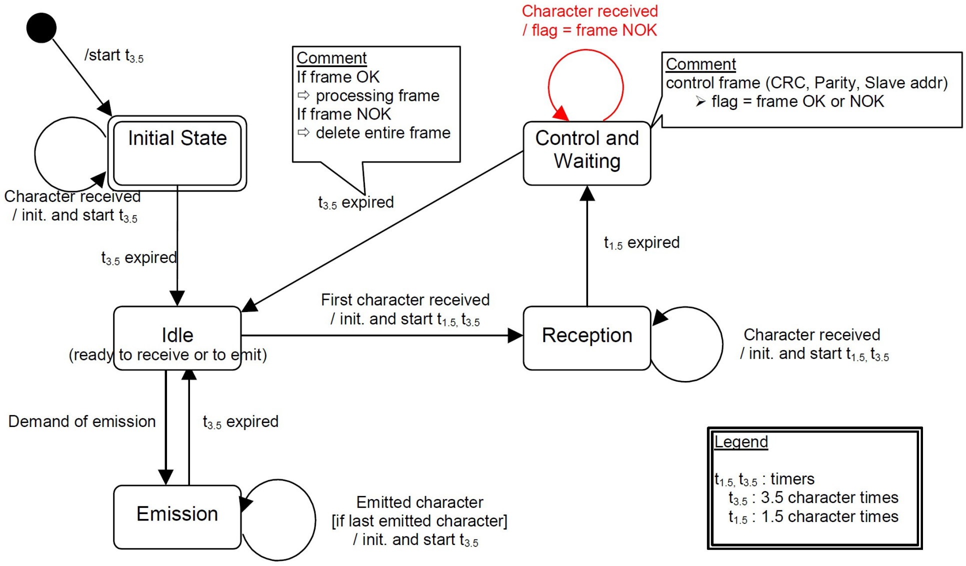
* When "No parity" is activated, the parity bit is replaced by a stop bit.

The default parity mode must be even parity.

The maximum size of the Modbus RTU frame is 256 bytes, its structure is as follows:

Slave Address	Function code	Data	CRC	
1 byte	1 byte	0 up to 252 byte(s)	2 bytes	
			CRC Low	CRC Hi

The following drawing provides a description of the RTU transmission mode state diagram. Both "Master" and "Slave" points of view are expressed in the same drawing.



- Transition from **Initial State** to **Idle** state needs an interval of at least 3,5 character times (time-out expiration = $t_{3.5}$).
- **Idle** state is the normal state when neither emission nor reception is active. In RTU mode, the communication link is declared in **Idle** state when there is no transmission activity after a time interval equal to at least 3.5 characters ($t_{3,5}$).
- A request can only be sent in **Idle** state. After sending a request, the Master leaves the **Idle** state and cannot send a second request at the same time.
- When the link is in **Idle** state, each transmitted character detected on the link is identified as the start of the frame. The link goes to **Active** state. Then the end of the frame is identified when no more character is transmitted on the link after the time interval of at least $t_{3.5}$.
- After detection of the end of frame, the CRC calculation and checking is completed. Afterwards the address field is analysed to determine if the frame is addressed to the device. If not, the frame is discarded. In order to reduce the reception processing time the address field can be analysed as soon as it is received without waiting the end of frame. In this case the CRC will be calculated and checked only if the frame is actually addressed to the Slave.

6.4 Modbus addresses

Modbus bases its data model on a series of tables with unique characteristics. The four primary tables are the followings:

- Discrete Input RO Single bit Typically provided by I/O systems
- Coils RW Single bit Can be changed by an application program
- Input registers RO 16 bit Typically provided by I/O systems
- Holding registers RW 16 bit Can be changed by an application program

The addresses in the Modbus messages are all referenced to zero. Holding register 40001 is addressed as register 0001 in the address data field of the message. The function code always specifies a holding register operation therefore the 4AAAA reference becomes implicit.

6.5 Function codes

As previously stated, the function code indicates to the Server what kind of action to perform. The function code field of a Modbus data unit is coded in one byte. Valid codes are in the range of 1 ... 255 decimal (the range 128 ... 255 is reserved and used for Exception Responses). When a message is sent from a Client to a Server device the function code field tells the Server what kind of action to perform. Function code "0" is not valid.

There are three categories of Modbus function codes, they are: **public function codes**, **user-defined function codes** and **reserved function codes**.

Public function codes are in the range 1 ... 64, 73 ... 99 and 111 ... 127; they are well defined function codes, validated by the MODBUS-IDA.org community and publicly documented; furthermore they are guaranteed to be unique. Ranges of function codes from 65 to 72 and from 100 to 110 are **user-defined function codes**: user can select and implement a function code that is not supported by the specification, it is clear that there is no guarantee that the use of the selected function code will be unique. **Reserved function codes** are not available for public use.

6.4.1 Implemented function codes

Lika Modbus inclinometers only implement public function codes, they are described hereafter.

03 Read Holding Registers

FC = 03 (03 hex) ro

This function code is used to READ the contents of a contiguous block of holding registers in a remote device; in other words, it allows to read the values set in a group of work parameters placed in order. The Request PDU specifies the starting register address and the number of registers. The number of registers to

read must be in the range of 1 to 125 (0x7D). This limit guarantees that the maximum length of the frame for response messages is always less than or equal to 252 bytes. The starting address plus the number of registers to read cannot go beyond the Modbus table, otherwise, an error condition is detected and an error message with exception code 0x02 (ILLEGAL DATA ADDRESS, refer to the "7.3 Exception response and codes" section on page 56) is transmitted in response to the request frame. In the PDU registers are addressed starting at zero. Therefore registers numbered 1-16 are addressed as 0-15.
The register data in the response message are packed as two bytes per register, with the binary contents right justified within each byte. For each register, the first byte contains the high order bits (msb) and the second contains the low order bits (lsb). This is according to big endian convention.
For the complete list of holding registers accessible using **03 Read Holding Registers** function code please refer to the "7.2.1 Description of the registers" section on page 36.

Request PDU

Slave address	1 byte	Device dependent
Function code	1 byte	**03 hex**
Starting address	2 bytes	0000 hex to FFFF hex
Quantity of registers	2 bytes	1 to 125 (007D hex)
Error check CRC	2 bytes	Frame dependent

Response PDU

Slave address	1 byte	Device dependent
Function code	1 byte	**03 hex**
Byte count	1 byte	2 x **N** *
Register value	**N** * x 2 bytes	
Error check CRC	2 bytes	Frame dependent

* N = Quantity of registers

Exception Response PDU

Slave address	1 byte	Device dependent
Error code	1 byte	**83 hex (=03 hex + 80 hex)**

Exception code	1 byte	01 or 02 or 03 or 04
Error check CRC	2 bytes	Frame dependent

Here is an example of a request to read the first five registers in the table, i.e. X and Y values of the 2-axis mode (**X angle [0001 hex]** and **Y angle [0002 hex]** registers), 180 deg and 360 deg format angle value of the 1-axis mode (**180 angle [0003 hex]** and **360 angle [0004 hex]** registers) and the temperature value (**Temperature [0005 hex]** register).

Request	
Field name	(Hex)
Slave address	**64**
Function	**03**
Starting address Hi	**00**
Starting address Lo	**01**
No. of registers Hi	**00**
No. of registers Lo	**05**
Error check CRC LSB	**DD**
Error check CRC MSB	**FC**

Response	
Field name	(Hex)
Slave address	**64**
Function	**03**
Byte count	**0A**
Register 1 value Hi	**00**
Register 1 value Lo	**C3**
Register 2 value Hi	**FF**
Register 2 value Lo	**A2**
Register 3 value Hi	**00**
Register 3 value Lo	**00**
Register 4 value Hi	**00**
Register 4 value Lo	**00**
Register 5 value Hi	**00**
Register 5 value Lo	**19**
Error check CRC LSB	**42**
Error check CRC MSB	**07**

06 Write Single Register

FC = 06 (06 hex)

This function code is used to WRITE a single holding register in a remote device. The Request PDU specifies the address of the register to be written. Registers are addressed starting at zero. Therefore register numbered 1 is addressed as 0.
The normal response (without error) is an echo of the request, returned after the register contents have been written. For the complete list of registers accessible using **06 Write Single Register** function code please refer to the "7.2.1 Description of the registers" section on page 36.

