OPERATING INSTRUCTIONS

LMS400

LASER MEASUREMENT SENSORS

SICK
Sensor Intelligence.

Software version described

Software/tool	Function	Status
LMS400-XXXX	Firmware	V 2.51 or higher
Device description LMS400-XXXX (jar file)	Device specific software module for SOPAS ET	V 01.01.20 or higher
SOPAS ET	Configuration software	V 03.00 or higher

ATTENTION

The LMS400 laser measurement sensor is intended exclusively for use in industrial environments. When used in residential areas it can cause interferences.

Getting started ...

- **Delivery scope**
  - Section *3.1 “Delivery” on page 19*
- **WARNING!**
  - Chapter *2 “For your safety” on page 13*
- **Mounting the device**
  - Chapter *4 “Mounting” on page 54*
- **Connecting the device electrically**
  - Chapter *5 “Electrical installation” on page 57*
- **Familiarisation with the device and its functions in general**
  - Chapter *3 “Product description” on page 19*
  - Chapter *3.8 “Master/slave operation” on page 33*
  - Chapter *10 “Technical specifications” on page 75*
- **Setting up of the device for measurement**
  - Chapter *7 “Configuration (parameterisation)” on page 64*
- **Help with resolving problems**
  - Chapter *9 “Troubleshooting” on page 73*
- **Where is what?**
  - *“Table of contents” on page 5*

Installation steps (overview)

1. Check delivery for completeness.
2. Connect LMS400 to voltage supply.
3. Switch on the PC and start Windows (minimum requirements: Windows 98).
4. Connect PC to LMS400 using the Ethernet interface.
5. Install SOPAS ET configuration software on PC.
   To do so, download SOPAS ET from the Internet at *www.sick.com/sopas*.
6. Start the SOPAS ET configuration software user interface and load the LMS400 device driver in the device catalogue.
7. Click SEARCH SETTINGS, select LMS4xx device family, and choose the IP address of the device (default = 192.168.0.1).
8. From the DEVICE menu select the LOGIN DEVICE command and log in to the device as AUTHORIZED CLIENT using the password “client”.

ATTENTION

Do not switch off the voltage supply during configuration!

Switching off the voltage supply during configuration causes all parameters already configured to be lost.

9. Configure the LMS400 with the aid of the parameters on the device pages of SOPAS ET for the desired application. The parameter values are transferred to the RAM in the LMS400 so that you can check their effect immediately.
10. Check the measurement area of the sensor (PROJECT TREE, LMS400-XX00, MONITOR, SCAN VIEW).

Recommendation

Use the graphic scan view in SOPAS ET to verify the measured values generated and to verify the measurement area online. During this process, note that the monitor cannot display the data in real-time and therefore does not display all measured values.

11. Test the configuration under real conditions.
12. If necessary, correct and optimise the parameters set.
13. Save parameter set permanently in the LMS400 (menu LMS400_XX00, PARAMETER, SAVE PERMANENT). In this way it is saved in EEPROM and is available after a power failure.
14. Save parameter record as a configuration file (extension “*.sdv“ respectively “*.sopas“) (menu PROJECT, SAVE PROJECT AS...).
15. The LMS400 is ready for use with the application-specific configuration. Now you are able to send measured values telegrams to the system and poll measured values.

Table of contents

Abbreviations

BCC Block Character Check

CAN Controller Area Network = standardised fieldbus system with message-based protocol for exchanging data

CS Checksum

EEPROM Electrically Erasable Programmable Read-only Memory

FSI Fast Serial Interface = SICK communication interface

HTML Hypertext Markup Language = page description language in the Internet

LED Light Emitting Diode

LMS SICK AG laser measurement sensor

RAM Random Access Memory = volatile memory with direct access

RIS Remission Information System

ROM Read-only Memory (permanent)

SDD SOPAS Device Driver file

SOPAS ET SICK OPEN PORTAL for APPLICATION and SYSTEMS Engineering Tool = configuration software for the configuration of the LMS400

Tables

Figures

1 About this document

Please read this chapter carefully before working with this documentation and the LMS400 laser measurement sensor.

1.1 Function of this document

These operating instructions are designed **to address the technical personnel** in regards to safe mounting, installation, configuration, electrical installation, commissioning, operation and maintenance of the LMS400 laser measurement sensor.

1.2 Target group

The intended target group for this document is people in the following positions:

1.2.1 Mounting, electrical installation, maintenance and replacement

Factory electricians and service engineers

1.2.2 Commissioning, operation and configuration

Technicians and engineers

1.3 Scope

These operating instructions apply to the following variants:

LMS400 variant	Laser power	Material of front screen	Part.No.
LMS400-1000	7,5 mW	Float glass	1027897
LMS400-2000	10 mW	Float glass	1041725
LMS400-1000S02	7.5 mW	Polycarbonate[1)]	1070166

Tab. 1: Variants of the LMS400 laser measurement sensor

1) Used in order to prevent fragments of glass being produced in the event of mechanical damage (in food applications, for instance). Plastic panes have a higher optical attenuation value than glass panes. This may result in a reduction in the measurement accuracy and detection capacity of the device as compared with the standard variant.

1.4 Depth of information

These operating instructions contain the following information on the LMS400 laser measurement sensor:

- mounting
- electrical installation
- commissioning and configuration
- maintenance
- troubleshooting and rectification
- ordering information
- conformity and approval

Planning and using measurement sensors such as the LMS400 also require specific technical skills which are not detailed in this documentation.

The LMS400 is configured on-site for the related application using the SOPAS ET configuration software (see chapter *7 “Configuration (parameterisation)” on page 64*).

When operating the LMS400 laser measurement sensor, the national, local and statutory rules and regulations must be observed.

Notes

- In the following document, LMS400 refers to the LMS400-1000 and the LMS400-2000 except when it is necessary to be more specific.
- Please refer also to the LMS400 information in the Internet at *www.sick.com/lms4xx*.

1.5 Symbology used

Recommendation Recommendations are designed to give you assistance in the decision-making process with respect to a certain function or a technical measure.

Note Refer to notes for device special features.

Explanation Explanations provide background knowledge on technical relationships.

MENU COMMAND This typeface indicates a term in the SOPAS ET user interface.

Terminal output This typeface indicates messages that the LMS400 outputs via its aux interface.

➢ Take action ... Instructions for taking action are shown by an arrow. Read carefully and follow the instructions for action.

This symbol refers to additionally available documentation.

ATTENTION

Warning!

A warning indicates an actual or potential hazard. They are designed to help you to prevent accidents and to protect the device from being damaged.

Read carefully and follow the warning notices!

Software notes show where you can make the appropriate settings and adjustments in the SOPAS ET configuration software.

2 For your safety

This chapter deals with your own safety and the safety of the equipment operators.

➢ Please read this chapter carefully before working with the LMS400.

2.1 Authorized personnel

The LMS400 laser measurement sensor must be installed, commissioned and serviced only by adequately qualified personnel.

The following qualifications are necessary for the various tasks:

2.1.1 Mounting and maintenance

- basic technical training
- knowledge of the current safety regulations in the workplace

2.1.2 Electrical installation and replacement

- practical electrical training
- knowledge of current electrical safety regulations
- knowledge on the use and operation of devices in the related application (e.g. conveyors)

2.1.3 Commissioning, operation and configuration

- knowledge on the use and operation of devices in the related application (e.g. conveyors)
- knowledge on the software and hardware environment in the related application (e.g. conveyors)
- basic knowledge of the Windows operating system
- basic knowledge of an HTML browser (e.g. Internet Explorer)
- basic knowledge of data transmission

2.2 Applications of the device

The LMS400 laser measurement sensor is intended exclusively for use in industrial environments. When used in residential areas it can cause interferences. The LMS400 measures inanimate objects of any shape and determines the following contour data:

- angular position
- distance to the zero point on the respective angular position
- RIS value (Remission Information System, see section *3.6.2 “Remission value” on page 27*)

This information is passed by the sensor over one of its data interfaces to the customer’s computer for further processing.

Compatibility of devices with older firmware versions

The parameter sets for a LMS400 with a firmware version lower than V1.13 saved in a SOPAS ET project cannot be transferred to devices with a firmware version from V1.13.

2.3 Correct use

The LMS400 laser measurement sensor must be used only as defined in section *2.2 "Applications of the device" on page 13*. It must be initialised only by qualified personnel and only in industrial environments.

If the device is used for any other purposes or modified in any way — also during mounting and installation — any warranty claim against SICK AG shall become void.

2.4 General safety notes and protective measures

Safety notes

Please observe the following items in order to ensure the correct and safe use of the LMS400 laser measurement sensor.

- The notices in these operating instructions (e.g. on use, mounting, installation or integration into the existing machine controller) must be observed.
- National/international rules and regulations apply to the installation, commissioning, use and periodic technical inspections of the laser measurement sensor, in particular:
  - work safety regulations/safety rules
  - other relevant health and safety regulations
- Manufacturers and operators of the system are responsible for obtaining and observing all applicable safety regulations and rules.
- The tests must be carried out by specialist personnel or specially qualified and authorized personnel and must be recorded and documented to ensure that the tests can be reconstructed and retraced at any time.
- The operating instructions must be made available to the operator of the system where the LMS400 laser measurement sensor is used. The operator of the system is to be instructed in the use of the device by specialist personnel and must be instructed to read the operating instructions.

Risk of injury from electrical power!

The LMS400 laser measurement sensor is connected to 24 V DC.

➢ Observe the current safety regulations when working on electrical systems.

The LMS400 laser measurement sensor is intended exclusively for use in industrial environments. When used in residential areas it can cause interferences.

2.4.1 Laser radiation

WARNUNG	WARNING	AVERTISSEMENT
Laserstrahlung!	**Laser radiation!**	**Rayonnement laser !**
LASER PRODUKT **KLASSE 2**	**CLASS 2** **LASER PRODUCT**	**APPAREIL À** **LASER DE CLASSE 2**
Nicht in den Strahl blicken! Wellenlänge: 660 nm (sichtbares Rotlicht) IEC 60825-1:2014 Identische Laserklasse für Ausgabe EN/IEC 60825-1:2007 **LMS400-1000:** Max Leistung: 7.4 mW cw Wellenlänge: 660 nm **LMS400-2000:** Max Leistung: 9.5 mW cw Wellenlänge: 660 nm	**Do not stare into beam!** Wavelength: 660 nm (visible red light) IEC 60825-1:2014 Identical laser class for issue EN/IEC 60825-1:2007 **LMS400-1000:** Max. output: 7.4 mW cw Wavelength: 660 nm **LMS400-2000:** Max. output: 9.5 mW cw Wavelength: 660 nm	**Ne pas regarder dans le faisceau !** Longueur d'onde: 660 nm (lumière rouge visibles) IEC 60825-1:2014 Même classe laser pour l'édition EN/CEI 60825-1:2007 **LMS400-1000:** Puissance max.: 7.4 mW cw Longueur d'onde: 660 nm **LMS400-2000:** Puissance max.: 9.5 mW cw Longueur d'onde: 660 nm
21 CFR 1040.10 und 1040.11 wird erfüllt, mit Ausnahme der Abweichungen nach Laser Notice 50 vom Juni 2007.	Complies with 21 CFR 1040.10 and CFR1040.11 except for deviations pursuant to Laser Notice No. 50, June 2007.	Soit 21 CFR 1040.10 et 1040.11 à l'exception de différences sur les indications du Laser N° 50, juin 2007.
Die im normalen Betrieb austretende Strahlung ist ungefährlich für die Augen und die menschliche Haut. Bei längerem Blick in den Strahlengang kann die Netzhaut im Auge beschädigt werden.	The radiation emitted in normal operation is not harmful to the eyes and human skin. On extended beam exposure, the retina in the eye may be damaged	Le rayonnement émis en fonctionnement normal n'est pas dangereux pour les yeux et la peau humaine. Toute exposition prolongée au faisceau peut entraîner des lésions de la rétine.
VORSICHT – Bestimmungsfremder Einsatz kann zu gefährlichen Strahlungsexpositionen führen.	CAUTION – the use of controls, or adjustments or performance of procedures other than those specified herein may result in hazardous radiation exposure.	PRUDENCE – tout usage de commandes, réglages ou toute application de procédures autres que ceux décrits dans ce document peut entraîner une exposition dangereuse au rayonnement.
Laserwarnschild Siehe *Fig. 1 on page 17*.	**Laser warning sign** See *Fig. 1 on page 17*.	**Avertissement laser** Voir *Fig. 1 on page 17*.

Tab. 2: Laser warnings

WARNUNG	WARNING	AVERTISSEMENT
Um die Einhaltung der Laserklasse 2 zu gewährleisten, ist keine Wartung notwendig.	No maintenance is necessary to ensure compliance with laser class 2.	Aucune maintenance n'est nécessaire pour assurer la conformité avec la classe laser 2.
➢ Gehäuse nicht öffnen (durch das Öffnen wird der Laser nicht abgeschaltet). ➢ Beachten Sie die Laserschutzbestimmungen gemäß IEC 60825-1 (neueste Fassung).	➢ Do not open the housing (opening the housing will not switch off the laser). ➢ Pay attention to the laser safety regulations as per IEC 60 825-1 (latest version).	➢ Ne pas ouvrir le boîtier. (La diode laser n'est pas désactivée en cas d'ouverture du boîtier). ➢ Se conformer aux dernières consignes de protection en date contre le rayonnement laser (IEC 60825-1).
Laseraustrittsöffnung Die Laseraustrittsöffnung ist die Frontscheibe am LMS400. Siehe *Fig. 1 on page 17*.	**Laser output aperture** The laser output aperture is the front screen on the LMS400. See *Fig. 1 on page 17*.	**Orifice de sortie** L'orifice de sortie du faisceau laser correspond à la vitre dans son ensemble. Voir *Fig. 1 on page 17*.

Tab. 2: Laser warnings

WARNING

Damage to the eye from laser radiation!

The LMS400 uses a red laser of class 2. On extended beam exposure, the retina in the eye may be damaged.

The entire front screen serves as the laser output aperture.

Warning — inappropriate use of the LMS400 can result in hazardous exposure to radiation and the laser class may be exceeded.

- ➢ Never look directly into the beam (similar to sunlight).
- ➢ Do not point the device laser beam at people.
- ➢ During mounting and adjustment of the LMS400, pay attention to possible reflections of the laser beam on reflective surfaces.
- ➢ Do not open the housing. (Opening the housing does not interrupt the power to the laser diode during the read cycle.)
- ➢ Observe the latest valid version of the laser safety regulations.

Laser output aperture

The laser output aperture is the front screen of the LMS400.

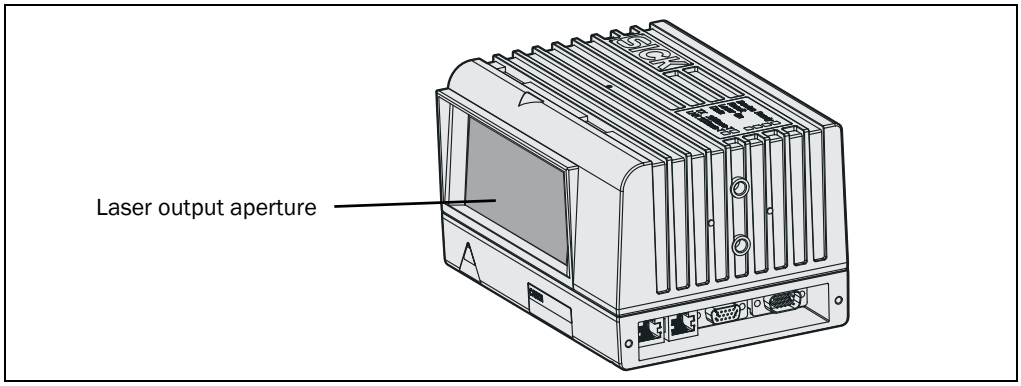


Fig. 1: Laser output aperture

Note When operated with trigger, the laser diode is only switched on when the switch on signal is present and is switched off again with the shut down signal. For free running data output the laser is continuously switched on.