Request PDU

Slave address	1 byte	Device dependent
Function code	1 byte	**06 hex**
Register address	2 bytes	0000 hex to FFFF hex
Register value	2 bytes	0000 hex to FFFF hex
Error check CRC	2 bytes	Frame dependent

Response PDU

Slave address	1 byte	Device dependent
Function code	1 byte	**06 hex**
Register address	2 bytes	0000 hex to FFFF hex
Register value	2 bytes	0000 hex to FFFF hex
Error check CRC	2 bytes	Frame dependent

Exception Response PDU

Slave address	1 byte	Device dependent
Error code	1 byte	**86 hex (=06 hex + 80 hex)**
Exception code	1 byte	01 or 02 or 03 or 04
Error check CRC	2 bytes	Frame dependent

Here is an example of a request to change the digital filter of the inclinometer (**Filter [000F hex]** register) to value "50" (0x32 hex).

Request	
Field name	(Hex)
Slave address	**64**
Function	**06**
Register address Hi	**00**
Register address Lo	**0F**
Register value Hi	**00**
Register value Lo	**32**
Error check CRC LSB	**31**
Error check CRC MSB	**E9**

Response	
Field name	(Hex)
Slave address	**64**
Function	**06**
Register address Hi	**00**
Register address Lo	**0F**
Register value Hi	**00**
Register value Lo	**32**
Error check CRC LSB	**31**
Error check CRC MSB	**E9**

6.6 Modbus function codes not supported

The following function codes are not supported:

- 01 (0x01) Read Coils
- 02 (0x02) Read Discrete Inputs
- 04 (0x04) Read Input Registers
- 05 (0x05) Write Single Coil
- 07 (0x07) Read Exception Status
- 08 (0x08) Diagnostics
- 11 (0x0B) Get Comm Event Counter
- 12 (0x0C) Get Comm Event Log
- 15 (0x0F) Write Multiple Coils
- 16 (0x10) Write Multiple Registers
- 17 (0x11) Report Server ID
- 20 (0x14) Read File Record
- 21 (0x15) Write File Record
- 22 (0x16) Mask Write Register
- 23 (0x17) Read/Write Multiple Registers
- 24 (0x18) Read FIFO Queue
- 43 (0x2B) Encapsulated Interface Transport

7 Programming parameters

7.1 Modbus table

ADD (dec)	ADD (hex)	Register	Data type	Description	Default	NVM store	Immediately effective
0001	0x0001	X angle [0001 hex]	SIG16	2D inclin. X axis angle [-60.00° ... +60.00°]			
0002	0x0002	Y angle [0002 hex]	SIG16	2D inclin. Y axis angle [-60.00° ... +60.00°]			
0003	0x0003	180 angle [0003 hex]	SIG16	1D inclin. angle [-179.99° ... +180.00°]			
0004	0x0004	360 angle [0004 hex]	UNS16	1D inclin. angle [0.00° ... 359.99°]			
0005	0x0005	Temperature [0005 hex]	SIG16	Internal temperature [-128°C ... +127°C]			
0006	0x0006	Status [0006 hex]	UNS16	Device Status register	0		
0010	0x000A	Baud rate [000A hex]	UNS16	Baud rate	4	YES	
0011	0x000B	Parity [000B hex]	UNS16	None[1];Even[2];Odd[3]	2	YES	
0012	0x000C	Stop bits [000C hex]	UNS16	1StopBit[1];2StopBits[3]	1	YES	
0013	0x000D	Node address [000D hex]	UNS16	Inclinometer Slave address	100	YES	
0014	0x000E	TermRes [000E hex]	UNS16	OFF[1];ON[2]	1	YES	
0015	0x000F	Filter [000F hex]	UNS16	Filter level 1 ... 512	100	YES	YES
0020	0x0014	X zero [0014 hex]	UNS16	Auto Zero X angle [1]	0		YES
0021	0x0015	X preset [0015 hex]	SIG16	Preset value X angle [1]	0		YES
0022	0x0016	X offset [0016 hex]	SIG16	X angle offset value [1]	0	YES	YES
0023	0x0017	X inversion [0017 hex]	SIG16	X angle sign inversion [1] OFF[1];ON[2]	1	YES	YES
0024	0x0018	X range [0018 hex]	SIG16	X angle range [1]	30	YES	YES
0030	0x001E	Y zero [001E hex]	UNS16	Auto Zero Y angle [2]	0		YES
0031	0x001F	Y preset [001F hex]	SIG16	Preset value Y angle [2]	0		YES
0032	0x0020	Y offset [0020 hex]	SIG16	Y angle offset value [2]	0	YES	YES
0033	0x0021	Y inversion [0021 hex]	SIG16	Y angle sign inversion [2] OFF[1];ON[2]	1	YES	YES
0034	0x0022	Y range [0022 hex]	SIG16	Y angle range [2]	30[2axis] 180[1axis]	YES	YES
0040	0x0028	DevCode [0028 hex]	UNS16	Product code	57		
0041	0x0029	DevSerial [0029 hex]	UNS16	Serial number			
0042	0x002A	DevLot [002A hex]	UNS16	Lot number			
0043	0x002B	ProdDay [002B hex]	UNS16	Production day			
0044	0x002C	ProdMonth [002C hex]	UNS16	Production month			
0045	0x002D	ProdYear [002D hex]	UNS16	Production year			
0046	0x002E	ProdFw [002E hex]	UNS16	Firmware version	110		
0050	0x0032	Store all [0032 hex]	UNS16	Save user configuration	0		

0051	0x0033	Reload all [0033 hex]	UNS16	Load factory configuration	0		
0052	0x0034	ResetDev [0034 hex]	UNS16	Reboot device	0		

1 This register can be accessed for reading and writing operations in both single axis and dual axis inclinometers

2 This register can be accessed for reading and writing operations in dual axis inclinometers only

7.2 Parameters available

Hereafter the parameters available for the MODBUS inclinometers are listed and described as follows:

Parameter name [Register address]

[Register number, data types, attribute]

- The register address is expressed in hexadecimal notation.
- The register number is expressed in decimal notation.
- Attribute:
  ro = read only access
  wo = write only access
  rw = read and write access

7.2.1 Description of the registers

Registers are accessible for both writing and reading; to read the value set in a parameter use the **03 Read Holding Registers** function code (reading of multiple registers); to write a value in a parameter use the **06 Write Single Register** function code (writing of a single register); for any further information on the implemented function codes refer to the "6.4.1 Implemented function codes" section on page 30.

X angle [0001 hex]

[Register 2, Signed16, ro]

X angle [0001 hex] register stores the X tilt angle of the dual-axis mode. Value in the **X angle [0001 hex]** register is always zero in single-axis devices. **X angle [0001 hex]** value is expressed as a 2's complement 16-bit integer fixed-point value with 0.01 deg of resolution.

X angle [0001 hex] register is a RO register. The response time depends on the setting of the digital filter (**Filter [000F hex]** register).

X angle [0001 hex] value is affected by the content of the **X offset [0016 hex]** and **X inversion [0017 hex]** registers.

The value in the X range [0018 hex] register defines the maximum and minimum values for the measured angle.
X angle [0001 hex] = ($X_{RAW-ANGLE}$ + X offset [0016 hex]) * X inversion [0017 hex]

Y angle [0002 hex]

[Register 3, Signed16, ro]
Y angle [0002 hex] register stores the Y tilt angle of the dual-axis mode. Value in the Y angle [0002 hex] register is always zero for single-axis devices. Y angle [0002 hex] value is expressed as a 2's complement 16-bit integer fixed-point value with 0.01 deg of resolution.
Y angle [0002 hex] register is a RO register. The response time depends on the setting of the digital filter (Filter [000F hex] register).
Y angle [0002 hex] value is affected by the content of the Y offset [0020 hex] and Y inversion [0021 hex] registers.
The value in the Y range [0022 hex] register defines the maximum and minimum values for the measured angle.

Y angle [0002 hex] = ($Y_{RAW-ANGLE}$ + Y offset [0020 hex]) * Y inversion [0021 hex]

180 angle [0003 hex]

[Register 4, Signed16, ro]
The 180 angle [0003 hex] register stores the single-axis tilt angle in the ±180 degree format. Value in the 180 angle [0003 hex] register is always zero for dual-axis devices. The 180 angle [0003 hex] register value is expressed as a 2's complement 16-bit integer fixed point value with 0.01 deg of resolution. The 180 angle [0003 hex] register is a RO register. The response time depends on the setting of the digital filter (Filter [000F hex] register).
The 180 angle [0003 hex] register value is affected by the content of the X offset [0016 hex] and X inversion [0017 hex] registers.
The value in the X range [0018 hex] register defines the maximum and minimum values for the measured angle.

180 angle [0003 hex] = ($180_{-AngleRAW}$ + X offset [0016 hex]) * X inversion [0017 hex]

360 angle [0004 hex]

[Register 5, Unsigned16, ro]
360 angle [0004 hex] register stores the single-axis tilt angle in the 0-360 degree format. Value in the 360 angle [0004 hex] register is always zero for dual-axis devices. The 360 angle [0004 hex] register value is expressed as a 16-bit unsigned integer fixed-point value with 0.01 deg of resolution. The 360

angle [0004 hex] register is a RO register. The response time depends on the setting of the digital filter (**Filter [000F hex]** register).
360 angle [0004 hex] register value is affected by the content of the **X offset [0016 hex]** and **X inversion [0017 hex]** registers. **X range [0018 hex]** register value does not apply to **360 angle [0004 hex]** value.