Laser warning label

On the LMS400 laser measurement sensor are several laser warning labels and laser warning symbols (see *Fig. 2*). The warning text varies depending on the laser power.

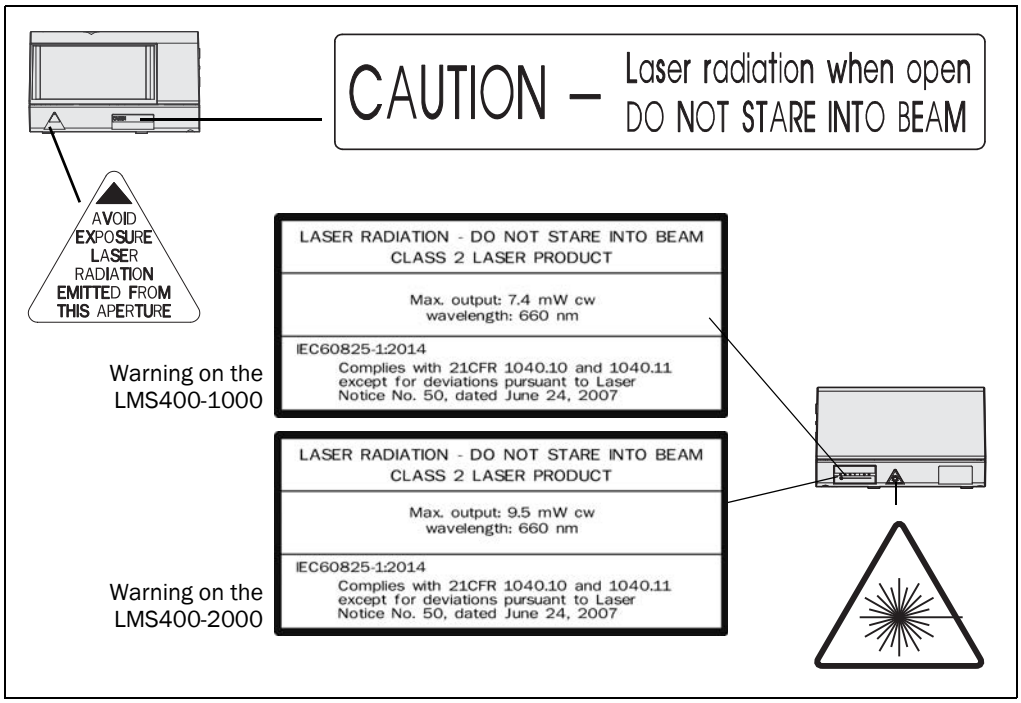


Fig. 2: Laser warning labels on the LMS400

Notes

- Prior to commissioning, the warning label in English on the device “LASER RADIATION - DO NOT STARE INTO BEAM” is to be replaced with a laser warning label in a language understood by the operators of the system. Laser warning labels in German/English and French/English are in the delivery. Leave the “CAUTION …” and “AVOID EXPOSURE LASER …” laser warning labels in English.

- If the LMS400 laser measurement sensor is installed in a system/a trim panel such that the laser warning labels are covered, further warning labels (not in the delivery) are to be affixed beside the outlet aperture for the laser beam on the system/on the trim panel!
- The LMS400 automatically monitors the generation of the beam and automatically shuts down the laser diode in case of irregularities. In this case the red LED lights up and the scanner transmits no more measured values.

2.5 Quick stop and Quick restart

Note Quick stop and Quick restart can also be performed using a configuration telegram (see *11.5.7 on page 102*).

2.5.1 Switch the LMS400 off

➢ Switch off the voltage supply for the LMS400 or disconnect the power supply cable.

The LMS400 retains parameters stored in the internal, non-volatile memory. Measured values on the interface are lost.

2.5.2 Switch on the LMS400

➢ Switch on the voltage supply for the LMS400 or reconnect the supply cable.

The LMS400 restarts operation with the last saved parameters.

2.6 Environmental protection

The LMS400 laser measurement sensor has been designed to minimise environmental impact. It uses only a minimum of power.

While working, always act in an environmentally responsible manner. For this reason please note the following information on disposal.

2.6.1 Power consumption

The LMS400 consumes max. 25 W of power.

2.6.2 Disposal after final de-commissioning

➢ Always dispose of unserviceable devices in compliance with local/national rules and regulations on waste disposal.

➢ Dispose of all electronic assemblies as hazardous waste. The electronic assemblies are straightforward to dismantle.

See section *8.2 “Disposal” on page 72*.

3 Product description

This chapter provides information on the special features and properties of the LMS400 laser measurement sensor. It describes the construction and the operating principle of the device, in particular the different operating modes.

Note Please read this chapter before mounting, installing and commissioning the device.

3.1 Delivery

The LMS400 laser measurement sensor as delivered comprises:

- 1 LMS400 laser measurement sensor. Model type depends on order.
- 1 Notes on device with electrical circuit diagram for getting started

An overview about the available LMS400 variants, its accessories and, device documentation is available from the web product page

➢ *www.sick.com/lms4xx*

3.2 Special features of the LMS400

- electro-sensitive, active measurement technique
- measurement of objects with almost any shape
- works with a very wide range of surface structures
- flexible system configurations
- various filters for pre-processing the measured values

Special features of the Level Control application

- division of the measurement area into vertical sub-sections (columns) for the qualified evaluation for example of levels in containers, complete filling or emptying
- simple equidistant or detailed customer-specific configuration of the columns
- evaluation of the Z values (level) within a sub-section for simple applications in the Y axis, for complex applications also in the X axis
- area measurement, even within a container

Special features of the LMS400-1000

- laser power 7.5 mW
- object remission 6.5% ... 200%

Special features of the LMS400-2000

- laser power 10 mW
- object remission 4.5% ... 100%

3.3 Planning

3.3.1 System requirements of the LMS400 laser measurement sensor

The maximum working range of the LMS400 is 3 m (9.84 in). The smallest permitted distance of the measurement object from the zero point of the LMS400 is 700 mm (27.56 in). The zero point is marked both on the top of the housing and on the underside of

the housing (see *Fig. 42 on page 77*).

The working area of the LMS400 covers an angle of 70° (see *Fig. 3*).

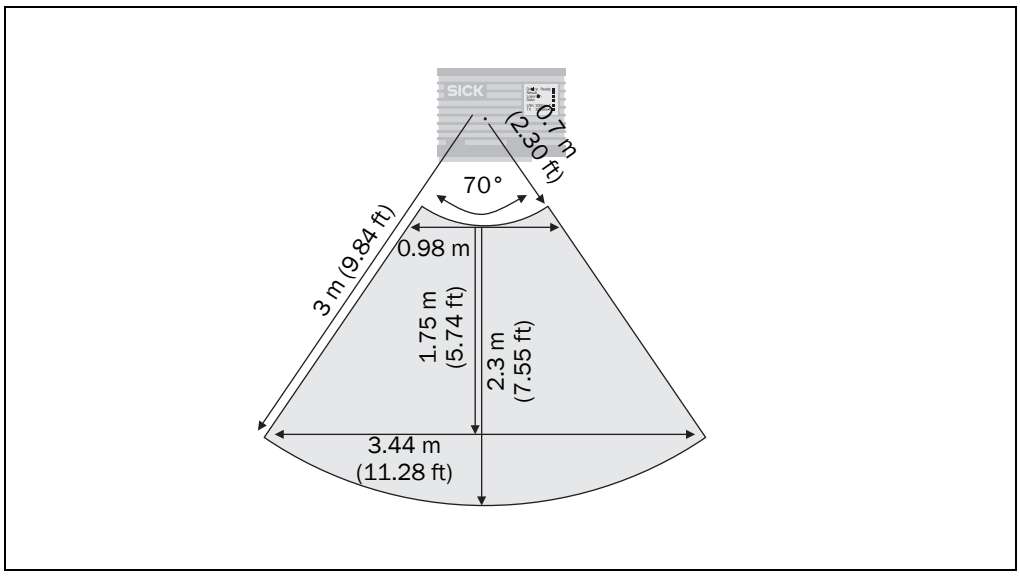


Fig. 3: Operating principle of the LMS400 laser measurement sensor

For the commissioning and operation of the LMS400 laser measurement sensor the following are required:

- typical space required for the LMS400 installation above: about 700 mm (27.56 in) above the highest object

Note The LMS400 must have a clear view of the object to be measured.

- operating voltages:
  LMS400: 24 V DC ± 15% in compliance with IEC 364-4-41 (protective extra-low voltage), output max. 25 W
- data interface RS-232, RS-422, Ethernet

Note To quickly connect the LMS400 to a host or a PLC in a manner suitable for industrial use, the LMS400 can be connected using a connection module (CDM490) and/or a plug cover can be connected (see *5.4 "Connection via connection module or plug cover" on page 61*).

3.3.2 Object specifications

The LMS400 can only safely detect parts of objects, e.g. edges, surrounds or protruding parts, when the area visible for the LMS400 is at least three times the angular resolution resulting from the distance to the zero point. If the area is smaller, distance measurements outside the tolerance of the LMS400 may be produced.

3.3.3 Mounting requirements

The LMS400 must be mounted as follows:

- robust (weight LMS400: approx. 2.3 kg resp. 5.1 lb)
- without vibration
- without oscillations

Note The mounting kit is easy to mount on an 80-mm-item aluminium profile. The mounting kit is matched to these profiles (see section *10.2.2 "Dimensional drawing mounting kit No. 2030421 for LMS400" on page 78*). For mounting on other brackets, see section *10.2.1 "Dimensional drawing LMS400 laser measurement sensor" on page 77*.

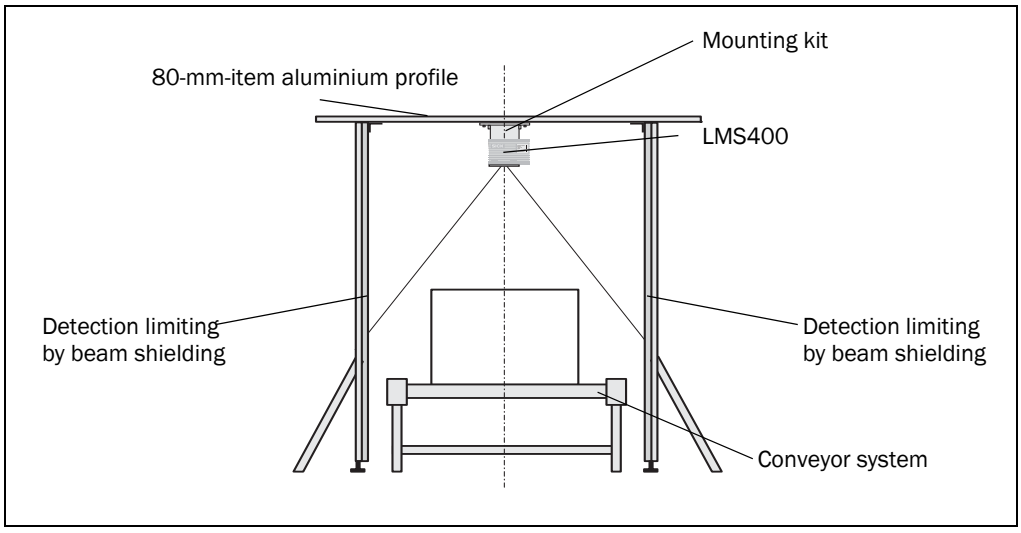


Fig. 4: Example for mounting above a conveyor system

To obtain an optimal measurement result, the following points should be noted:

- Typical space required for the LMS400 installation above: about 700 mm (27.56 in) above the highest object
- The LMS400 must have a clear view of the conveyor system.
- The LMS400 laser beams should not reach beyond the application area so that persons or items transported on neighbouring conveyor systems are not detected (detection limiting by beam shielding).
- The maximum detection must be limited to a working range of three metres as otherwise measuring inaccuracies can occur.
- Adequate distance of the LMS400 from bends, induction lines, start-stop areas, inclined areas and separators on the conveyor system

3.3.4 Requirements on the conveyor system (on usage of the Level Control application)

- The conveyor system must have a constant conveyor velocity or an incremental encoder must be installed.
- The objects can be moved on a conveyor system with a flat transporting surface. Rotation, vibration, swaying and slipping of the objects on the conveyor system as well as uneven transporting surfaces can reduce the measurement accuracy and degrade the evaluation.

3.4 Operating principle of the LMS400

The LMS400 laser measurement sensor opto-electronically scans a two-dimensional measurement area.

The LMS400 does not require any reflectors or position markers. This is an active system with a red laser. It is not necessary to illuminate the objects.

Phase shift principle

The LMS400 uses the principle of phase shift (continuous wave). The propagation time of the light and the wavelength used result in a phase shift between the beam sent and the beam received. This phase difference is converted to a frequency. The LMS400 determines the distance of the object from the zero point based on this frequency.

Measured value output

The LMS400 supplies the measured values to its interfaces if this action is requested using a telegram. Distance and remission values, only distance values or only remission values can be queried from the LMS400.

Notes

- The LMS400 outputs the data after the start of the measurement using the same interface over which the measured values were requested.
- It is only possible to output all measured values of a scan in real-time using the Ethernet interface.

In case of an error, the measured value output is stopped immediately and an error code output that can be evaluated by the application connected. The error code can also be queried via SOPAS ET from the LMS400 (see *9.3 "Detailed error analysis" on page 73*).

In principle a differentiation is to be made between continuous and triggered measured value output.

Level Control

With the aid of the Level Control application, for instance levels in containers, the complete filling of pallets or the complete emptying of transport containers can be checked.

For this purpose mount the LMS400 above a conveyor system. From there it electro-sensitively scans objects moving past.

The application splits the scanning area of the LMS400 into vertical columns (mesurement from above). The columns can be used to detect the object height of material in transported cases. The evaluation results (material protruding yes/no) are indicated by up to 4 assignable switching outputs. If required, the evaluation results can be requested via telegram over the data interface.

3.5 Measured value output

3.5.1 Free running measured value output

In case of free running measured value output, measured value telegrams are output after the LMS400 has received the measured value request until the output of measured values is stopped by a stop telegram. Measurements are performed continuously between the reception of the measured value request and the reception of the stop telegram; the laser diode is switched on.

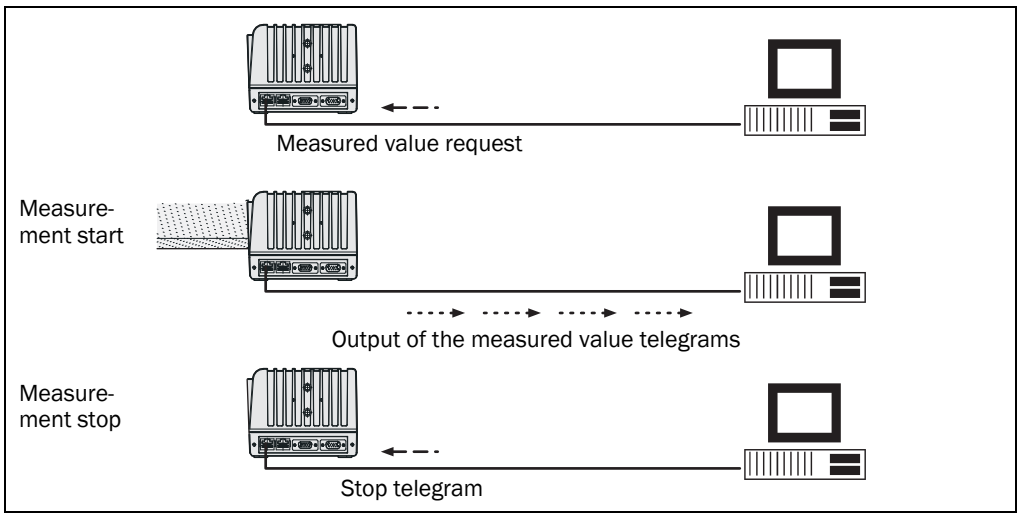


Fig. 5: Continuous measured value output

ATTENTION

Limit the duration of the measurement with free running measured value output!

The service life of the laser diode and as a result of the system will be reduced by continuous measurement. Only start the measurement if objects to be measured are present. Stop the measurement if there are no objects to be measured present. Control the measurement either using your application or use the triggered measurement or the laser control (see *3.5.2 on page 23* and *3.5.3 on page 24*).

3.5.2 Control of the measurement process using a gate

PROJECT TREE, LMS400-XX00, PARAMETER, DIGITAL INPUTS/TRIGGER, area GATE

With the triggered measurement the start and end of the measurement is defined by a so-called gate. For the measured value output, measured values are initially requested using the sMN mLRreqtrigdata telegram (see *11.3.2 on page 86*). Measured values are then only output if the gate is opened. The measurement and the output of measured values are stopped when the gate is closed.

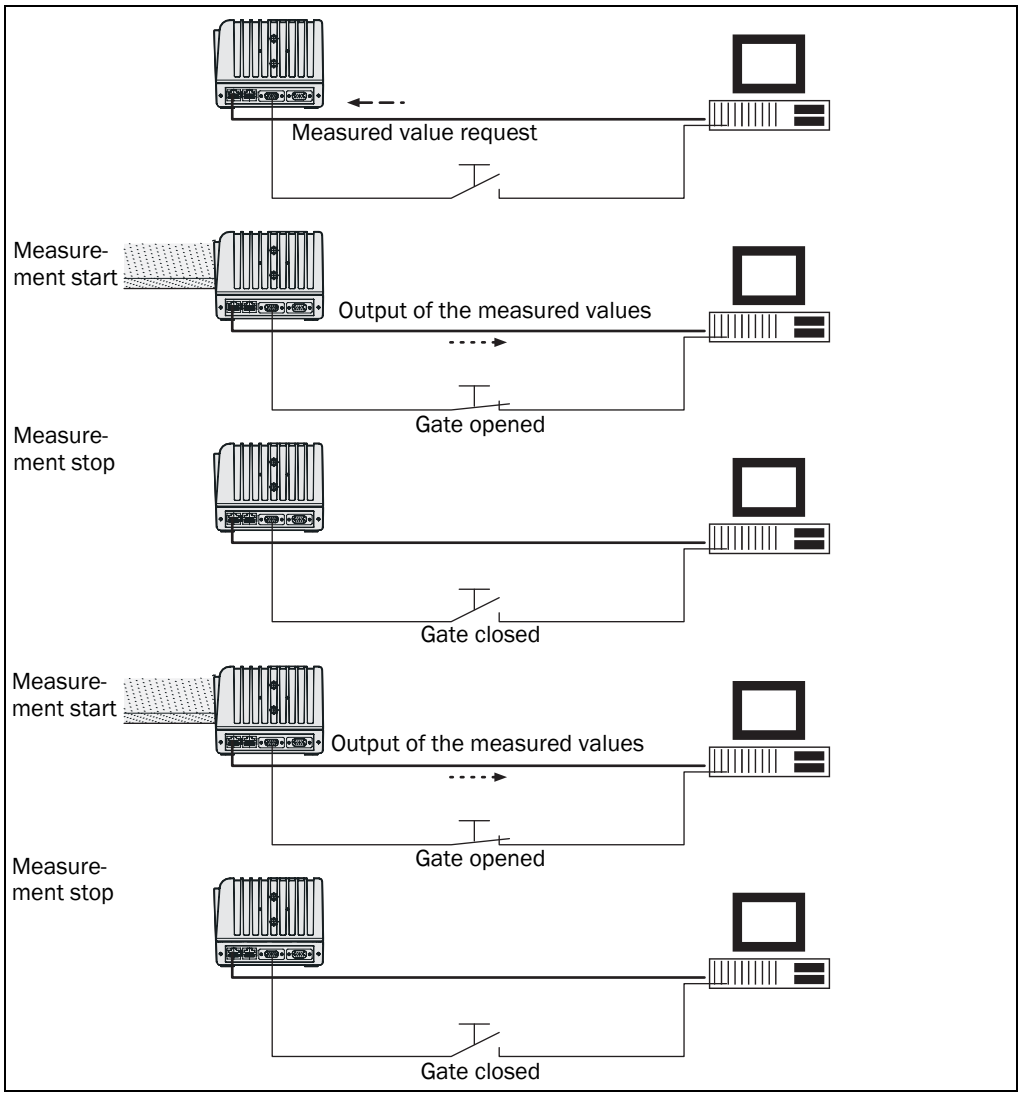


Fig. 6: Triggered measured value output with gate

The LMS400 provides several trigger sources for gate control:

- digital inputs (The electrical connection is described in section *5.3.3 ""I/O" connection" on page 59*.)
- software trigger (see *11.3.3 on page 87*)
- CAN-BUS
- test trigger
- triggering by the LMS400 master

Note A trigger has no effect when the device has shut down the laser diode as a result of an error during beam generation.

3.5.3 Laser control

PROJECT TREE, LMS400-XX00, PARAMETER, DIGITAL INPUTS/TRIGGER, area LASER CONTROL

Note If laser control is active, the laser diode is only switched on when objects are measured. This increases the service life of the system. Without laser control, the laser remains on all the time. This has a negative effect upon the service life of the system.

With laser control active, the switching on of the laser is controlled by the gate configured or is controlled independently by a dedicated source.

- controlled by the gate
  The laser is controlled by the start and stop trigger source configured in the GATE SETTINGS area (see *"Control of the measurement process using a gate" on page 23*).
- independent
  The laser is controlled by the source configured in SOPAS ET. The control is independent of the settings made in the Gate settings area.

For independent laser control a photoelectric switch, for instance, is connected to the LMS400 as a trigger. When an object passes the photoelectric switch, the laser is switched on.

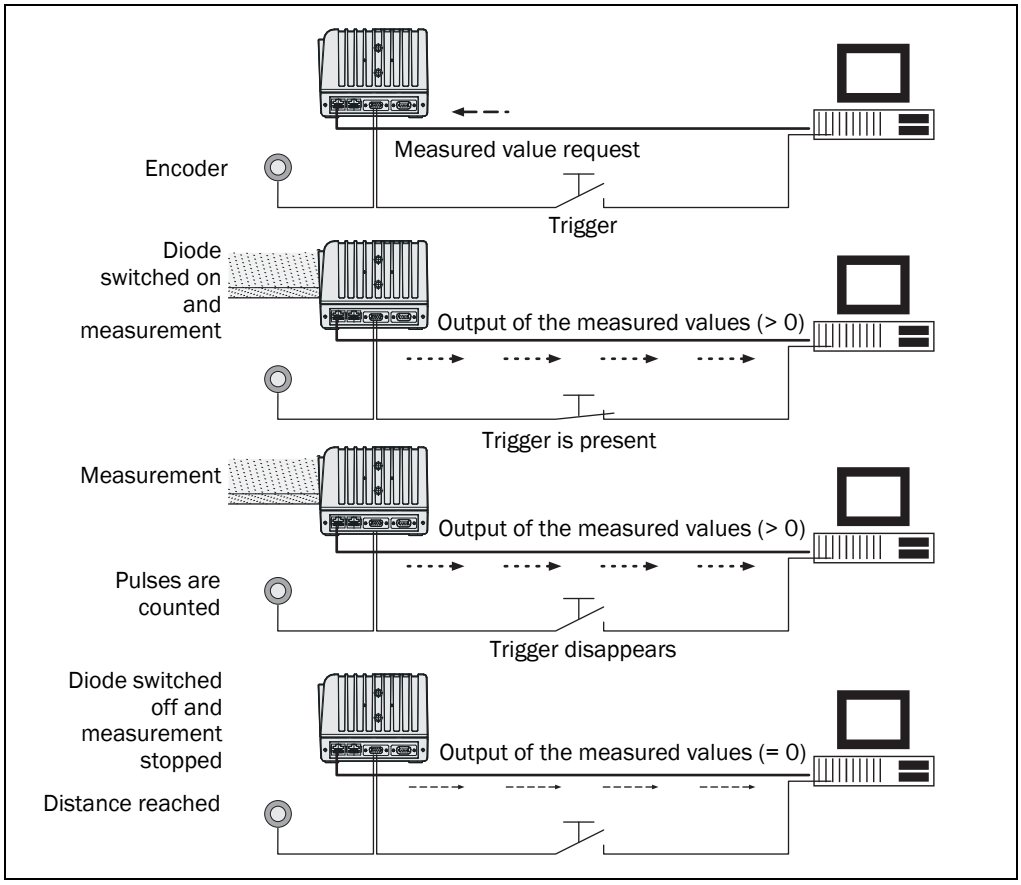


Fig. 7: Measured value output with laser control

With independent laser control the laser is switched on by a trigger. However, if this trigger disappears, the measured value output is not stopped.

Only when the LMS400 has calculated using encoder pulses that the laser switch off distance has been reached does it switch off the laser. Only invalid measured values (= 0) are supplied.

In addition, a laser switch-off delay can be configured. Should the conveyor system of the LMS400 remain stationary after the trigger pulse for operational reasons or due to a malfunction, the laser is switched off after the laser switch-off delay configured has elapsed.

If, between switching on and off the laser, the start trigger is set again (for instance by a new object passing a photoelectric switch), the distance calculation and the laser switch-off delay counter are set to zero and re-started. The laser therefore **remains switched on**.

3.5.4 Connection of encoders

PROJECT TREE LMS400-XX00, PARAMETER, INCREMENT CONFIG./SYNC., area INCREMENT SETTINGS

If the LMS400 is mounted floating or the objects to be measured move, in general the application also needs the position data to be able to further process the measured values. You can connect the data signal from various encoder types to the LMS400. The encoder data is then available to the application on the same interface together with the usual measured values during a scan (see *"Continuous measured value output" on page 83*). The application can calculate the position data from these data.

You can use the following encoders:

- single-channel, is only connected to IN2,
  no direction detection
- dual-channel, is connected to IN2 and IN4
  The pulses have a phase offset of 90°, as a result direction detection is possible.
- dual-channel, is connected to IN2 and IN4
  The pulses are present on IN2, the direction is indicated on IN4 using the level 0 or 1.

The electrical connection of encoders is described in section *5.3.3 ""I/O" connection" on page 59*.

3.6 Measuring accuracy of the LMS400

Several characteristics are available to evaluate the quality of a measured value or a scan:

- the typical systematic measuring error (device specific, see *3.6.1*)
- the remission (object and application dependent, see *3.6.2*)
- the measured value quality (configuration dependent, see *3.6.4*)

3.6.1 Measuring accuracy of the distance measurement

The typical systematic measuring error of the LMS400 is ±4 mm (±0.16 in). This information applies for the individual measurement point at an object remission of 10% to 100% at room temperature.

The statistical measuring error is dependent on the remission and distance of the object. *Tab. 3* shows typical and maximum measuring errors for measured value quality 7, room temperature and maximum external light of 2000 Lux.

Remission	Distance	Statistical error (1 sigma)	
		Typical	Maximum
100/200%	700 to 3000 mm (27.5 to 118 in)	3 mm (0.12 in)	
78%	700 to 3000 mm (27.5 to 118 in)	3 mm (0.12 in)	
	1000 to 2500 mm (39 to 98 in)		5 mm (0.20 in)
	<1000 or >2500 mm (<39 or >98 in)		7 mm (0.28 in)
40%	700 to 3000 mm (27.5 to 118 in)	4 mm (0.16 in)	
	1000 to 2500 mm (27.5 to 118 in)		8 mm (0.31 in)
	<1000 or >2500 mm (39 to 98 in)		9 mm (0.35 in)
10%	700 to 3000 mm (27.5 to 118 in)	9 mm (0.35 in)	
	1000 to 2500 mm (39 to 98 in)		12 mm (0.47 in)
	<1000 or >2500 mm (<39 or >98 in)		15 mm (0.59 in)
6.5%	700 to 3000 mm (27.5 to 118 in)	10 mm (0.39 in)	

Tab. 3: Statistical measuring error

Notes

- The systematic measuring error can be reduced by using external reference targets.
- As in practice, there are several measured values for an object, the statistical measuring error can be reduced by the application. For example, by offsetting the measured values with each other.
- Glossy surfaces and other effects can reduce the accuracy.

3.6.2 Remission value

The remission is the capability of a material to reflect the light back. The remission value expresses the signal strength with different object surfaces.

- **Glossy objects** have different remission values at the same distance with varying angle of incidence. The maximum remission of glossy objects is achieved on perpendicular incidence of the beam.
- **Matt and dull surfaces** have a diffuse remission and therefore have relatively similar remission values independent of the distance from the zero point with a constant angle of incidence.

Tab. 4 shows a few typical remission values.

Material	Typical relative remission value
Photo cardboard (black, matt)	10%
Cardboard (grey)	20%
Wood (rough pine, dirty)	40%
PVC (grey)	50%
Paper (white, matt)	80%
Aluminium (black anodised)	110 ... 150%
Steel (stainless, glossy)	120 ... 150%
Steel (very glossy)	140 ... 200%

Tab. 4: Typical remission values for frequently used materials (source: Kodak standard)

3.6.3 Measurement area expansion

In general, distance measurements are specified with the accuracies given in *Tab. 3* only for remissions of ≥6.5% to 200% (LMS400-1000) or ≥4.5% to 100% (LMS400-2000). The LMS400 therefore only outputs measured values and remission values for objects with the specified remission values.

To be able to also measure distances to objects with lower or higher remission, the LMS400 has the EXTENDED RIS DETECTIVITY function (RIS = Remission Information System). This facilitates the improved detection of dark or light bodies with **reduced accuracy**.

PROJECT TREE, LMS400-XX00, PARAMETER, BASIC PARAMETER, ADVANCED PARAMETERS
With the function activated, the LMS400 also returns the measured values (see *3.6.4 on page 28*) if the remission value is <4.5% or >100% or <6.5% or >200%. For these measured values the following applies:

- The distance value may not have the measuring accuracy defined in section *3.6.1 on page 26*.
- Measurement points/objects with remission values <4,5% or <6,5% will not necessarily be detected by another LMS400 because the remission value determined is dependent on the factory calibration of the system, outside the specified range the calibration may be subject to minor variations.

With extended RIS detectivity, your application must therefore evaluate, based on additional criteria, whether the distance value is to be used or not.

3.6.4 Measured value quality

The “Measured Value Quality” information expresses how much computation time is available to the sensor for the calculation of the measured distance value. The measured value quality should ideally be ≥7. A higher value corresponds to better measured value quality.

Note The information in the data sheet (see section *10.1 “Data sheet LMS400 laser measurement sensor” on page 75*) relate to the measured value quality 7. If the measured value quality determined from the parameters is <7, the sensor is no longer compliant with the values given in the technical data. If the measured value quality is ≥7, the technical data apply.

PROJECT TREE, LMS400-XX00, PARAMETER, BASIC PARAMETER, CURRENT DEVICE PARAMETERS

With a coarse angular resolution and a low scanning frequency, the sensor achieves a measured value quality that tends to be higher than with a fine angular resolution and a high scanning frequency. *Tab. 5* shows the possible measured value quality.

LMS400					
Angular resolution	**f_{scan}**	**Measured value quality**	**Angular resolution**	**f_{scan}**	**Measured value quality**
0.1333°	360	6	0.2857°	420	7
0.1428°	380	6	0.3077°	450	7
0.1538°	410	6	0.3333°	490	7
0.1667°	450	6	0.3636°	500	7
0.1818°	490	6	0.5000°	380	8
0.2500°	370	7	1.0000°	390	9
0.2667°	390	7			

Tab. 5: Possible measured value quality

3.7 Filtering measured values

The LMS400 has digital filters for the pre-processing and optimisation of the measured distance values.