360 angle [0004 hex] = ($360_{\text{-AngleRAW}}$ + **X offset [0016 hex]**)

For **360 angle [0004 hex]** register value, the **X inversion [0017 hex]** register sets the positive direction of rotation: CW or CCW.
According to factory configuration the positive rotation direction is CW (**360 angle [0004 hex]** register value increases from 0 degrees to 360 degrees, according to the stated rotation).

Temperature [0005 hex]

[Register 6, Signed16, ro]
The **Temperature [0005 hex]** register stores the internal temperature of the inclinometer. The temperature value is expressed as a 2'complement 16-bit integer value with 1°C of resolution. The temperature register is a RO register. The temperature value ranges from -128°C to +127°C (-198.4°F +260.6°F).

Status [0006 hex]

[Register 7, Unsigned16, ro]
The **Status [0006 hex]** register stores information about the functional state of the inclinometer. Reading the **Status [0006 hex]** register allows the operator to detect general error conditions. The **Status [0006 hex]** register is a RO register.
Each bit in the register represents a specific status or error condition as described hereafter. If there are no error conditions, the register contains only the information on the type of inclinometer: dual axis type (bit 3) or single axis type (bit 1).

Bit 15
Not used.

Bit 14
Not used.

Bit 13
Not used.

Bit 12
Y-Angle Err.Rng.High
The detected tilt angle for the Y axis of the inclinometer is over the higher limit value of the measuring range defined by factory default.
This status bit is valid only for dual-axis inclinometers.

Bit 11
Y-Angle Err.Rng.Low
The detected tilt angle for the Y axis of the inclinometer is below the lower limit value of the measuring range defined by factory default.
This status bit is valid only for dual-axis inclinometers.

Bit 10
X-Angle Err.Rng.High
The detected tilt angle for the X axis of the inclinometer is over the higher limit value of the measuring range defined by factory default.
This status bit is valid for both dual-axis and single-axis inclinometers.

Bit 9
X-Angle Err.Rng.Low
The detected tilt angle for the X axis of the inclinometer is below the lower limit value of the measuring range defined by factory default.
This status bit is valid for both dual-axis and single-axis inclinometers.

Bit 8
Y-Angle Err.USER Rng.High
The detected tilt angle for the Y axis of the inclinometer has reached a value over the higher limit of the user-defined measuring range set in the **Y range [0022 hex]** register. This status bit is valid only for dual-axis inclinometers.

Bit 7
Y-Angle Err.USER Rng.Low
The detected tilt angle for the Y axis of the inclinometer has reached a value less than the lower limit of the user-defined measuring range set in the **Y range [0022 hex]** register. This status bit is valid only for dual-axis inclinometers.

Bit 6
X-Angle Err.USER Rng.High
The detected tilt angle for the X axis of the inclinometer has reached a value over the higher limit of the user-defined measuring range set in the **X range [0018 hex]** register. This status bit is valid only for dual-axis inclinometers.

Bit 5

X-Angle Err.USER Rng.Low

The detected tilt angle for the X axis of the inclinometer has reached a value less than the lower limit of the user-defined measuring range set in the **X range [0018 hex]** register. This status bit is valid only for dual-axis inclinometers.

Bit 4

2-Axis NVM.Error

Error in dual-axis data stored on the NVM. When this bit is set, measured angle values are not reliable.

Bit 3

2-Axis Mode

The inclinometer is a dual-axis type device.

Bit 2

1-Axis NVM.Error

Error in single-axis data stored on the NVM. When this bit is set, measured angle values are not reliable.

Bit 1

1-Axis Mode

The inclinometer is a single axis type device.

Bit 0

Error Presence

At least one error condition is detected and active in the inclinometer.

Baud rate [000A hex]

[Register 11, Unsigned16, rw]

The **Baud rate [000A hex]** register allows to set the baud rate of the inclinometer via software. The factory value for the **Baud rate [000A hex]** register is "4" (19,200 bit/s). **Baud rate [000A hex]** register is an RW register and its value can be stored on the NVM. Admissible values for writing are in the range from "1" to "5". Each value corresponds to a baud rate value as shown in the following table:

Value (dec)	Baud rate
1	2,400 bit/s
2	4,800 bit/s
3	9,600 bit/s

4	**19,200 bit/s**
5	38,400 bit/s

To be effective a new baud rate setting must be saved to the NVM and then the inclinometer must be rebooted (software reset -**ResetDev [0034 hex]** register or power off/on cycle). When the inclinometer restarts and initializes itself, the new baud rate value is loaded from the NVM and becomes effective.
Always check whether the baud rate setting of the inclinometer complies with the baud rate of the Modbus network. If needed, change the baud rate setting of the inclinometer before connecting the device to the Modbus network. It is not possible for an inclinometer to communicate if its baud rate does not match the setting of the Modbus network. A wrong baud rate value can lead to communication errors also for other networked devices.

Parity [000B hex]

[Register 12, Unsigned16, rw]
The **Parity [000B hex]** register allows to set the parity rules for the serial communication. The default setting is "2 = Even parity".
Parity check is applied to all received characters. In case of a parity rule violation, the entire frame is deleted from the receiving buffer. **Parity [000B hex]** register is a RW register and can be stored on the NVM. Valid values are in the range from "1" to "3".

Value (dec)	Type of parity check
1	None
2	**Even**
3	Odd

When the parity check is enabled ("2 = Even" or "3 = Odd"), the number of stop bits is always 1 and this causes the characters to be 11 bit long. When the parity check is disabled ("1 = None"), the user can choose the number of stop bits (1 bit or 2 bits) by setting the **Stop bits [000C hex]** register.
This register becomes effective after the value is saved to the NVM and the inclinometer is then rebooted (software reset -**ResetDev [0034 hex]** register- or power off/on cycle). When the inclinometer restarts and initializes itself, the new parity setting is loaded from the NVM and becomes effective.
Always check whether the parity setting of the inclinometer complies with the parity type of the Modbus network. If needed, change the parity setting of the inclinometer before connecting the device to the Modbus network as it is not possible to communicate with an inclinometer having wrong parity.

Stop bits [000C hex]

[Register 13, Unsigned16, rw]

The **Stop bits [000C hex]** register allows to set the stop bit character used for the serial communication rules. The default setting is "1 stop bit" (when "2 = Even parity" is enabled, see the **Parity [000B hex]** register). **Stop bits [000C hex]** register is a RW register and its value can be stored on the NVM. **Stop bits [000C hex]** register valid values are "1" for a single stop bit and "2" for an additional stop bit (2 stop bits).

When the parity check is enabled (**Parity [000B hex]** register: "2 = Even" or "3 = Odd"), the number of stop bits is always "1" to have 11 bit long characters. When the parity check is disabled (**Parity [000B hex]** register: "1 = None"), the user can choose the number of stop bits (1 bit or 2 bits) by setting the **Stop bits [000C hex]** register.

This register becomes effective after saving to the NVM and then rebooting (software reset -**ResetDev [0034 hex]** register- or power off/on cycle). When the inclinometer restarts and initializes itself, the new stop bit setting is loaded from the NVM and becomes effective.

Always check whether the stop bits number setting of the inclinometer complies with the stop bits number setting of the Modbus network. If needed, change the stop bit number setting of the inclinometer before connecting the device to the Modbus network as it is not possible to communicate with an inclinometer having a wrong number of stop bits.

Node address [000D hex]

[Register 14, Unsigned16, rw]

The **Node address [000D hex]** register allows to set via software the address of the inclinometer to be used in the Modbus network. The default node address value is "100" (0x064 hex). The **Node address [000D hex]** register is a RW register and its value can be stored on the NVM by means of the **Store all [0032 hex]** register. Admissible values are in the range from "1" to "247" as requested by Modbus addressing rules.

A new Node Address value must be saved to the NVM to be effective and then the inclinometer must be rebooted (software reset -**ResetDev [0034 hex]** register- or power off/on cycle). When the inclinometer restarts and initializes itself, the new Node Address value is loaded from the NVM and becomes effective.

Always set a Node Address that is unique in the Modbus network otherwise problems will arise for networked devices.

TermRes [000E hex]

[Register 15, Unsigned16, rw] – STILL NOT IMPLEMENTED

The **TermRes [000E hex]** register allows the software activation or deactivation of the integrated resistor of the inclinometer for RS-485 differential BUS line termination. Line Termination integrated resistor is deactivated by the factory: the default value is "1 = OFF". **TermRes [000E hex]** register is a RW register and its value can be stored on the NVM. Admissible values are "1 = OFF" and "2 = ON".