PROJECT TREE, LMS400-XX00, PARAMETER, FILTER

Notes

- You can combine the filters as required. If several filters are active, then the filters act one after the other on the result of the previous filter. The processing in this case follows the following sequence:

  edge filter, median filter, range filter, mean filter.
- Active filter functions change the measured values that are output. It is not possible to convert filtered output values back to the original measured values.

3.7.1 Edge filter

The edge filter prevents incorrect/extreme distance values at edges; these values are produced because it was not possible to determine the distance value for the previous or next point (e.g. if the previous/next measurement point was too dark or outside the measurement area of 3 metres).

With the edge filter enabled, the LMS400 also sets a distance value to 0 at each edge. *Tab. 6* shows an example of unfiltered and filtered measured values.

	Angle (distance_1 to _n)									
	1	2	3	4	5	6	7	8	9	...
Unfiltered Scan 1	0	750	1100	1150	1030	1050	1100	1800	0	0
Filtered scan 1	0	0	1100	1150	1030	1050	1100	0	0	0

Tab. 6: Measured values with edge filter

Note Using the edge filter, points can be completely suppressed at the outer edges of the object. In this case the width determined for an object is too narrow by up to 2 × the angular resolution.

3.7.2 Median filter

The median filter reduces individual extreme values over the entire measurement line by outputting the median for each measurement point (not: the mean/average) from a 3 × 3 matrix. The matrix comprises nine measured values: The distance values for the point and its neighbouring points, as well as the distance values determined for these points in the previous and subsequent scan.

	Angle (distance_1 to _n)									
	1	2	3	4	5	6	7	8	9	...
Scan 1	0	0	850	1100	1150	1030	1050	1100	0	0
Scan 2	0	0	950	1200	1250	1130	1150	1200	0	0
Scan 3	0	0	850	1150	1200	1080	1100	1150	0	0
...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...

Tab. 7: Example median filter: Unfiltered measured values

These nine measured values are sorted in ascending order, the fifth highest measured value is output as the measured value.

	Angle (distance_2 to _n)									
	1	2	3	4	5	6	7	8	9	...
1 = lowest value	Not measurable	0	0	850	1030	1030	1030	0	0	0
2		0	0	850	1080	1050	1050	0	0	0
3		0	0	950	1100	1080	1080	0	0	0
4		0	850	1100	1130	1100	1100	1050	0	0
5 = median		0	850	1150	1150	1130	1100	1100	0	0
6		0	950	1150	1150	1150	1130	1100	0	0
7		850	1100	1200	1200	1150	1150	1150	1100	...
8		850	1150	1200	1200	1200	1150	1150	1200	...
9 = highest value		950	1200	1250	1250	1250	1200	1200	1150	...

Tab. 8: Example median filter: Determining the median for scan 2

Tab. 9 shows the unfiltered and filtered measured values for scan 2 from the previous example.

	Angle (distance_1 to _n)									
	1	2	3	4	5	6	7	8	9	...
Unfiltered scan 2	0	0	950	1200	1250	1130	1150	1200	0	0
Median of scan 2	X	0	850	1150	1150	1130	1100	1100	0	X

Tab. 9: Measured values with median filter

The examples shows the following properties of the median filter:

- The measured values are smoothed, individual outliers are not taken into account.
- The edges of objects are, however, retained.

Notes

- It is not possible to determine a median for the first and last angular step in a scan. The distance value 0 is always output.
- For the first scan after confirmation of the measured value telegram (scan counter = 1) it is not possible to output any measured values.
- Following completion of the third scan, the median for the second scan is calculated and output. There is therefore always an offset in time of one scan. However, the correct value for the scan (= 2) is always output in the scan counter such that e.g. the I/O status can be assigned to the scan. *Tab. 10* shows the delay for different scanning frequencies (for the LMS400-1000 only values ≥360 Hz are relevant).

Scanning frequency	**Median filter enabled**
250 Hz	4.0 ms
300 Hz	3.3 ms
360 Hz	2.8 ms
400 Hz	2.5 ms
450 Hz	2.2 ms
500 Hz	2.0 ms

Tab. 10: Measured data output delay due to the median filter

- If median and mean filters are used together, it is not necessary to take into account any additional time offset for the median filter. This is due to the formation of the mean taking longer than the determination of the median and the median can be formed while the mean is determined.

  Examples:

  – median filter at 400 Hz scanning frequency = 2.5 ms delay

  – mean filter (2 means) + median filter at 250 Hz = 8 ms delay

3.7.3 Range filter

The range filter reduces the number of valid measured values by only outputting distance values that are within a specific distance range. For other measured values the filter delivers the distance value 0 and the remission value 0 as the result.

	Angle (distance_1 to _n)									
	1	2	3	4	5	6	7	8	9	...
Unfiltered scan 1	890	950	1500	1450	1330	1450	1600	1800	2050	2150
Filtered scan 1	0	0	1500	1450	1330	1450	1600	1800	0	0

Tab. 11: Measured values with an range filter from 1000 to 2000 mm (39 to 79 in)

Tab. 11 shows the following properties of the range filter:

- Measured values outside the configured range are not output.
- Measured values inside the configured range are not changed.

3.7.4 Mean filter

The mean filter smooths the distance value. For this purpose the filter forms the arithmetic mean over several scans. The number of scans can be configured.

	Angle (distance_1 to _n)									
	1	2	3	4	5	6	7	8	9	...
Scan 1	0	0	1100	1100	1150	1150	1380	1380	0	0
Scan 2	0	0	1200	1200	1190	950	1500	1500	0	0
Scan 3	0	0	1150	1450	1200	1200	1450	1450	0	0
Scan 4	0	0	1280	1280	1180	1180	1430	1430	0	0
Scan 5	0	0	1170	1170	1220	1220	1470	1150	0	0
1. Output value (scan 5)	0	0	1180	1240	1188	1140	1446	1382	0	0
Scan 1	0	0	1100	1100	1150	1150	1380	1380	0	0
Scan 2	0	0	1200	1200	1190	950	1500	1500	0	0
Scan 3	0	730	1150	1450	1200	1200	1450	1450	0	0
Scan 4	0	0	1280	1280	1180	1180	1430	1430	0	0
Scan 5	0	0	1170	1170	1220	1220	1470	1150	0	0
2. Output value (scan 10)	0	146	1180	1240	1188	1140	1446	1382	0	0

Tab. 12: Measurement values with mathematical filter for five scans

Tab. 12 shows the following properties of the means filter:

- Individual temporal outliers (grey rows in table) affect the mean.
- After confirmation of the measured value telegram, the first measured value is only output after the configured number of scans. There is therefore always an offset in time by the number of scans configured for forming the mean. However, the correct value for the scan is always output in the scan counter such that e.g. the I/O status can be assigned to the scan. *Tab. 13* shows the delay for different scanning frequencies (for the LMS400-1000 only values ≥360 Hz are relevant).

Scanning frequency	**Per mean in the mean filter**	**Means = 2**	**Means = 12**	**Means = 200**
250 Hz	4.0 ms	8 ms	48 ms	800 ms
300 Hz	3.3 ms	6.6 ms	39.6 ms	660 ms
360 Hz	2.8 ms	5.5 ms	33.6 ms	560 ms
400 Hz	2.5 ms	5 ms	30 ms	500 ms
450 Hz	2.2 ms	4.4 ms	26.4 ms	440 ms
500 Hz	2.0 ms	4.0 ms	24 ms	400 ms

Tab. 13: Measured data output delay due to mean filter

- If median and mean filters are used together, it is **not** necessary to take into account any additional time offset for the median filter. This is due to the formation of the mean taking longer than the determination of the median and the median can be formed while the mean is determined.

  Examples:

  – mean filter (12 means) at 360 Hz = 33.6 ms delay
  – mean filter (2 means) + median filter at 250 Hz = 8 ms delay

Recommendation Use the mean filter together with the median filter. In this way individual outliers will be smoothed in advance by the median filter.

3.8 Master/slave operation

In some applications, it is sensible or necessary to use two LMS400. Examples of such applications are:

- Doubling of the effective scanning frequency at the object by operating two LMS400 with the same field of view and with the same angular resolution and scanning frequency. The measured values supplied by the sensors are processed alternately scan for scan.
- Doubling of the field of view by operating two LMS400 with adjacent fields of view. The scans supplied by the sensors are processed in pairs.

So that the two devices can work together, they must be connected using the “System” connection. The electrical connection is described in section *5.3.2 ““System” connection” on page 58*. In addition, one device must be configured as the master, the other as the slave.

PROJECT TREE, LMS400-XX00, PARAMETER, INCREMENT CONFIG./SYNC., area SYNCHRONISATION MASTER/SLAVE

Notes

- On master/slave operation, master and slave must be configured to the same scanning frequency and angular resolution.
- Once the mirrors have run up, master and slave must synchronise with each other. This can take up to 120 seconds. Note that until synchronisation is complete, the measured values supplied by the two devices are not synchronised.
- The slave can use the master’s digital inputs for triggering. For this purpose the software trigger must be configured in the slave as the trigger source.

3.8.1 Phase offset of the rotating mirrors

On the use of two LMS400 in a master/slave system, it may occur that one LMS400 receives the beams from the other LMS400, whether directly (glare) or indirectly (reflection by an object). This can result in incorrect measured results.

To significantly reduce the probability of this problem, the two LMS400 can synchronise with each other so that their rotating mirrors rotate offset from each other by a specific angle. This angle is termed the “Phase”.

PROJECT TREE, LMS400-XX00, PARAMETER, INCREMENT CONFIG./SYNC., area SYNCHRONISATION MASTER/SLAVE

Determining the right phase

Typically the phase for the slave is 0°, and for the master 35°. Reason: The LMS400 has a 70° optical field of view. The probability of mutual interference between two LMS400 is the lowest if the mirrors rotate offset by half the field of view, that is by 35°.

You can configure the phase for each LMS400 individually. The phase of the slave is leading, the phase of the master is lagging.

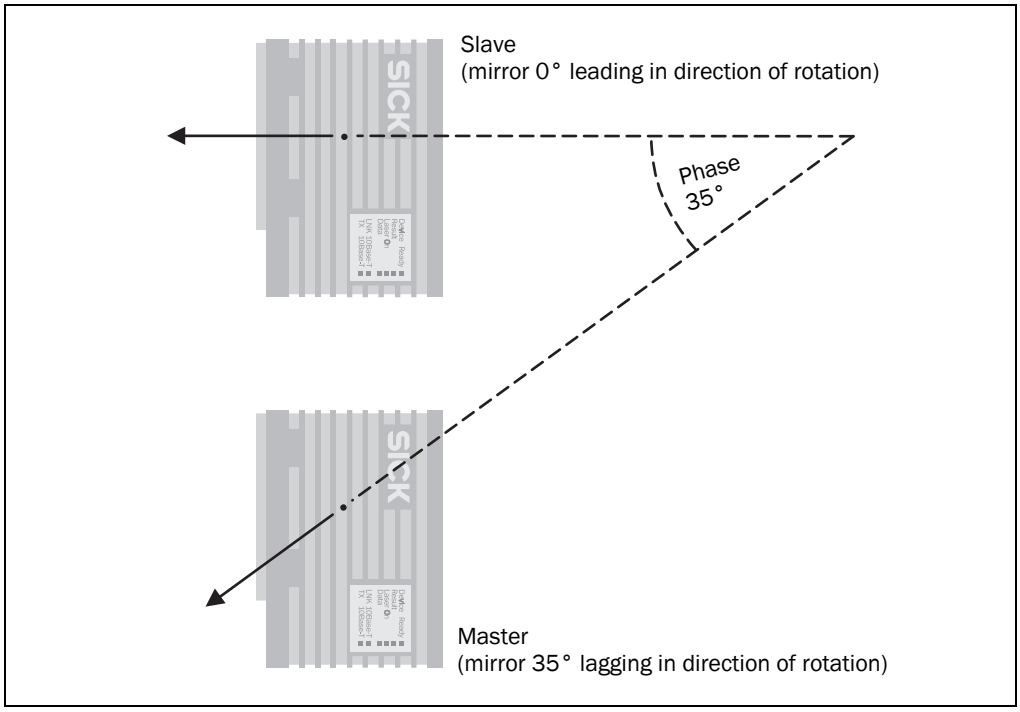


Fig. 8: Example for the phase configuration of a master/slave system

3.9 Level Control

With the aid of the Level Control application, for instance levels in containers, the complete filling of pallets or the complete emptying of transport containers can be checked.

For this purpose mount the LMS400 at an angle of 90° over a conveyor system. From there it electro-sensitively scans objects moving past (*Fig. 9*).

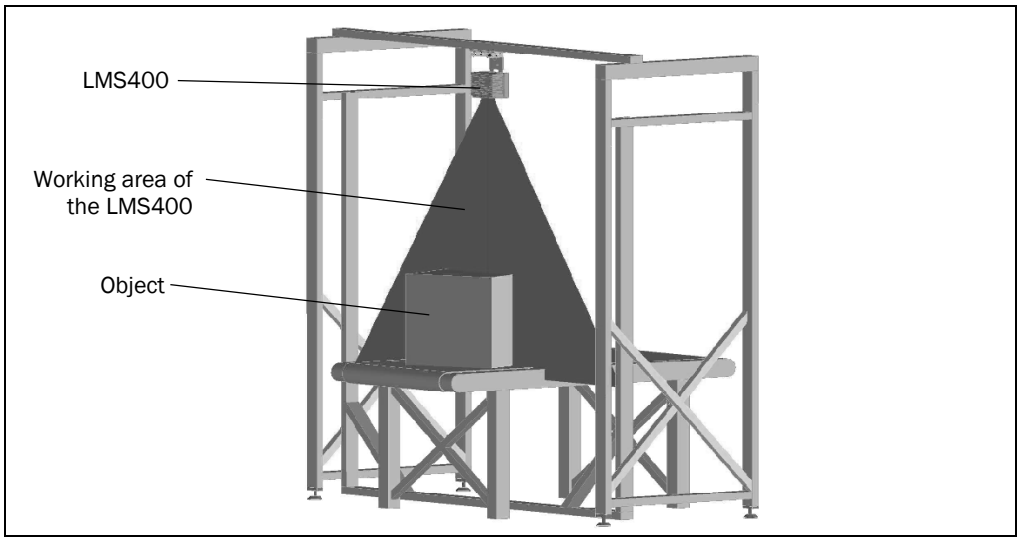


Fig. 9: LMS400 above a conveyor system

With the aid of the application you can divide a defined measurement area in the working area of the LMS400 into several vertical sub-sections. You can allocate switching points including hysteresis to these so-called columns (*Fig. 10*).

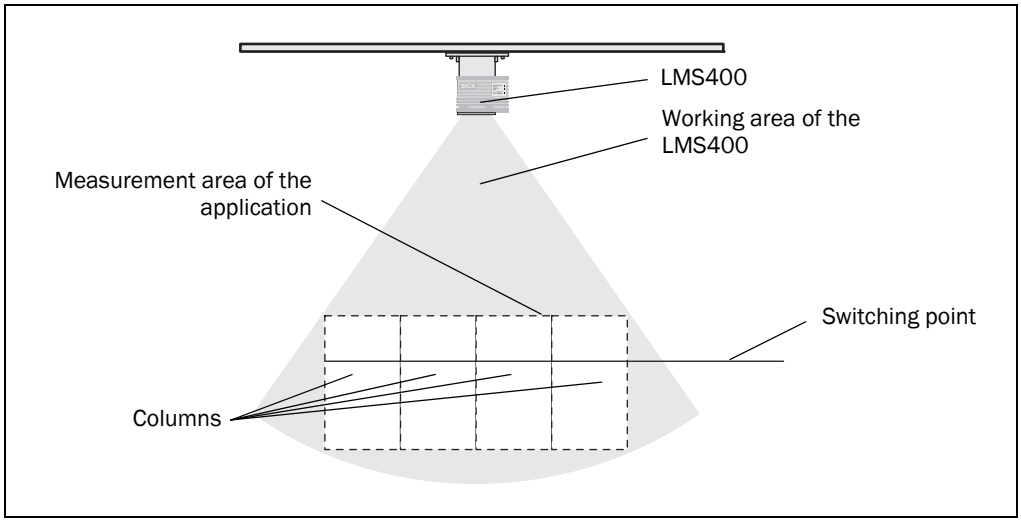


Fig. 10: Working area of the application

With the aid of an external trigger or with an internal gate you can define a length for the columns in the transporting direction.