A new **TermRes [000E hex]** setting must be saved to the NVM to be effective and then the inclinometer must be rebooted (software reset -**ResetDev [0034 hex]** register- or power off/on cycle).

When the inclinometer restarts and initializes itself, the new **TermRes [000E hex]** value is loaded from the NVM and becomes effective.

Never use more than two termination resistors for the RS-485 differential line.

Filter [000F hex]

[Register 16, Unsigned16, rw]

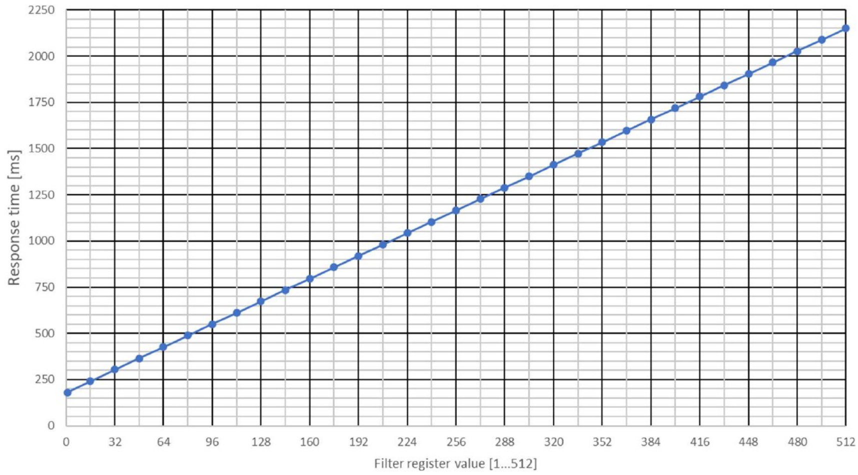
The **Filter [000F hex]** register sets the digital filter response of the inclinometer. A high filter value can reduce the noise on measured angle values, but slows down the response time to fast angle variations. On the contrary, a low filter value speeds up the response time and allows more noise to affect the measure. It is always a good practice to test different values of filter in order to find out the right compromise between noise and response time for the specific application. The **Filter [000F hex]** register is a RW register and can be stored on the NVM. Admissible values are in the range from "1" (fast response time; high noise) to "512" (slow response time; low noise).

A new value in the **Filter [000F hex]** register is immediately effective just after the write command response frame is transmitted. The value can be saved to the NVM by means of the **Store all [0032 hex]** register command to retain the new value after turning off the power supply of the inclinometer.

When the inclinometer restarts and initializes itself, the last saved **Filter [000F hex]** register value is loaded from the NVM.

The digital filter of the IXM inclinometer can be adjusted by setting the **Filter [000F hex]** register value. As stated, admissible values for the **Filter [000F hex]** register range from "1" to "512". A low value means high response time and more noise to affect the measured angle values. A high value reduces the noise on the measured angle but slows down the response time.

Response time versus Filter [000F hex] register value is shown in the following chart:



X zero [0014 hex]

[Register 21, Unsigned16, wo]

The X zero [0014 hex] register starts the auto computing function for the X offset [0016 hex] value that sets the X angle [0001 hex] value to zero. The X zero [0014 hex] register can be used also to define the offset value that is needed to set to zero the 180 angle [0003 hex] and 360 angle [0004 hex] registers in the single-axis inclinometers. This function is useful after the installation of the device, in order to compensate the mechanical offsets. To start the X zero [0014 hex] function of the inclinometer, write the ASCII values corresponding to the string "ZX" (0x5A58 hex or 23'128 dec) in the X zero [0014 hex] register. The inclinometer will calculate the needed X offset [0016 hex] value using the current raw X axis tilt angle of the inclinometer. If an X offset [0016 hex] value is already defined, this value is not considered. X zero [0014 hex] register is a a WO register, a reading command will always return a zero value.

The X offset [0016 hex] value for single-axis inclinometers is calculated as follows:

X offset [0016 hex]1 = - (360 angle [0004 hex] – X offset [0016 hex]$_{PREVIOUS}$)

The X offset [0016 hex] value for dual-axis inclinometers is calculated as follows:

X offset [0016 hex]2 = - (X angle [0001 hex] – X offset [0016 hex]$_{PREVIOUS}$)

Note that the X angle [0001 hex] value in the above expression includes the current X inversion [0017 hex] setting so the sign of the new value in the X offset [0016 hex] register takes into account the current sign of the X axis. For this reason, always use the X zero [0014 hex] function after the activation of the X inversion [0017 hex]. If you start the X zero [0014 hex] function before activating the inversion of the X axis sign, a calculation error will occur in the tilt angle for X axis.

For dual-axis inclinometers, the X offset [0016 hex] register value must be less than or equal to the half of the X axis measuring range of the inclinometer as set in the X range [0018 hex] register (-½ * X range [0018 hex] ... +½ * X range [0018 hex]). If the calculated X offset [0016 hex] value results to be out of this range of values, the inclinometer transmits an error response message with exception code 0x03.

For single-axis devices, the X offset [0016 hex] value must be in the range of -180,00 deg to +180,00 deg.
A new value of the X offset [0016 hex] register is immediately effective just after the X zero [0014 hex] command response frame is transmitted. The value can be saved to the NVM by means of the Store all [0032 hex] command to retain the new value after turning off the power supply of the inclinometer. When the inclinometer restarts and initializes itself the last saved X offset [0016 hex] register value is loaded from the NVM and used to calculate the measured X angle [0001 hex] value.

X preset [0015 hex]

[Register 22, Signed16, wo]
The X preset [0015 hex] register starts the auto computing function for the X offset [0016 hex] value that sets the X angle [0001 hex] value equal to the value of frame. The X preset [0015 hex] register can be used also to define the offset value for 180 angle [0003 hex] and 360 angle [0004 hex] registers of single-axis inclinometers. This function is very useful after installation, to get specific X angle [0001 hex] value when the inclinometer detects the desired position. The X preset [0015 hex] register value must be a 2's complement 16-bit integer fixed-point value with 0.01 deg of resolution. The inclinometer will calculate the new X offset [0016 hex] value according to raw X axis tilt angle and X preset [0015 hex] values. If an X offset [0016 hex] value is already defined, this value is not considered for the new X offset [0016 hex] value. X preset [0015 hex] register is a WO register, a reading command will always return a zero value.

The X offset [0016 hex] value for single-axis inclinometers is calculated as follows:

X offset [0016 hex] = [X preset [0015 hex] - (360 angle [0004 hex] – X offset [0016 hex]$_{PREVIOUS}$)]

The X offset [0016 hex] value for dual-axis inclinometers is calculated as follows:

X offset [0016 hex] = [X preset [0015 hex] - (X angle [0001 hex] – X offset [0016 hex]$_{PREVIOUS}$)]

Note that the X angle [0001 hex] value in the above expression includes the current X inversion [0017 hex] setting, therefore the sign of the new value in X offset [0016 hex] takes into account the current sign of the X axis. For this reason, set the X preset [0015 hex] register always after the activation of X inversion [0017 hex]. If you set the X preset [0015 hex] register before activating the inversion of the X axis sign, a calculation error will occur.

For dual-axis inclinometers, the X offset [0016 hex] register value must be less than or equal to the half of the X axis measuring range of the inclinometer as set in the X range [0018 hex] register (-½ * X range [0018 hex] ... +½ * X range [0018 hex]). If the calculated X offset [0016 hex] value results to be out of this range of values, the inclinometer transmits an error response message with exception code 0x03.

For single-axis devices, the X offset [0016 hex] value must be in the range of -180.00 deg to +180.00 deg.
A new X offset [0016 hex] register value is immediately effective just after setting the X preset [0015 hex] register. The X offset [0016 hex] value can be saved to the NVM by means of the Store all [0032 hex] command to retain the new value after turning off the power supply of the inclinometer.
When the inclinometer restarts and initializes itself, the last saved X offset [0016 hex] register value is loaded from the NVM and used to calculate the measured X angle [0001 hex] value.

X offset [0016 hex]

[Register 23, Signed16, rw]

The X offset [0016 hex] register stores the offset value applied to the measured raw X axis tilt of the inclinometer. The X offset [0016 hex] register value is a 2's complement 16-bit integer fixed-point value with 0.01 deg resolution. The X offset [0016 hex] register is a RW register.

For dual-axis inclinometers, the X offset [0016 hex] register value must be in the range of -½ * X range [0018 hex] to +½ * X range [0018 hex]. For single-axis devices, the X offset [0016 hex] register value must be in the range of -180.00 deg to +180.00 deg. If you set an X offset [0016 hex] value that is out of range, the inclinometer will transmit an error message containing the exception code 0x03 in response to the writing single register command frame.

The X offset [0016 hex] register value is immediately effective just after the write command response frame is transmitted by the inclinometer. The X offset [0016 hex] value can be saved to the NVM by means of the Store all [0032 hex] command to retain the new value after turning off the power supply of the inclinometer. When the inclinometer restarts and enters the initialization phase, the last saved X offset [0016 hex] register value is loaded from the NVM and used to calculate the measured X angle [0001 hex] value.

X inversion [0017 hex]

[Register 24, Signed16, rw]

The X inversion [0017 hex] register activates or deactivates the inversion of the X axis sign for the measured tilt angle of the inclinometer. The default setting is X inversion [0017 hex] = "1 = OFF". The X inversion [0017 hex] register is a RW register. Valid values are "1 = OFF" to deactivate the inversion and "2 = ON" to activate the inversion. Axis inversion function is performed before applying the X offset [0016 hex] register value.

In dual-axis mode and single axis 180 angle [0003 hex] mode, when you activate the X inversion [0017 hex] option, you will invert the sign of the measured angle.

For the single-axis 360 angle [0004 hex] mode, when you activate the X inversion [0017 hex] option, you will change the positive rotation direction from CW (1 = OFF) to CCW (2 = ON). Activate the X inversion [0017 hex] option always before starting the X zero [0014 hex] function or setting the X preset [0015 hex] register. Otherwise a calculation error will occur.

The X inversion [0017 hex] register value is immediately effective just after setting. The X inversion [0017 hex] value can be saved to the NVM by means of the Store all [0032 hex] command to retain the new value after turning off the power supply of the inclinometer. When the inclinometer restarts and initializes itself, the last saved X inversion [0017 hex] register value is loaded from the NVM and used to calculate the measured X angle [0001 hex] value.

X range [0018 hex]

[Register 25, Signed16, rw]

The X range [0018 hex] register contains the user-defined range for detected tilt angle in the X axis of the inclinometer. IXM inclinometers are factory configured to a default measuring range (e.g.: ±30 deg or ±60 deg) depending on the model. This range defines the maximum inclination angle that the inclinometers can detect on the X axis. If the measure tilt angle is out of the factory range, the X angle [0001 hex] register is "clamped" to the factory range value (positive or negative, depending on the tilt angle inclinometer) and an error condition is highlighted in the Status [0006 hex] register.

For single-axis devices the error bits that signal the overrun/underrun conditions in the range are the two factory range error bits only (see X-Angle Err.Rng.High and X-Angle Err.Rng.Low bits in the Status [0006 hex] register). To adjust the inclinometer to different purposes it is also possible to set the user-defined X range [0018 hex] register value to further limit the maximum measured angle value. When no user-defined X range [0018 hex] value is defined (e.g.: after a Reload all [0033 hex] command), the register contains the factory range value for X axis. The factory range for the X axis of the inclinometer has always a higher priority over the user-defined X range [0018 hex] register value. This must be considered when using a user-defined X range [0018 hex] value in combination with an X offset [0016 hex] register value. If the measured tilt angle reaches first the factory-defined X axis range value, the X angle [0001 hex] register value is "clamped" to the factory-defined X axis range value (positive or negative depending on the tilt angle of the inclinometer) and an error condition is highlighted in the Status [0006 hex] register.

X range [0018 hex] register is a RW register.

For dual-axis devices, the user-defined X range [0018 hex] register value must be greater than or equal to 1 deg and less than or equal to the X axis factory range value. For single-axis devices, the user-defined X range [0018 hex] register value must be in the range of 1 deg to 180 deg. For a single-axis device, the X range [0018 hex] limit value applies only to the 180 angle [0003 hex] register value. The 360 angle [0004 hex] register value is not affected by the X range [0018 hex] register value. The value to be set in the X range [0018 hex] register must be expressed as a 16-bit unsigned integer value with 1 deg of resolution. Attempts to set invalid values in the X range [0018 hex] register will cause the inclinometer to transmit an error message containing the exception code 0x03 in response to the writing single register command frame.

The **X range [0018 hex]** register value is immediately effective just after the write single register command response frame is transmitted by the inclinometer. The **X range [0018 hex]** value can be saved to the NVM by means of the **Store all [0032 hex]** command to retain the new value after turning off the power supply of the inclinometer. When the inclinometer restarts and initializes itself, the last saved **X range [0018 hex]** register value is loaded from the NVM and used to control the validity of the measured **X angle [0001 hex]** value.

Y zero [001E hex]

[Register 31, Unsigned16, wo]

The **Y zero [001E hex]** register starts the auto computing function for the **Y offset [0020 hex]** value that sets the **Y angle [0002 hex]** value to zero. This function is useful after the installation of the device, in order to compensate the mechanical offsets. To start the **Y zero [001E hex]** function of the inclinometer, write the ASCII values corresponding to the string "ZY" (0x5A59 hex or 23'129 dec) in the **Y zero [001E hex]** register. The inclinometer will calculate the needed **Y offset [0020 hex]** value using the current raw Y axis tilt angle of the inclinometer. If an **Y offset [0020 hex]** value is already defined, this value is not considered. **Y zero [001E hex]** register is a WO register, a reading command will always return a zero value.

The **Y offset [0020 hex]** value is not available for single-axis devices.

The **Y offset [0020 hex]** value for dual-axis inclinometers is calculated as follows:

Y offset [0020 hex]2 = - (**Y angle [0002 hex]** – **Y offset [0020 hex]**$_{\text{PREVIOUS}}$)

Note that the **Y angle [0002 hex]** value in the above expression includes the current **Y inversion [0021 hex]** setting so sign of the new value in the **Y offset [0020 hex]** register takes into account the current sign of the Y axis. For this reason, use the **Y zero [001E hex]** function always after the activation of the **Y inversion [0021 hex]**. If you start the **Y zero [001E hex]** function before activating the inversion of the Y axis sign, a calculation error will occur in the tilt angle for Y axis.

For dual-axis inclinometers, the **Y offset [0020 hex]** register value must be less than or equal to the half of the Y axis measuring range of the inclinometer as set in the **Y range [0022 hex]** register (-½ * **Y range [0022 hex]** ... +½ * **Y range [0022 hex]**). If the calculated **Y offset [0020 hex]** value results to be out of this range of values, the inclinometer transmits an error response message with exception code 0x03.

A new **Y offset [0020 hex]** register value is immediately effective just after the **Y zero [001E hex]** command response frame is transmitted. The value can be saved to the NVM by means of the Store all [0032 hex] command to retain the new value after turning off the power supply of the inclinometer. When the inclinometer restarts and initializes itself, the last saved Y offset [0020 hex] register value is loaded from the NVM and used to calculate the measured Y angle [0002 hex] value.

Y preset [001F hex]

[Register 32, Signed16, wo]

The Y preset [001F hex] register starts the auto computing function for the Y offset [0020 hex] value that sets the Y angle [0002 hex] value equal to the value of the frame. This function is very useful after installation, to get specific Y angle [0002 hex] value when the inclinometer detects the desired position. The Y preset [001F hex] register value must be a 2's complement 16-bit integer fixed-point value with 0.01 deg of resolution. The inclinometer will calculate the new Y preset [001F hex] value according to the raw Y axis tilt angle and Y preset [001F hex] values. If an Y offset [0020 hex] value is already defined, this value is not considered for the new Y offset [0020 hex] value. Y preset [001F hex] register is a WO register, a reading command will always return a zero value.

The Y offset [0020 hex] value for single-axis inclinometers is not available.

The Y offset [0020 hex] value for dual-axis inclinometers is calculated as follows:

Y offset [0020 hex] = [Y preset [001F hex] - (Y angle [0002 hex] – Y offset [0020 hex]$_{PREVIOUS}$)]

Note that the Y angle [0002 hex] value in the above expression includes the current Y inversion [0021 hex] setting, therefore the sign of the new Y offset [0020 hex] value takes into account the current sign of the Y axis. For this reason, always set the Y preset [001F hex] register after the activation of the Y inversion [0021 hex]. If you set the Y preset [001F hex] register before activating the inversion of the Y axis sign, a calculation error will occur.

For dual-axis inclinometers, the Y offset [0020 hex] register value must be less than or equal to the half of the Y axis measuring range as set in the Y range [0022 hex] register (-½ * Y range [0022 hex] … +½ * Y range [0022 hex]). If the calculated Y offset [0020 hex] value results to be out of this range of values, the inclinometer transmits an error response message with exception code 0x03.