The LMS400 can now evaluate the Z values for the columns in the Y and X directions and output the result on four digital outputs and one analogue output. In addition, using telegrams you can query the measured values within the columns and whether the values are above or below the switching points. This information can then be processed in another application.

In the example below crates are checked for complete filling. The crates are transported through the scan line of the LMS400 for this check (*Fig. 11*).

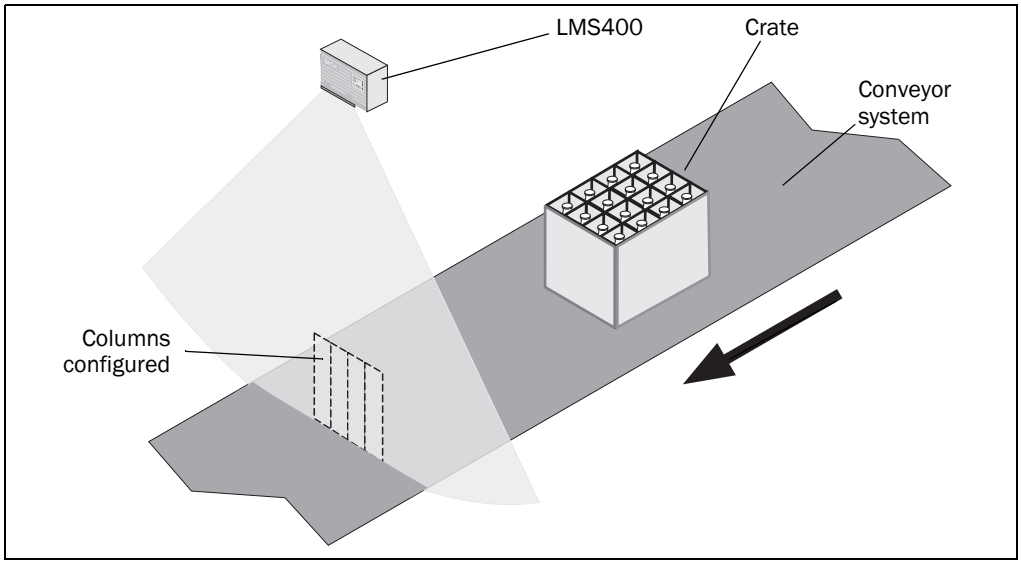


Fig. 11: Application example

When the crates move through the columns of the measurement area, among other aspects the tops on the bottles are scanned. This action results in all columns exceeding the switching points if the row of bottles is complete.

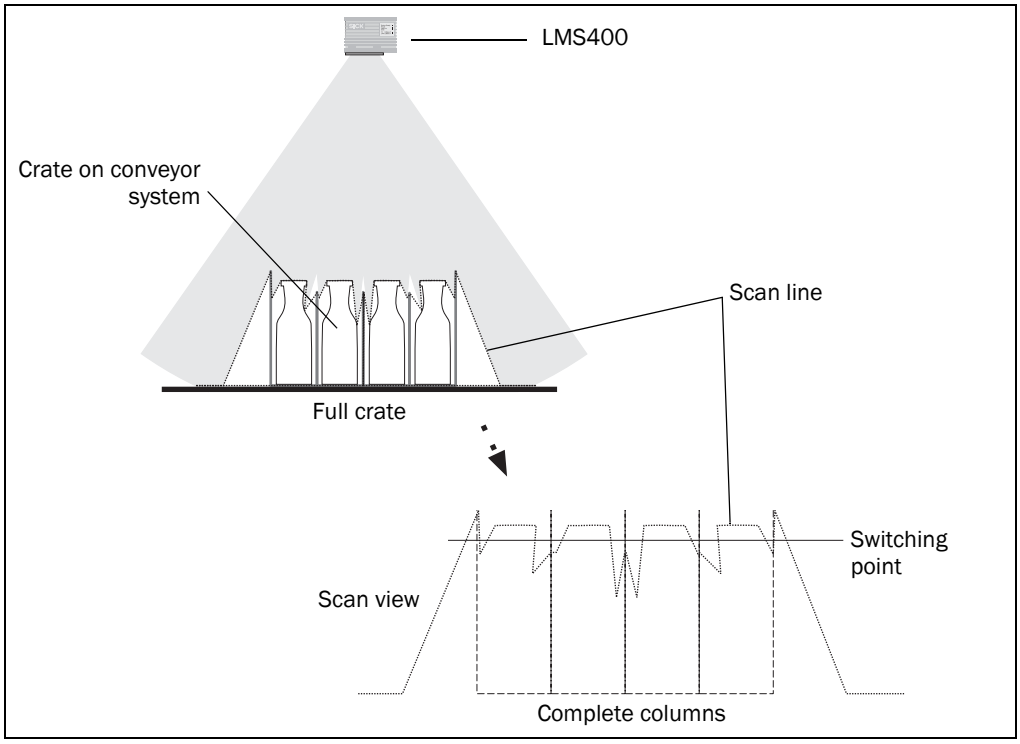


Fig. 12: Example for the evaluation of the columns

If one or more bottles are missing, the switching point for the related column is not reached, the crate is detected as not full.

The result can be indicated with the aid of the digital output on the LMS400:

- switching points exceeded on all columns = yes, output = high
- switching points not reached on all columns = no, output = low

Limits of the application

Due to the radial scanning during the measurement, fewer measured values are obtained in the outer columns than in the columns in the middle. The measurement is denser in the inner area.

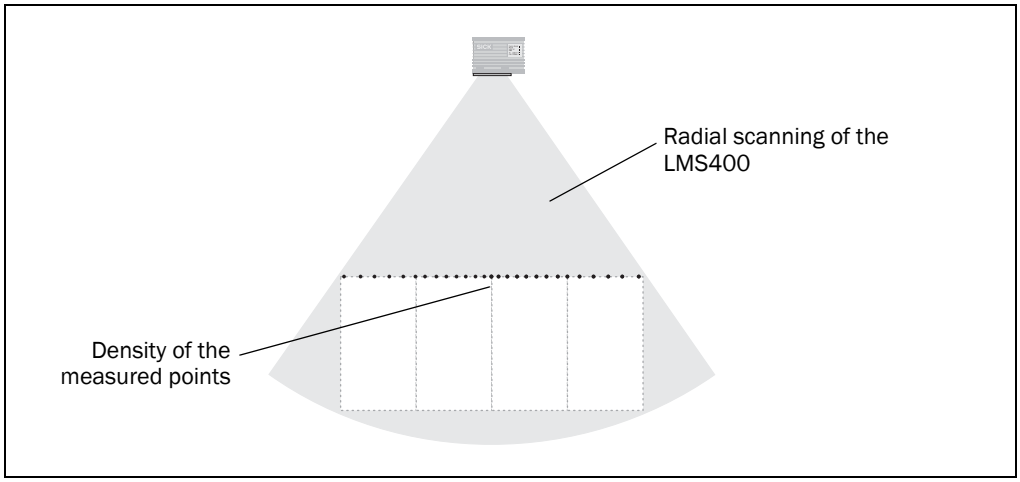


Fig. 13: Density of the measured points

Also shadows are produced during the measurements due to the radial scanning of the LMS400. The LMS400 cannot measure in these shadows. These shadows increase the higher the objects to be measured and the further the objects are from the vertical beam of the LMS400.

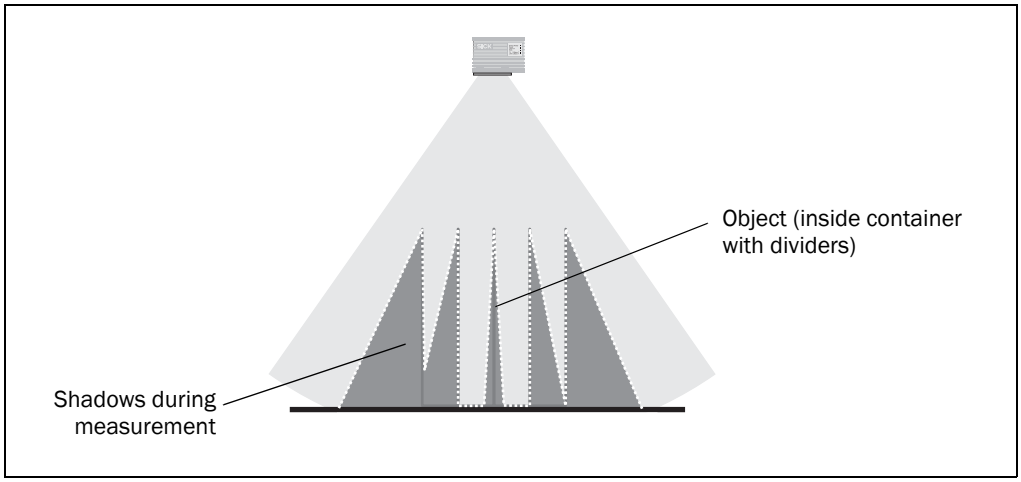


Fig. 14: Shadows during measurement

Recommendations

- Mount the LMS400 as high as possible over the objects.
- Mount the LMS400 centrally over the objects and define a measurement area centred on the vertical beam.

3.10 Configurable functions for Level Control

The parameters for the Level Control application are displayed in SOPAS ET immediately after you have activated the application.

PROJECT TREE, LMS400-XX00, PARAMETER, APPLICATION

3.10.1 Global zero point

As delivered the zero point for the distance measurement is at the origin of the laser (marked by a dot on the top and underside of the housing).

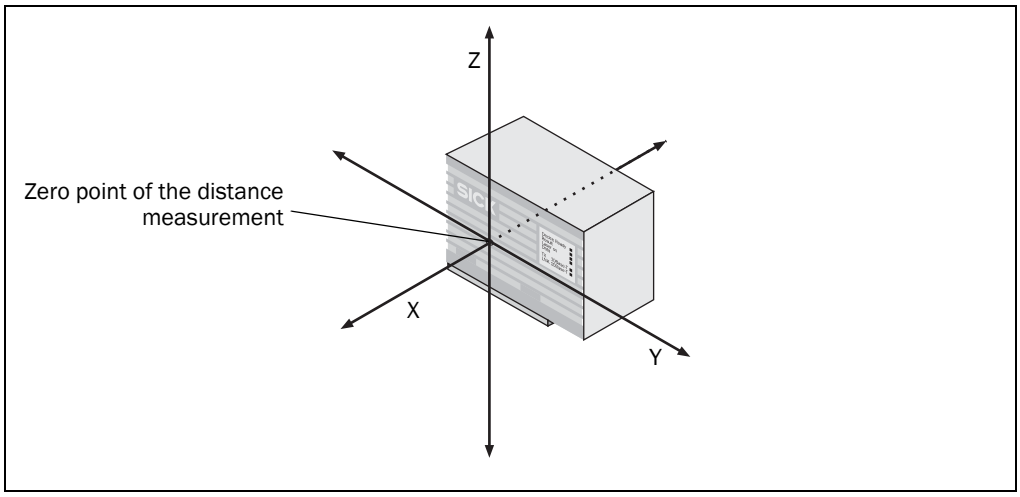


Fig. 15: Zero point of the LMS400 as delivered

In addition, the Level Control application requires a global zero point to which all other settings refer (e.g. right edge of the conveyor system as seen in the transporting direction). You can define this global zero point with the aid of the coordinates in SOPAS ET.

PROJECT TREE, LMS400-XX00, PARAMETER, POSITION, area COORDINATES
The installation setup in SOPAS ET will support you during the definition of the coordinates (see section *7.2.4 on page 68*).

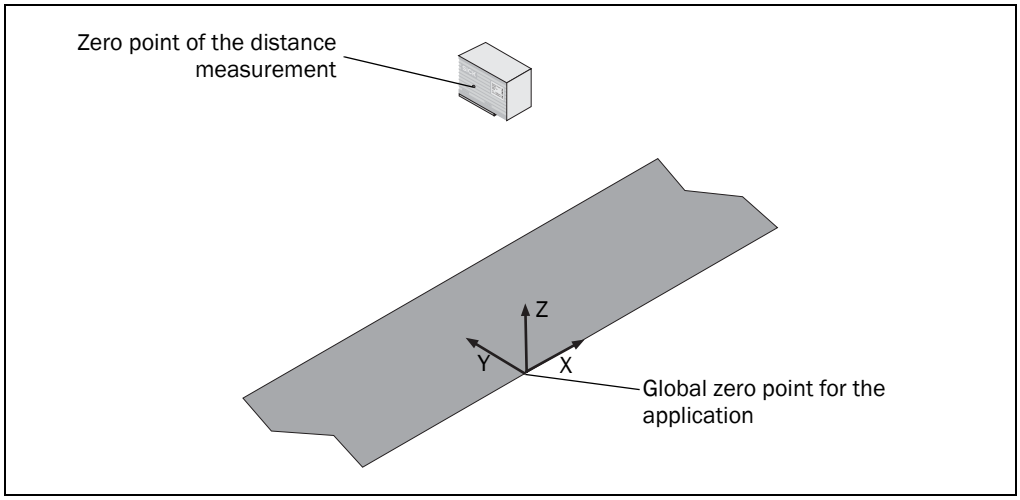


Fig. 16: Global zero point for the application

The global zero point for all the subsequent parameters for the application is defined by the coordinates.

Note **In all the following considerations Y extends to the left!**

3.10.2 Measurement area

You define the working area relevant for the Level Control application as the measurement area. You define for this measurement area a left and right border as well as a top and bottom border. All four values refer to the global zero point defined previously.

PROJECT TREE, LMS400-XX00, PARAMETER, POSITION, area MEASUREMENTAREA

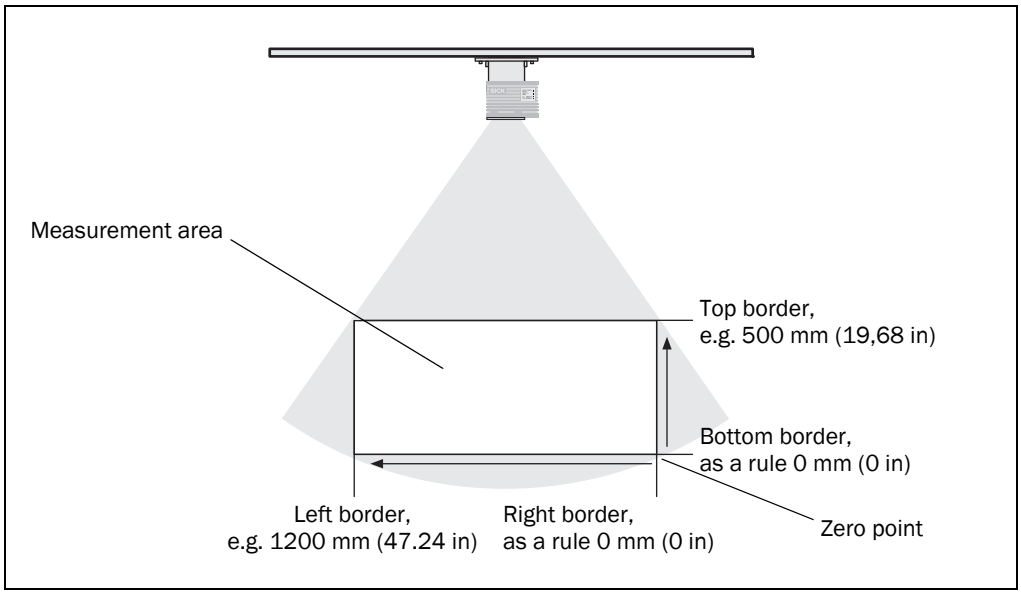


Fig. 17: Measurement area

Notes

- Ensure that the zero point is on the right and Y extends to the left. For this reason the value for the left border is always greater than the value for the right border.
- The bottom border can have a negative value, e.g. to include the conveyor system in the evaluation.
- The bottom border can have a positive value, e.g. to exclude the conveyor system from the evaluation.
- All measured values outside the measurement area are invalid and are not taken into account.

3.10.3 Evaluation in Y direction

To be able to evaluate the measurement results in the measurement area, you divide the measurement area into columns. During this process it is important that you adjust the size of the columns to suit your application.

PROJECT TREE, LMS400-XX00, PARAMETER, LEVEL CONTROL, COLUMN WIDTH

Equidistant column division

With the aid of the equidistant column division (equidistant = same distance apart) you can divide the measurement area into one to 50 columns of the same width. You can configure for all columns:

- a common switching point (Z level) with hysteresis
- a common evaluation mode
- a quality applicable to all columns

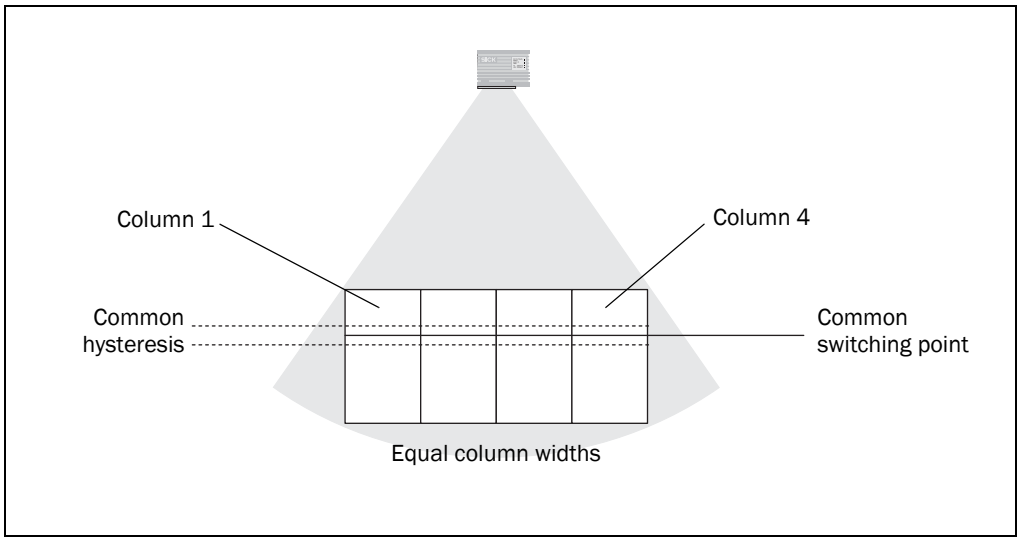


Fig. 18: Equidistant column division

Note The numbering of the columns starts at the left of the measurement area.

User-defined column division

With the aid of the user-defined column division you can divide the measurement area into one to 30 columns. You can configure for each column:

- a left and a right border

Note Note that Y extends to the left. For this reason the value for the left border is always greater than the value for the right border.
If you configure a column entirely or partially outside the measurement area, then the measured values inside the column but outside the measurement area are not taken into account.

- a dedicated switching point with hysteresis
- an individual evaluation mode
- a quality that applies only to this column

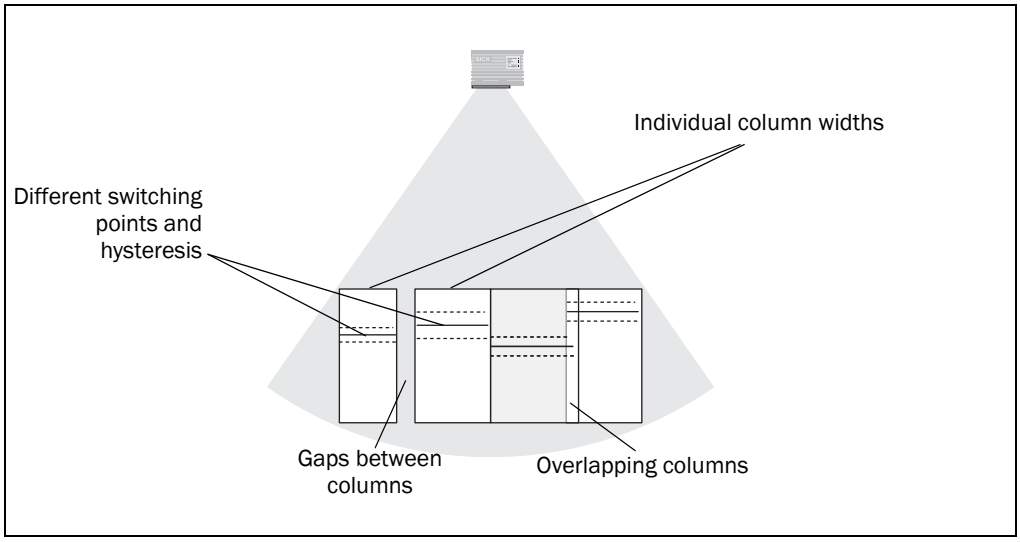


Fig. 19: User-defined column division

With the user-defined column division there is no fixed order for the columns. Columns can overlap and there can be gaps between columns; the gaps are not evaluated.

Definition of the quality necessary

Due to the formation of shadows and the radial measurement of the LMS400, there is a varying number of measured points in the columns in practically every scan (see also *"Limits of the application" on page 36*). By entering a quality you define how many measured points there must be in a column for the column result to be used in the evaluation.

Recommendation To be able to evaluate a column, there should be at least **three** measured values in the column (SOPAS ET default setting: quality = 3).

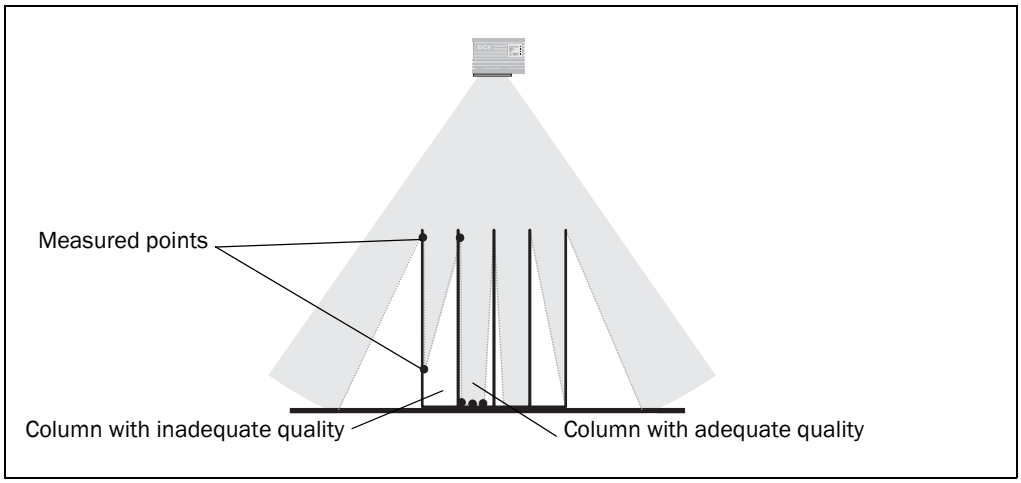


Fig. 20: Quality of the column result

If the quality is **not reached** for a column, the result for the column is discarded.

Column evaluation mode

With the mode you define how a measured value is determined for the measured points in a column.

- arithmetic mean
  The arithmetic mean is formed from the values measured.
  Example: **152**, **180**, **145** => 159
- min/max mean
  The mean from the two extreme values is formed from the values measured.
  Example: 152, **180**, **145** => 162.5
- maximum
  The highest value from the values measured is used for the evaluation.
  Example: 152, **180**, 145 => 180
- minimum
  The lowest value from the values measured is used for the evaluation.
  Example: 152, 180, **145** => 145

Switching point and hysteresis

You define a switching point in millimetres for each column. After the evaluation in the Y direction and X direction it is determined whether the results are above or below the switching point.

The column evaluation mode you have defined is crucial here. If you have, for instance, entered 160 mm (6.30 in) as the Z level, the value is only exceeded in the examples shown above in the "Min/max mean" and "Maximum" mode.

The hysteresis prevents the output value “flickering” if the measured value is near the switching point. The hysteresis is symmetrical around the switching point.
Example: A hysteresis value of 10 mm (0.39 in) on a switching point of 1000 mm (39.37 in) means that the column delivers the result configured when the column exceeds 1005 mm (39.57 in) and drops below 995 mm (39.17 in).

3.10.4 Evaluation in X direction

Along with the Y direction, you can also evaluate the columns in the X direction (transporting direction).

Note In SOPAS ET the output interval is set to “immediate” by default. In this way the evaluation in the X direction is skipped. You can configure a gate as the output interval for evaluation in the Y and also X direction.

PROJECT TREE, LMS400-XX00, PARAMETER, LEVEL CONTROL, COLUMN LENGTH, area PARAMETER, option OUTPUT INTERVAL

You can define the column length either with the aid of an external gate or an internal gate.

Determining the column length using an external gate

PROJECT TREE, LMS400-XX00, PARAMETER, DIGITAL INPUTS/TRIGGER, area GATE

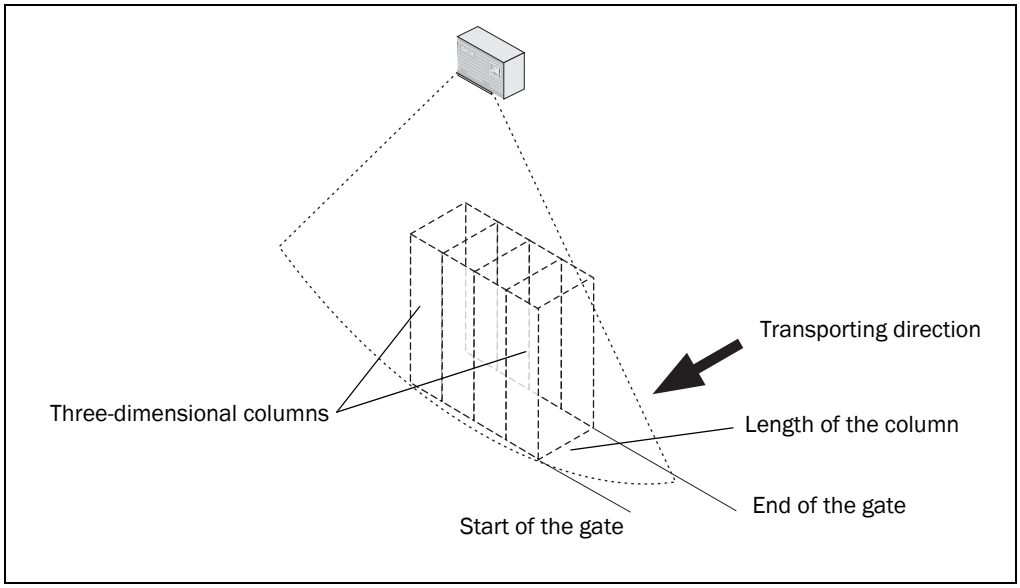


Fig. 21: External gate

If you use an external gate, then the start and end of the gate (that is e.g. switching on and off a photoelectric switch on a digital input) define the length of the columns.

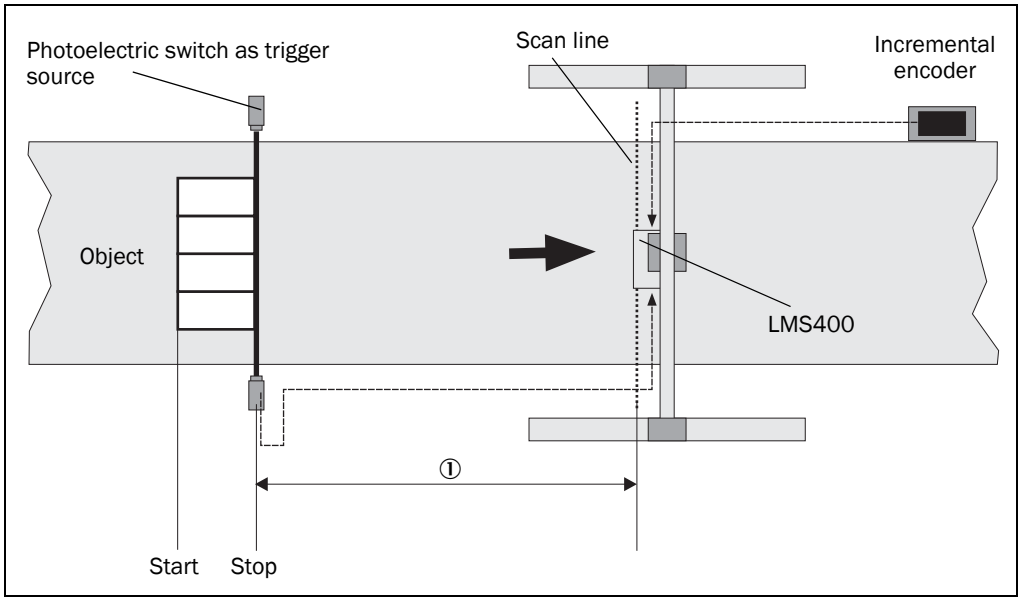


Fig. 22: Distance delay for the gate

Using a distance delay (can be configured in SOPAS ET) a photoelectric switch offset in relation to the scan line can be defined.

In the example, the object initialises the measurement start by entering the photoelectric switch. The end of the measurement is indicated by leaving the photoelectric switch.

However the actual measurement in the column only starts after the distance delay path has been covered ①. It ends as soon as the object leaves the scan line.

The column length is therefore defined by the length of the object, the actual timing of the measurement by the distance delay.

In addition, you can extend the gate produced by the triggering with a so-called expansion distance. This expansion distance is added to or deducted from the gate depending on whether it is positive or negative. It is always added or deducted on both sides. Using a negative expansion distance, e.g. the edges of the crate can be removed from the columns.

Determining the column length using an internal gate

PROJECT TREE, LMS400-XX00, PARAMETER, DIGITAL INPUTS/TRIGGER, ADVANCED IO SETTINGS, area INTERNAL GATE

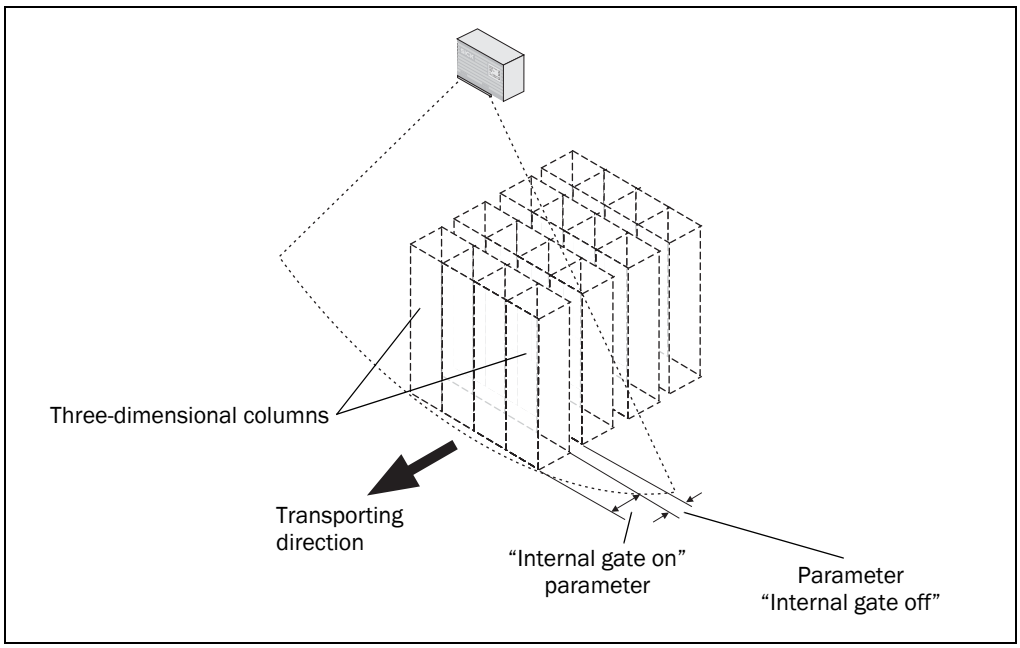


Fig. 23: Internal gate

If you use an internal gate, the "Internal gate on" parameter defines the length of the columns and the "Internal gate off" parameter the gap between two column series.