A new Y offset [0020 hex] register value is immediately effective just after setting the Y preset [001F hex] register. The Y offset [0020 hex] value can be saved to the NVM by means of the Store all [0032 hex] command to retain the new value after turning off the power supply of the inclinometer. When the

inclinometer restarts and initializes itself, the last saved Y offset [0020 hex] register value is loaded from the NVM and used to calculate the measured Y angle [0002 hex] value.
The Y preset [001F hex] register is available only for dual-axis inclinometers.

Y offset [0020 hex]

[Register 33, Signed16, rw]
The Y offset [0020 hex] register stores the offset value applied to the measured raw X axis tilt value of the inclinometer. The Y offset [0020 hex] register value is a 2's complement 16-bit integer fixed-point value with 0.01 deg resolution. The Y offset [0020 hex] register is a RW register.
For dual-axis inclinometers, the Y offset [0020 hex] register value must be in the range of -½ * Y range [0022 hex] to +½ * Y range [0022 hex]. If you set an Y offset [0020 hex] value that is out of range, the inclinometer will transmit an error message containing the exception code 0x03 in response to the writing single register command frame.
The Y offset [0020 hex] register value is immediately effective just after the write command response frame is transmitted by the inclinometer. The Y offset [0020 hex] value can be saved to the NVM by means of the Store all [0032 hex] command to retain the new value after turning off the power supply of the inclinometer. When the inclinometer restarts and enters the initialization phase, the last saved Y offset [0020 hex] register value is loaded from the NVM and used to calculate the measured Y angle [0002 hex] value.
The Y offset [0020 hex] register is available only for dual-axis inclinometers.

Y inversion [0021 hex]

[Register 34, Signed16, rw]
The Y inversion [0021 hex] register activates or deactivates the inversion of the Y axis sign for the measured tilt angle of the inclinometer. The default setting is Y inversion [0021 hex] = "1 = OFF". The Y inversion [0021 hex] register is a RW register. Valid values are "1 = OFF" to deactivate the inversion and "2 = ON" to activate the inversion. Axis inversion function is performed before applying the Y offset [0020 hex] register value.
In dual-axis mode, when you activate the Y inversion [0021 hex] option, you will invert the sign of the measured angle.
The Y inversion [0021 hex] register value is immediately effective just after setting. The Y inversion [0021 hex] value can be saved to the NVM by means of the Store all [0032 hex] command to retain the new value after turning off the power supply of the inclinometer. When the inclinometer restarts and initializes itself, the last saved Y inversion [0021 hex] register value is loaded from the NVM and used to calculate the measured Y angle [0002 hex] value.

The **Y inversion [0021 hex]** register is available only for dual-axis inclinometers.

Y range [0022 hex]

[Register 35, Signed16, rw]

The **Y range [0022 hex]** register contains the user-defined range for detected tilt angle in the X axis of the inclinometer. IXM inclinometers are factory configured to a default measuring range (e.g.: ±30 deg or ±60 deg) depending on the model. This range defines the maximum inclination angle that the inclinometers can detect on the Y axis. If the measure tilt angle is out of the factory range, the **Y angle [0002 hex]** register is "clamped" to the factory range value (positive or negative, depending on the tilt angle of the inclinometer) and an error condition is highlighted in the **Status [0006 hex]** register.

To adjust the inclinometer to different purposes it is also possible to set the user-defined **Y range [0022 hex]** register value to further limit the maximum measured angle value. When no user-defined **Y range [0022 hex]** value is defined (e.g.: after a **Reload all [0033 hex]** command), the register contains the factory range value for the X axis. The factory range for the X axis of the inclinometer has always a higher priority over the user-defined **Y range [0022 hex]** register value. This must be considered when using a user-defined **Y range [0022 hex]** value in combination with an **Y offset [0020 hex]** register value. If the measured tilt angle reaches first the factory-defined Y axis range value, the **Y range [0022 hex]** register value is "clamped" to the factory-defined Y axis range value (positive or negative depending on the tilt angle of the inclinometer) and an error condition is highlighted in the **Status [0006 hex]** register.

Y range [0022 hex] register is a RW register.

For dual-axis devices, the user-defined **Y range [0022 hex]** register value must be greater than or equal to 1 deg and less than or equal to the Y axis factory range value. The value to be written in the **Y range [0022 hex]** register must be expressed as a 16-bit unsigned integer value with 1 deg of resolution. Attempts to set invalid values in the **Y range [0022 hex]** register will cause the inclinometer to transmit an error message containing the exception code 0x03 in response to the writing single register command frame.

The **Y range [0022 hex]** register value is immediately effective just after the write single register command response frame is transmitted by the inclinometer. The **Y range [0022 hex]** value can be saved to the NVM by means of the **Store all [0032 hex]** command to retain the new value after turning off the power supply of the inclinometer. When the inclinometer restarts and initializes itself, the last saved **Y range [0022 hex]** register value is loaded from the NVM and used to control the validity of the measured **Y angle [0002 hex]** value.

The **Y range [0022 hex]** register is available only for dual-axis inclinometers.

DevCode [0028 hex]

[Register 41, Unsigned16, ro]
The **DevCode [0028 hex]** register contains the factory identification code for the specific model of the IXM inclinometer. This information is useful for any technical support question. **DevCode [0028 hex]** register is a RO register.

DevSerial [0029 hex]

[Register 42, Unsigned16, ro]
The **DevSerial [0029 hex]** register contains the serial number. This information is useful for any technical support question. **DevSerial [0029 hex]** register is a RO register.

DevLot [002A hex]

[Register 43, Unsigned16, ro]
The **DevLot [002A hex]** register contains the production lot number. This information is useful for traceability purpose. **DevLot [002A hex]** register is a RO register.

ProdDay [002B hex]

[Register 44, Unsigned16, ro]
The **ProdDay [002B hex]** register contains the day of the production in the month. This information is useful for traceability purpose. **ProdDay [002B hex]** register is a RO register.

ProdMonth [002C hex]

[Register 45, Unsigned16, ro]
The **ProdMonth [002C hex]** register contains the month of the production in the year. This information is useful for traceability purpose. **ProdMonth [002C hex]** register is a RO register.

ProdYear [002D hex]

[Register 46, Unsigned16, ro]
The **ProdYear [002D hex]** register contains the year of the production. This information is useful for traceability purpose. **ProdYear [002D hex]** register is a RO register.

ProdFw [002E hex]

[Register 47, Unsigned16, ro]

The ProdFw [002E hex] register contains the firmware version of the inclinometer. This information is useful for any technical support question. ProdFw [002E hex] register is a RO register.

Store all [0032 hex]

[Register 51, Unsigned16, wo]

The Store all [0032 hex] register allows to save the configuration of the inclinometer to the non volatile memory (NVM). To save the current configuration on the NVM, write the ASCII values corresponding to the string "ST" (0x5354 hex or 21'332 dec) in the Store all [0032 hex] register. Once the saving process is completed the inclinometer sends the response message.

Store all [0032 hex] register is a WO register. Read holding register command will always return zero.

The registers that are stored on the NVM are:

- Baud rate [000A hex]
- Parity [000B hex]
- Stop bits [000C hex]
- Node address [000D hex]
- TermRes [000E hex]
- Filter [000F hex]
- X offset [0016 hex]
- X inversion [0017 hex]
- X range [0018 hex]
- Y offset [0020 hex] (only 2 axis inclinometers)
- Y inversion [0021 hex] (only 2 axis inclinometers)
- Y range [0022 hex] (only 2 axis inclinometers)

Reload all [0033 hex]

[Register 52, Unsigned16, wo]

The Reload all [0033 hex] register allows to restore to the factory values the configuration of the inclinometer in the non volatile memory (NVM). All the user-defined values and settings will be lost and the inclinometer is configured back to the default configuration. Be careful especially if the device is connected to an existing Modbus network, that uses a serial configuration different from default or has nodes with the same default node address (e.g.: 100 dec 0x64 hex).

To launch the load factory configuration process, write the ASCII values corresponding to the string "LD" (0x4C44 hex or 19'524 dec) in the Reload all

[0033 hex] register. When the restoring factory configuration process ends, the inclinometer sends the response message for the single register command.
The Reload all [0033 hex] register is a WO register. A read holding registers command will always return a zero value for the Reload all [0033 hex] register.
The parameters that are restored to the factory values by means of the Reload all [0033 hex] register process are:

- Baud rate [000A hex]
- Parity [000B hex]
- Stop bits [000C hex]
- Node address [000D hex]
- TermRes [000E hex]
- Filter [000F hex]
- X offset [0016 hex]
- X inversion [0017 hex]
- X range [0018 hex]
- Y offset [0020 hex] (only 2 axis inclinometers)
- Y inversion [0021 hex] (only 2 axis inclinometers)
- Y range [0022 hex] (only 2 axis inclinometers)

ResetDev [0034 hex]

[Register 53, Unsigned16, wo]
The ResetDev [0034 hex] register allows the software reboot of the inclinometer. To trigger the reboot process of the inclinometer write the ASCII values corresponding to the string "RS" (0x5253 hex or 21'075 dec) in the ResetDev [0034 hex] register. When the write holding register command frame is received, the inclinometer sends the response message and then starts the rebooting process. Configuration values are retained only if they have been previously saved to the NVM. All configuration registers values are initialized to the last saved values that can be loaded from the NVM.
The ResetDev [0034 hex] register is a WO register. A read holding registers command will always return a zero value for the ResetDev [0034 hex] register.