The columns can be "generated" continuously by entering the internal gate or instead limited by an external gate.

If you use an external gate, the start and end of the gate define the start and end of column generation. Between this start and end the internal gate defines the length of the columns.

A typical example application is the evaluation of a crate. The crate starts the gate by entering a photoelectric switch. The "Internal gate on" parameter is the same size as a bottle, the "Internal gate off" parameter the same size as the gap between two bottles. A pattern similar to that and in *Fig. 23* is produced.

About the evaluation of scans used

At large column lengths and low conveyor speed a high quantity of scans will be obtained that is inappropriate for the evaluation of a column in the X direction. For this reason you can define an interval in detected per millimetres. One scan is used for the measurement

in the X direction per distance entered (see *“Interval” on page 47*).

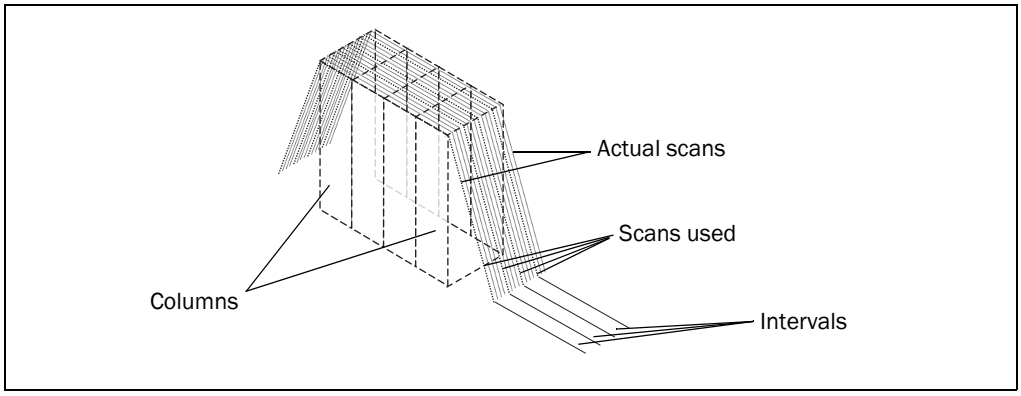


Fig. 24: Interval

Evaluation within the length of the column

The results evaluated in the Y direction are used for the evaluation. Only the valid results (with sufficient quality) are used.

You also define a quality for the X direction. This quality defines how many valid Y results there must be in the column for the X result (overall result) to be valid.

Using the mode you can define how a measured value is determined from the measured points in the X direction. The same options are available as for the Y mode (see *“Column evaluation mode” on page 41*).

3.10.5 Procedure in case of invalid values

Definition of the column value

You can define the value for the result of the evaluation for columns in which the quality is not reached. You have two options:

- calculate
  The column value is determined from the values present in the X direction that are actually inadequate. If a value cannot be determined, the bottom border of the measurement area is used as the value (see *3.10.2 “Measurement area” on page 39*).
- user defined
  The column has the value defined in SOPAS ET.

PROJECT TREE, LMS400-XX00, PARAMETER, LEVEL CONTROL, COLUMN LENGTH, area IRREGULAR COLUMN VALUE

Definition of the switching value

You can define the switching point for an invalid column independent of the column value for an invalid column. You have four options:

- calculate
  Even though the quality is not achieved, a switching value is determined from the measured values available and is used.
- over switching point
  Invalid switching values set the state to “Above switching point”.
- under switching point
  Invalid switching values set the state to “Below switching point”.

- retain value
  The previous state for the switching point is retained (above or below).

PROJECT TREE, LMS400-XX00, PARAMETER, LEVEL CONTROL, COLUMN LENGTH, area IRREGULAR SWITCHPOINT

3.10.6 Results of the column evaluation

You can output the result of the column evaluation using digital outputs or the analogue output.

Digital outputs

The LMS400 has four digital outputs. Per output you have four options for the allocation of the output.

PROJECT TREE, LMS400-XX00, PARAMETER, LEVEL CONTROL, RESULT, area DIGITAL OUTPUTS

- A single column switches the output if the switching point in this column is exceeded.
- A single column switches the output if the switching point in this column is not reached.

Note If the column selected is not activated or present, the output behaves as if you had configured it as inactive.

- All columns switch the output if the switching point in all columns is exceeded.
- All columns switch the output if the switching point in all columns is not reached.

Retention time

As soon as the selected criterion has occurred, the output switches to the active state (as a rule “high”, in case of inversion “low”). For each digital output you configure a retention time in milliseconds for which the output remains in this active output state. Adjust the retention time to the input electronics on the downstream system.

Note The output state is also retained for the duration of the retention time after the gate is shut down.

Logic

The digital outputs are high in the active state. However, you can also configure each digital output so that it is inverted. The output will then switch “low” in the active state.

Analog output

The LMS400 has an analogue output with a constant current source of 4-20 mA. You can allocate the output to a single column or all columns.

PROJECT TREE, LMS400-XX00, PARAMETER, LEVEL CONTROL, RESULT, area ANALOG OUTPUTS

- A single column defines the output current.

  The measurement area between the top and bottom border is represented linearly on the current range of the analogue output from 4 mA to 20 mA (4 mA means that the quality was not met).

Note If the column selected is not activated or present, the output behaves as if you had configured it as inactive.

- All columns switch the output (group evaluation).
  - output current = 4 mA, if the switching point is not reached in all columns
  - output current = 20 mA, if the switching point is exceeded in all columns
  - output current = 12 mA, if none of the previous two states is achieved

Retention time

You can configure a retention time in milliseconds for the analogue output. Adjust this retention time to the input electronics on the downstream system.

The output remains in the related state for this time, even if the column results change in the meantime. “Incoming” column results are processed during the retention time and output subsequently.

3.10.7 Output of telegrams

PROJECT TREE, LMS400-XX00, PARAMETER, LEVEL CONTROL, TELEGRAM, area SETTINGS

In SOPAS ET you can define which measured value telegram is to be output by the LMS400 if the telegrams are polled by a host (see *11.14.1 on page 131*).

- The “Column value” measured value telegram outputs the values measured in the columns in detail (see *Tab. 112 on page 134*).
- The “Switching point states” measured value telegram outputs whether the switching point in a column is exceeded or not reached (see *Tab. 113 on page 136*).

In addition, you can control the output of the telegrams via the switching point for a column.

- Telegram output starts (after polling by a host, see *11.14.1 on page 131*) as soon as the switching value for a specific column is exceeded.
- Telegram output stops as soon as the switching value for this column is not achieved.

3.10.8 Filtered measurement

Note The area filter, the mathematical filter and the median filter are not available for the Level Control application. The filters are hidden in SOPAS ET.

Edge filter, Z median filter and interval act on the measured values before the column values are determined.

If several filters are active, then the filters act one after the other on the result of the previous filter. The processing in this case follows the following sequence:

- edge filter
- interval
- Z median filter

Edge filter

See section *3.7.1 “Edge filter” on page 29*.

Z median filter

PROJECT TREE, LMS400-XX00, PARAMETER, FILTER, area LEVEL CONTROL

A special median filter is available for the Level Control application. This forms a 3 by 3 median from the Z values for the measured points (and not, like the median filter, from the polar distance values).

Interval

PROJECT TREE, LMS400-XX00, PARAMETER, FILTER, area LEVEL CONTROL

The number of scans in a column is defined as a function of the column length, conveyor speed and scanning frequency. At large column lengths and low conveyor speed a quantity of scans will be obtained that is inappropriate for the evaluation of a column in the X direction.

In this case with the aid of an interval you can define the magnitude of the distance between two scans that are to be used for the measurement (see also *3.10.4 “Evaluation in X direction” on page 42*).

Note If more than 3000 scans are reached per column length, an error occurs that is output in the measured value telegram under “Status” (see *11.14.1 on page 131*).

3.11 Interface specification

The LMS400 has three different interfaces for the configuration for the transmission of measured values. You can configure the LMS400 and also receive measured values via each of these interfaces.

Note It is only possible to output all measured values of a scan in real-time using the Ethernet interface.

3.11.1 Ethernet interface

The Ethernet interface has a data transmission rate of 10 MBaud (10Base-T). The interface is a TCP/IP peer to peer interface. Only half duplex is supported. Please ensure that the interface of your application is set to half duplex.

The factory setting for the Ethernet interface is as follows:

- IP ADDRESS: 192.168.0.1
- TCP/IP PORT: 2111
- SUBNET MASK: 255.255.255.0

If necessary, adjust the TCP/IP configuration for the Ethernet interface to enable a connected PC (client) to communicate with the LMS400 via Ethernet: PROJECT TREE, LMS400-XX00, INTERFACE, ETHERNET, area ETHERNET

Notes

- The parameters for the Ethernet interface can only be configured using SOPAS ET if the PC is connected via the Aux interface or the host interface.
- If, on the other hand, the LMS400 is configured using telegrams (see *11.11 “Configuration telegrams for the Ethernet interface” on page 123*), the Ethernet interface can then also be configured using telegrams, if the host is connected to the Ethernet interface.
- To make the changes to the interface parameters effective, after configuration the LMS400 must be reset (see *2.5 “Quick stop and Quick restart” on page 18*).

You will find a description of the electrical interface in section *5.3.1 ““Ethernet” connection” on page 58*.

3.11.2 Aux interface

The Aux interface allows the configuration of the LMS400 as well as the output of measured values. However, this feature is primarily intended to provide a reliable data connection for configuration (also with simultaneous operation of the host interface). Therefore, the following interface parameters can not be changed:

- 9600 Baud
- 8 data bits
- 1 stop bit
- no parity

You will find a description of the electrical interface in section *5.3.4 ""Serial" connection" on page 60*.

3.11.3 Host interface

The host interface allows the configuration of the LMS400 as well as the output of measured values.

You can choose whether to configure the pins 6 to 9 as RS-232 or as RS-422: PROJECT TREE, LMS400-XX00, INTERFACE, SERIAL, area SERIAL HOST, option HARDWARE

The interface parameters are freely configurable. The factory setting for the host interface is as follows:

- RS-232
- 9600 Baud
- 8 data bits
- 1 stop bit
- no parity

Note The interface parameters for the host interface can be configured only using the Aux interface or the Ethernet interface.

You will find a description of the electrical interface in section *5.3.4 ""Serial" connection" on page 60*.

3.12 Data communication using telegrams

The LMS400 sends telegrams over the interfaces described above to communicate with the connected application. The following functions can be run using telegrams:

- request for measured values by the application and subsequent output of the measured values by the LMS400
- parameter setting by the application for the configuration of the LMS400
- parameters and status log querying by the application

The telegrams each comprise a frame (see *3.12.1 on page 50*) and the data. Different telegram types are used as data:

- methods for querying measured values
- variables for configuration
- methods for configuration

These three types have different syntaxes. This situation must be taken into account on writing the software interface for your application (see *3.12.2* and *3.12.3*).

A detailed description of the different telegrams can be found in the annex (see *11.2 "Overview of the telegrams" on page 80*).

3.12.1 Frame and coding for the telegrams

The data is placed in different frames and coded differently depending on the interface used.

Telegrams on the aux interface or the host interface

The table shows the pre-setting for the frame for the aux and host interfaces.

	Frame	Telegram	Frame
Code	STX	Data (see *11.2 on page 80*)	ETX
Length (byte)	1	≤2498	1
Description	Start of text character	ASCII coded. The length is dependent on the previous send telegram.	End of text character

Tab. 14: Frame for the telegrams on the aux interface or the host interface

The frame for the host interface can be configured in SOPAS ET or via configuration telegrams: PROJECT TREE, LMS400-XX00, INTERFACE, SERIAL, area SERIAL HOST or *11.10.5 on page 118* to *11.10.9 on page 122*.

In this way, you can use two stop bytes, for example (e.g. to terminate telegrams with CR/LF), or insert a block check byte before or after the stop byte.

Telegrams on the Ethernet interface

	Frame						Telegram	Frame	
Code	TCP/IP Start Frame	STX	STX	STX	STX	Telegram length	Data (see *11.2 on page 80*)	Check-sum	TCP/IP Stop Frame
Length (byte)	Defined by the trans-mission	1	1	1	1	4	≤2495	1	Defined by the trans-mission
Description		Start of text character				Data length without CS, Motorola format	Binary encoded. The length is dependent on the previous send telegram.	See *"Calcu-lation of the check-sum"* further below	

Tab. 15: Frame for the telegrams on the Ethernet interface

Calculation of the checksum

The checksum is calculated using an XOR operator for every byte of the data, that is without the frame.

3.12.2 Methods for querying measured values

There are various measured value telegrams for triggering measured values output. Measured values are requested using telegrams in four steps:

1. The terminal sends a measured value telegram, comprising the command **sMN** (SOPAS method by name), the telegram name and one or more parameters.
2. The LMS400 sends a confirmation that the telegram is being processed, comprising the confirmation **sMA** (SOPAS method acknowledge) for the telegram name.
3. After the processing of the telegram, the LMS400 sends a reply that the parameters have been set, comprising **sAN** (SOPAS answer), the telegram name for an error code (00000000 = no error).
4. The LMS400 sends measured values continuously.

   Or (on control with the aid of a trigger):

   The LMS400 sends the measured values as long as a trigger is present.

   Or (on sampling a specific number of scans):

   The LMS400 sends as many measured values as have been requested.

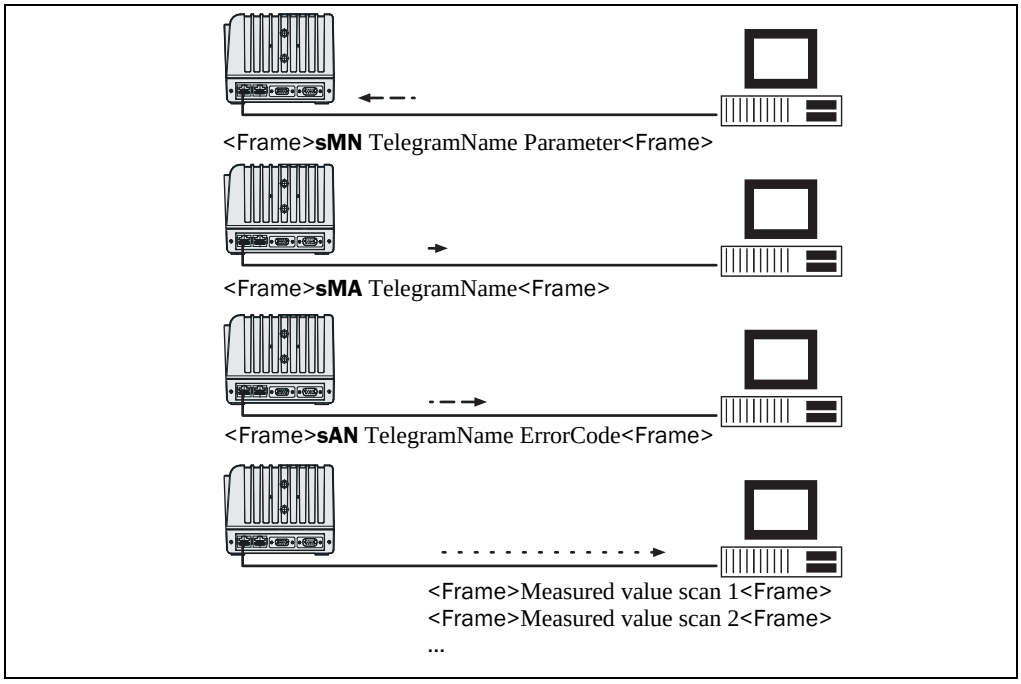


Fig. 25: Use of methods for querying measured values

3.12.3 Variables and methods for configuration

Variables for configuration

Variables are used if parameters can be set immediately by the LMS400 and confirmed. Configuration using variables is carried out in two steps:

1. The terminal sends a telegram, comprising the command **sWN** (SOPAS write by name), the telegram name and one or more parameters.
2. The LMS400 sends a reply that the parameters have been set, comprising **sWA** (SOPAS write answer) for the telegram name.