7.3 Exception response and codes

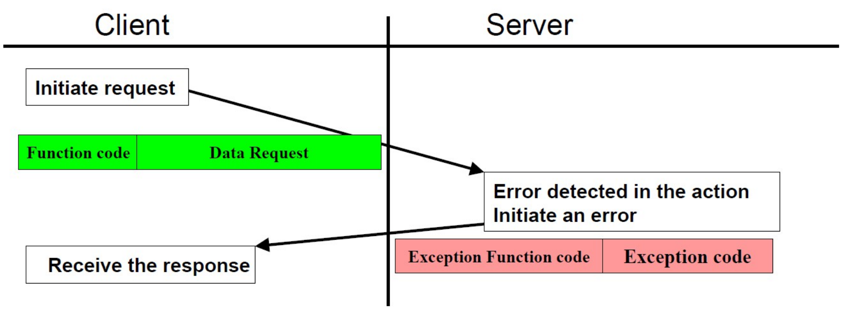
When a Client device sends a request to a Server device it expects a normal response. One of four possible events can occur from the Master's query.

- If the Server device receives the request without a communication error and can handle the query normally, it returns a normal response.
- If the Server does not receive the request due to a communication error, no response is returned. The client program will eventually process a timeout condition for the request.
- If the Server receives the request, but detects a communication error, no response is returned. The Client program will eventually process a timeout condition for the request.
- If the Server receives the request without a communication error, but cannot handle it (for example, if the request is to read a non-existent output or register), the Server will return an **exception response** informing the Client about the nature of the error.

The exception response message has two fields that differentiate it from a normal response:

FUNCTION CODE FIELD: in a normal response, the Server echoes the function code of the original request in the function code field of the response. All function codes have a most significant bit (msb) of 0 (their values are all below 80 hexadecimal). In an exception response, the Server sets the msb of the function code to 1. This makes the function code value in an exception response exactly 80 hexadecimal higher than the value would be for a normal response. With the function code's msb set, the client's application program can recognize the exception response and can examine the data field for the exception code.

DATA FIELD: in a normal response, the Server may return data or statistics in the data field (any information that was requested in the request). In an exception code, the Server returns an exception code in the data field. This defines the Server condition that caused the exception.



NOTE
Please note that here follows the list of the exception codes indicated by MODBUS but not necessarily supported by the manufacturer.

MODBUS Exception codes		
Code	**Name**	**Meaning**
01	ILLEGAL FUNCTION	The function code received in the query is not an allowable action for the server. This may be because the function code is only applicable to newer devices, and was not implemented in the unit selected. It could also indicate that the server is in the wrong state to process a request of this type, for example because it is not configured and is being asked to return register values.
02	ILLEGAL DATA ADDRESS	The data address received in the query is not an allowable address for the server. More specifically, the combination of reference number and transfer length is invalid. For a controller with 100 registers, the PDU addresses the first register as 0, and the last one as 99. If a request is submitted with a starting register address of 96 and a quantity of registers of 4, then this request will successfully operate (address-wise at least) on registers 96, 97, 98, 99. If a request is submitted with a starting register address of 96 and a quantity of registers of 5, then this request will fail with Exception Code 0x02 "Illegal Data Address" since it attempts to operate on registers 96, 97, 98, 99 and 100, and there is no register with address 100.
03	ILLEGAL DATA VALUE	A value contained in the query data field is not an allowable value for server. This indicates a fault in the structure of the remainder of a complex request, such as that the implied length is incorrect. It specifically does NOT mean that a data item submitted for storage in a register has a value outside the expectation of the application program, since the MODBUS protocol is unaware of the significance of any particular value of any particular register.
04	SERVER DEVICE FAILURE	An unrecoverable error occurred while the server was attempting to perform the requested action.

05	ACKNOWLEDGE	Specialized use in conjunction with programming commands. The server has accepted the request and is processing it, but a long duration of time will be required to do so. This response is returned to prevent a timeout error from occurring in the client. The client can next issue a Poll Program Complete message to determine if processing is completed.
06	SERVER DEVICE BUSY	Specialized use in conjunction with programming commands. The server is engaged in processing a long-duration program command. The client should retransmit the message later when the server is free.
08	MEMORY PARITY ERROR	Specialized use in conjunction with function codes 20 and 21 and reference type 6, to indicate that the extended file area failed to pass a consistency check. The server attempted to read record file, but detected a parity error in the memory. The client can retry the request, but service may be required on the server device.
0A	GATEWAY PATH UNAVAILABLE	Specialized use in conjunction with gateways, indicates that the gateway was unable to allocate an internal communication path from the input port to the output port for processing the request. Usually means that the gateway is misconfigured or overloaded.
0B	GATEWAY TARGET DEVICE FAILED TO RESPOND	Specialized use in conjunction with gateways, indicates that no response was obtained from the target device. Usually means that the device is not present on the network.

For any information on the available exception codes and their meaning refer to the "MODBUS Exception Responses" section on page 47 of the "MODBUS Application Protocol Specification V1.1b3" document.

8 Programming examples

Hereafter are some examples of both reading and writing parameters. All values are expressed in hexadecimal notation.

IXM1 inclinometer is installed (1 axis mode)
The Master device is set up as follows: 19,200 bit/s, Even parity, 1 stop bit; the Node Address is set to factory default = "100" (0x64 hex)

8.1 Reading the Status [0006 hex] register

Request

Request of the Master to read the **Status [0006 hex]** register

64 03 00 06 00 01 6D FE

Response

64 03 02 00 02 75 8D

Status [0006 hex] = 0x0002 = 0010_2 = (bit 0 = 0 = **Error Presence** status is NOT active); (bit 1 = 1 = **1-Axis Mode** status is active)

8.2 Reading the first five registers in the table

Request

Request of the Master to read the registers from **X angle [0001 hex]** to **Temperature [0005 hex]**

64 03 00 01 00 05 DD FC

Response

64 03 0A 00 00 00 00 F8 71 85 11 00 1C B8 60

X angle [0001 hex] = 0x0000 because the Slave is a 1 axis inclinometer
Y angle [0002 hex] = 0x0000 because the Slave is a 1 axis inclinometer
180 angle [0003 hex] = 0xF871 = - 19.35 deg in ±180 deg
360 angle [0004 hex] = 0x8511 = 340.65 deg in 0...360 deg format
Temperature [0005 hex] = 0x001c= +28°C internal temperature of the Slave

8.3 Setting the Filter [000F hex] register

Request

Request of the Master to set the **Filter [000F hex]** register to "300" (0x12C hex)

64 06 00 0F 01 2C B0 71

Response

64 06 00 0F 01 2C B0 71

The new value has been set properly in the register

8.4 Setting the Baud rate [000A hex] register

Request

Request of the Master to set the **Baud rate [000A hex]** register to the sixth value (it does not exist!)

64 06 00 0A 00 06 20 3F

Response

64 86 03 20 3F

A error occurred while executing the write single register command: exception code 0x03 (Illegal Data Value). The 6th value for the **Baud rate [000A hex]** register does not exist, the max value for the register is "5", corresponding to 38,400 bit/s.

8.5 Setting the Baud rate [000A hex] register

Request

Request of the Master to set the **Baud rate [000A hex]** register to "3" = 9,600 bit/s

64 06 00 0A 00 03 E0 3C

Response

64 06 00 0A 00 03 E0 3C

The new value has been set properly in the register

8.6 Setting the Node address [000D hex] register

Request

Request of the Master to set the **Node address [000D hex]** register to "32" (0x0020)

64 06 00 DD 00 20 11 DD

Response

64 86 02 11 DD

An error occurred while executing the write single register command: exception code 0x02 (Illegal Data Address). The register address 0x00DD (221) does not exist in the Modbus table of the inclinometer.

8.7 Setting the Node address [000D hex] register

Request

Request of the Master to set the **Node address [000D hex]** register to "32" (0x0020)

64 06 00 0D 00 20 10 24

Response

64 06 00 0D 00 20 10 24

The new value has been set properly in the register

8.8 Activating the inversion in the X inversion [0017 hex] register

Request

Request of the Master to activate the inversion ("2 = ON") in the **X inversion [0017 hex]** register (please always activate the axis inversion option before the zero or preset command; otherwise angle values can be miscalculated)

64 06 00 17 00 02 B1 FA

Response

64 06 00 17 00 02 B1 FA

The inversion has been activated properly in the register

8.9 Reading the 180 angle [0003 hex] and 360 angle [0004 hex] registers

Request

Request of the Master to read the **180 angle [0003 hex]** and **360 angle [0004 hex]** registers

64 03 00 03 00 02 3D FE

Response

64 03 04 07 91 07 91 5C 30

180 angle [0003 hex] > 180 angle = 0x0791 = +19.37 deg

360 angle [0004 hex] > 360 angle = 0x0791 = 19.37 deg

For tilt angle values comprised between 0 deg and +180 deg, **180 angle [0003 hex]** and **360 angle [0004 hex]** registers return the same value.

8.10 Setting the X preset [0015 hex] register

Request

Request of the Master to set the **X preset [0015 hex]** register to "4500" (0x1194): the angle is preset to +45.00 deg (4500 = 0x1194 hex)

64 06 00 15 11 94 9C 04

Response

64 06 00 15 11 94 9C 04

The new value has been set properly in the register

8.11 Reading the X offset [0016 hex] register

Request

Request of the Master to read the **X offset [0016 hex]** register

64 03 00 16 00 01 6C 3B

Response

64 03 02 0A 06 72 EE

X offset [0016 hex] = 0x0A06 = +25.66 deg

Calculations: +45.00 – 19.34 = 25.66 deg

8.12 Reading the 180 angle [0003 hex] and 360 angle [0004 hex] registers

Request

Request of the Master to read the **180 angle [0003 hex]** and **360 angle [0004 hex]** registers

64 03 00 03 00 02 3D FE

Response

64 03 04 FD 86 8A 26 F9 CA

180 angle [0003 hex] > 180 angle = 0xFD86 = -6.34 deg

360 angle [0004 hex] > 360 angle = 0x8A26 = 353.66 deg

8.13 Setting the X range [0018 hex] register

Request

Request of the Master to set the **X range [0018 hex]** register to "45" (±45 deg = 0x2D)

64 06 00 18 00 2D C0 25

Response

64 06 00 18 00 2D C0 25

The new value has been set properly in the register

8.14 Saving the current configuration

Request

Request of the Master to save the current configuration of the inclinometer by setting the value 0x5354 = "ST" in ASCII code in the **Store all [0032 hex]** register

64 06 00 32 53 54 1C FF

Response

64 06 00 32 53 54 1C FF

The value has been set properly in the register

8.15 Resetting the inclinometer

Request

Request of the Master to reset the inclinometer by setting the value 0x5253 = "RS" in ASCII code in the **ResetDev [0034 hex]** register

64 06 00 34 52 53 BC AC

Response

64 06 00 34 52 53 BC AC

The value has been set properly in the register. The inclinometer replies and then reboots

8.16 Reading the 180 angle [0003 hex] and 360 angle [0004 hex] registers

Request

Request of the Master to read the **180 angle [0003 hex]** and **360 angle [0004 hex]** registers by using wrong communication settings

64 03 00 03 00 02 3D FE

Response

NO RESPONSE. WRONG COMMUNICATION SETTINGS

The configuration has changed. Now the Master device is set as follows: 9,600 bit/s, Even parity, 1 stop bit; the new Node Address is "32" (0x20 hex)

8.17 Reading the configuration of the communication parameters and the Filter [000F hex] register

Request

Request of the Master to read the configuration of the communication parameters and the value of the **Filter [000F hex]** register

20 03 00 0A 00 06 E3 7B

Response

20 03 0C 00 03 00 02 00 01 00 20 00 01 01 2C A4 C6

Baud rate [000A hex] > Baud Rate = 0x0003 > 9600 bit/s

Parity [000B hex] > Parity = 0x0002 > Even parity

Stop bits [000C hex] > Stop Bits = 0x0001 > 1 Stop Bit

Node address [000D hex] > Node Address = 0x0020 = 32

TermRes [000E hex] > Termination Resistor = 0x0001 > Disabled (currently this register is not implemented)

Filter [000F hex] > Filter = 0x012C = 300

8.18 Reading the configuration of the measure axis registers

Request

Request of the Master to read the configuration of the measure axis registers

20 03 00 14 00 05 C3 7C

Response

20 03 0A 00 00 00 00 0A 06 00 02 00 2D B0 20

X zero [0014 hex] > X Zero = 0x0000 > the register is a WO register

X preset [0015 hex] > X Preset = 0x0000 > the register is a WO register

X offset [0016 hex] > X Offset = 0x0A06 = +25.66 deg

X inversion [0017 hex] > X Inversion = 0x0002 > the inversion of the axis is enabled

X range [0018 hex] > X Range = 0x002D = 45 deg

8.19 Reading the last fifteen registers in the table

Request

Request of the Master to read the last fifteen registers in the table starting from register **DevCode [0028 hex]**, so the last register to be read will be the register 0037 hex

20 03 00 28 00 0F 83 77

Response

20 83 02 83 77

An error occurred while executing the read holding registers command: exception code 0x02 (Illegal Data Address). The last register in the Modbus table of the inclinometer is the **ResetDev [0034 hex]** register.

8.20 Reading the registers 180 angle [0003 hex], 360 angle [0004 hex], Temperature [0005 hex] and Status [0006 hex]

Request

Request of the Master to read the **180 angle [0003 hex]** and **360 angle [0004 hex]** registers as well as the internal temperature (**Temperature [0005 hex]** register) and the functional state of the inclinometer (**Status [0006 hex]** register)

20 03 00 03 00 04 B2 B8

Response

20 03 08 FD 8F 8A 2F 00 1C 00 02 00 55

180 angle [0003 hex] > 180 angle = 0xFD8F = -6.25 deg

360 angle [0004 hex] > 360 angle = 0x8A2F = 353.75 deg

Temperature [0005 hex] > Temperature = 0x001C = +28°C

Status [0006 hex] > Status register = 0010_2 = (bit 0 = 0 = **Error Presence** status is NOT active); (bit 1 = 1 = **1-Axis Mode** status is active)

Now the inclinometer is rotated until the measure tilt angle exceeds the +45 deg range upper limit.

8.21 Reading the registers 180 angle [0003 hex], 360 angle [0004 hex], Temperature [0005 hex] and Status [0006 hex]

Request

Request of the Master to read the **180 angle [0003 hex]** and **360 angle [0004 hex]** registers as well as the internal temperature (**Temperature [0005 hex]** register) and the functional state of the inclinometer (**Status [0006 hex]** register)

20 03 00 03 00 04 B2 B8

Response

20 03 08 11 94 14 C6 00 1C 04 03 27 E0

180 angle [0003 hex] > 180 angle = 0x1194 = +45.00 deg CLAMPED VALUE
360 angle [0004 hex] > 360 angle = 0x14C6 = 53.18 deg
Temperature [0005 hex] > Temperature = 0x001C = +28°C
Status [0006 hex] > Status register = 0x0403 = $0100\ 0000\ 0011_2$ = (bit 0 = 1 = Error Presence status is active); (bit 1 = 1 = **1-Axis Mode** status is active); (bit 10 = 1 = **X-Angle Err.USER Rng.High** error is active: the value exceeds the upper limit of the range)

9 Default parameters list

Registers list and address	Default value		
Baud rate [000A hex]	4 = 19,200		
Parity [000B hex]	2 = Even		
Stop bits [000C hex]	1 = 1 stop bit		
Node address [000D hex]	100		
TermRes [000E hex]	1		
Filter [000F hex]	100		
X zero [0014 hex]	0		
X preset [0015 hex]	0		
X offset [0016 hex]	0		
X inversion [0017 hex]	1 = OFF		
X range [0018 hex]	30		
Y zero [001E hex]	0		
Y preset [001F hex]	0		
Y offset [0020 hex]	0		
Y inversion [0021 hex]	1 = OFF		
Y range [0022 hex]	30 (2 axis device) 180 (1 axis device)		

This page intentionally left blank

Document release	Release date	Description	HW	SW	Interface
1.0	28.01.2021	First issue	-	-	-

This device is to be supplied by a Class 2 Circuit or Low-Voltage Limited Energy or Energy Source not exceeding 30 Vdc. Refer to the order code for supply voltage rate.
Ce dispositif doit être alimenté par un circuit de Classe 2 ou à très basse tension ou bien en appliquant une tension maxi de 30Vcc. Voir le code de commande pour la tension d'alimentation.

Dispose separately

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

Tel. +39 0445 806600
Fax +39 0445 806699

info@lika.biz • www.lika.biz