Note Some of the telegrams also return the result.

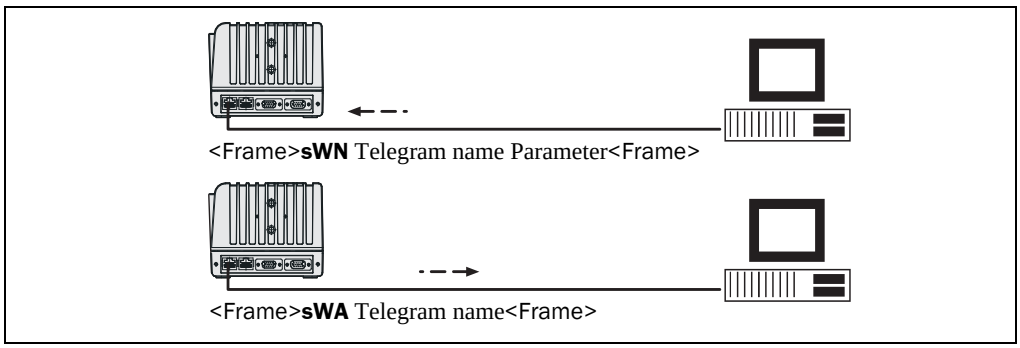


Fig. 26: Use of variables for configuration

Methods for configuration

Methods are used if the LMS400 needs a certain amount of time to set the parameters (e.g. to change its measuring frequency). Configuration using methods is always carried out in three steps:

1. The terminal sends a telegram, comprising the command **sMN** (SOPAS method by name), the telegram name and one or more parameters.
2. The LMS400 sends a confirmation that the telegram is being processed, comprising the confirmation **sMA** (SOPAS method acknowledge) for the telegram name.
3. After the processing of the telegram, the LMS400 sends a reply that the parameters have been set, comprising **sAN** (SOPAS answer), the telegram name for an error code (00000000 = no error).

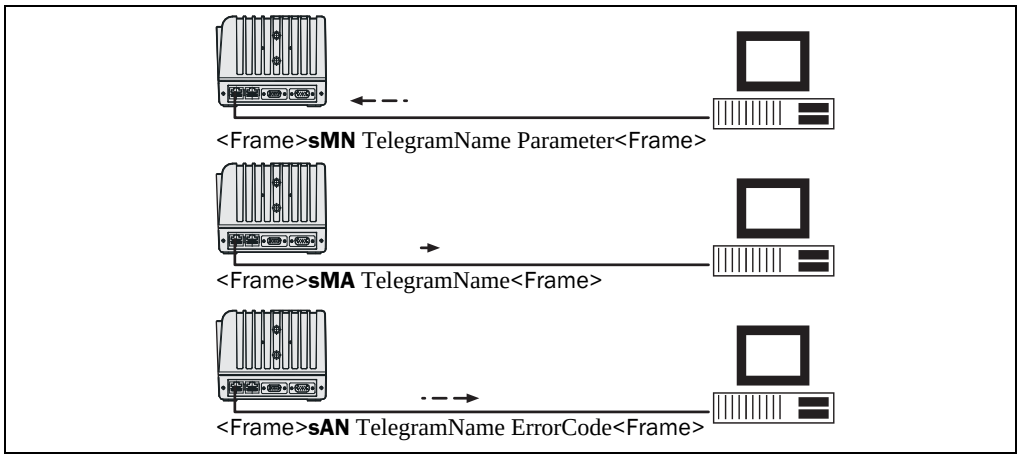


Fig. 27: Used of methods for configuration

Note The two methods for setting the basic parameters sMN mSCconfigbyfreq and sMN mSCconfigbyang also return the parameters actually used and the measured value quality (see *11.4.2 on page 91* and *11.4.3 on page 93*).

Sequence during configuration using telegrams

In exactly the same way as for configuration with the aid of SOPAS, certain parameters on the LMS400 are only allowed to be set or changed with appropriate authorisation. The modified parameters must also be permanently saved in the device following completion of the configuration. The table shows in principle the sequence during configuration with telegrams.

Telegram	Description
sMN SetAcessMode	Login
sWN Configuration telegram 1 ... sWN Configuration telegram n	Configuration steps
sMN mEEwriteall	Save parameters permanently
sMN Run	Terminate configuration

Tab. 16: Sequence during configuration using telegrams

Note With the two methods for setting the basic parameters sMN mSCconfigbyfreq and sMN mSCconfigbyang the procedure is different: Prior login is not necessary as this is part of the telegrams (see *11.4.2 on page 91* and *11.4.3 on page 93*).

3.13 Status indicators

Six LEDs on the LMS400 laser measurement sensor provide a visual indication of the operating status and any errors that have occurred.

LED	Display	Function
Device Ready	Green	Initialisation and self test successful. Device ready for operation
	Red	Error on initialisation or self-test or on the occurrence of errors in operation
Result	Off	Not assigned
Laser On	Green	Laser diode on
Data	Green	Flashes when the LMS400 is transmitting data over the host interface
LNK 10Base-T	Green	10Base-T: Ethernet link
TX 10Base-T	Orange	10Base-T: Flashes when the LMS400 is transmitting data to the computer over the Ethernet interface

Tab. 17: Meaning of the LEDs

4 Mounting

4.1 Mounting requirements

The LMS400 must be mounted as follows:

- robust (weight LMS400: approx. 2.3 kg resp. 5.1 lb)
- without vibration
- without oscillations

Important **Only if the LMS400 is mounted so it is not subjected to shocks or vibration can an optimal measurement result be expected!**

4.2 Overview of the mounting steps

- Install mounting kit for LMS400.
- Mount LMS400 on mounting kit.
- Connect LMS400 to the voltage supply.

4.3 Preparations for mounting

4.3.1 Components to be mounted

- an LMS400 laser measurement sensor (weight approx. 2.3 kg resp. 5.1 lb)
- a mounting kit for the LMS400
- a set of connection cables

4.3.2 Place material at hand

The following materials are required for the mounting kit and for the LMS400:

- M6 screws for mounting the mounting kit to the frame or other assembly. (The mounting kit has eight D6.6 holes.)
- three M8 × 12 screws with washers for fastening the LMS400 to the mounting kit (included in the delivery)
- tool set

4.4 Mounting with mounting kit No. 2030421

A mounting kit can be supplied for mounting the LMS400. This can be finely adjusted in two axes.

Note The LMS400 must be installed such that the intensity of light from external sources does not exceed 2 kLux.

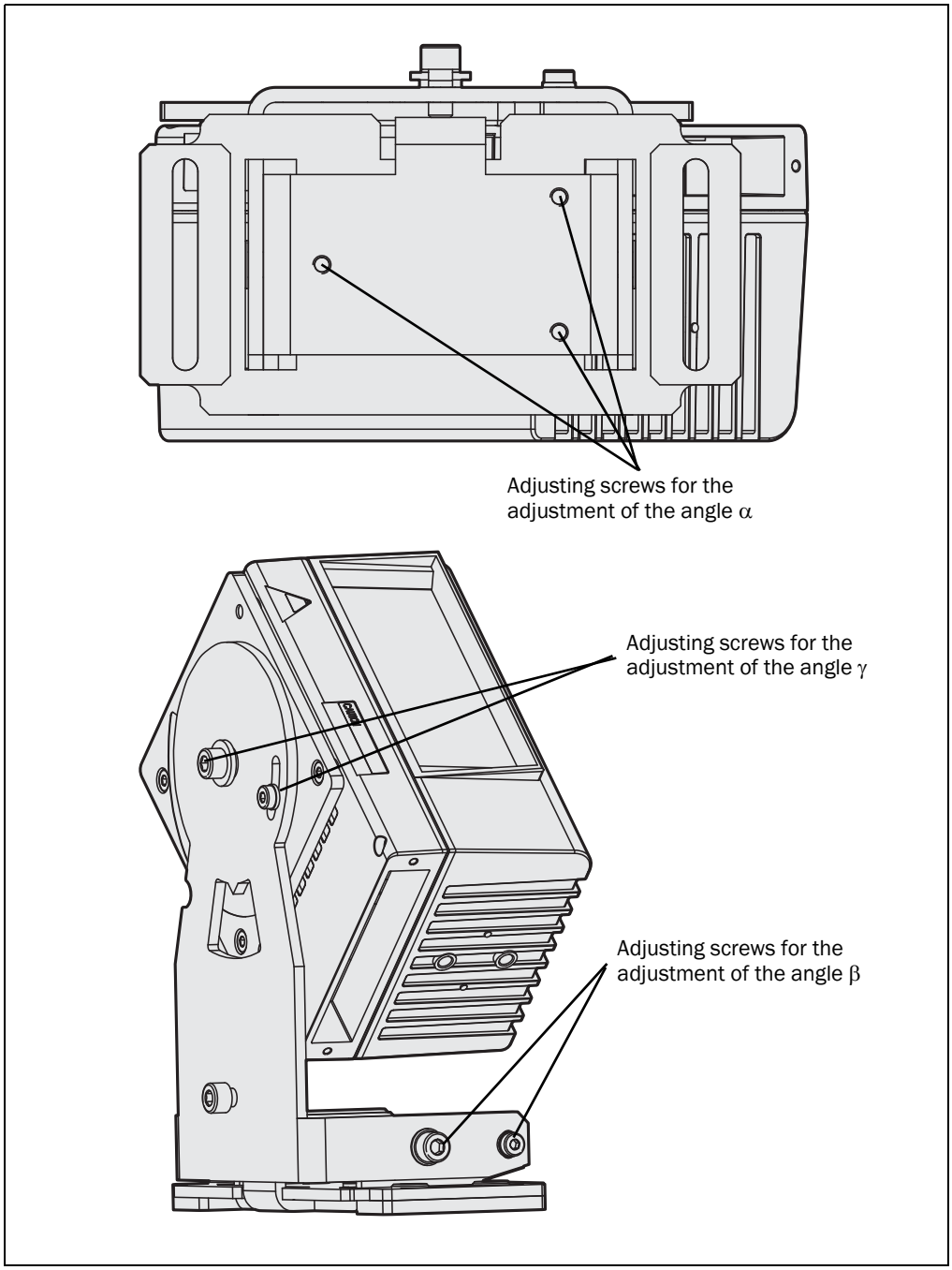


Fig. 28: Mounting kit for LMS400

Note You will find the dimensional drawing for the mounting kit in section *10.2.2 “Dimensional drawing mounting kit No. 2030421 for LMS400” on page 78*.

4.5 Dismantling the sensor

1. Switch off the supply voltage.
2. Remove the connection cables.
3. Remove LMS400 from the mounting kit.

Note On final decommissioning, please observe the disposal requirements in section *8.2 "Disposal" on page 72* for environmentally correct disposal.

5 Electrical installation

ATTENTION

Switch the entire machine/system off line!

The machine/system could inadvertently start up while you are connecting the device.

➢ Ensure that the entire machine/system is disconnected during the electrical installation.

Note The LMS400 laser measurement sensor complies with the requirements in the standard on the radiated emissions as defined for class A (industrial environment). It may cause radio interference in residential areas. If radio interference occurs, the person(s) affected may demand that the operator take appropriate action for suppressing interference.

5.1 Overview of the installation steps

- Connect supply voltage to the LMS400.
- Connect PC to the aux interface of the LMS400.

5.2 Electrical connections and cables

In the electrical connections, the LMS400 has interfaces for communication between master for slave as well as interfaces to the exterior.

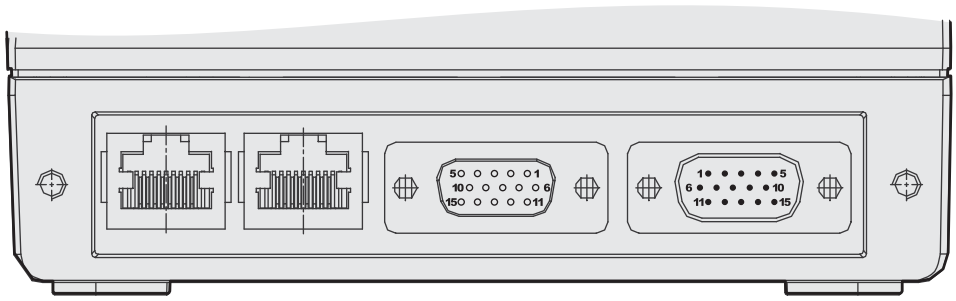


Fig. 29: Position of the electric connections of the LMS400

Connection	Type	Function
Ethernet	RJ-45	TCP/IP communication, exchange of telegrams
System	RJ-45	Synchronisation master/slave
I/O	D-Sub	Connection of external sensors, supply voltage
Serial	D-Sub	Serial communication, exchange of telegrams, supply voltage

Tab. 18: Function of the electric connections of the LMS400

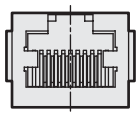
Notes

- You can connect the supply voltage alternatively at the “I/O” connection or the “Serial” connection.
- To quickly connect the LMS400 to a host or a PLC in a manner suitable for industrial use, the LMS400 can be connected using a connection module (CDM490) and/or a plug cover can be connected (see *5.4 “Connection via connection module or plug cover” on page 61*).

5.3 Pin assignment of the connections

5.3.1 “Ethernet” connection

The LMS400 can be connected with a standard Cat. 5 patch cable.

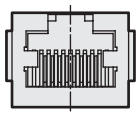


Pin	Signal	Function
1	TX+	Ethernet interface
2	TX–	Ethernet interface
3	RX+	Ethernet interface
4	Not assigned	Do not use!
5	Not assigned	Do not use!
6	RX–	Ethernet interface
7	Not assigned	Do not use!
8	Not assigned	Do not use!

Tab. 19: Pin assignment of the “Ethernet” connection (8-pin RJ45 female connector)

5.3.2 “System” connection

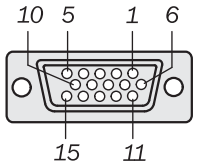
Via the “System” connection master and slave of two connected LMS400 are synchronised. For the connection of master and slave a standard Cat. 5 crossover cable is suitable.



Pin	Signal	Function
1	FSIOP	Synchronisation master/slave
2	FSION	Synchronisation master/slave
3	FSIIP	Synchronisation master/slave
4	Not assigned	Do not use!
5	Not assigned	Do not use!
6	FSIIN	Synchronisation master/slave
7	Not assigned	Do not use!
8	Not assigned	Do not use!

Tab. 20: Pin assignment of the “System” connection (8-pin RJ45 female connector)

5.3.3 “I/O” connection



Pin	Signal	Function
1	V_S	Supply voltage Sensor
2	IN3	Input 3 (trigger), digital
3	IN1	Input 1 (trigger), digital
4	OUT1	Output 1, digital
5	GND	Ground Sensor
6	IN2	Input 2 (encoder), digital
7	IN4	Input 4 (encoder), digital
8	OUT2	Output 2, digital
9	GND IN1...4	Ground Inputs 1 ... 4
10	OUT3	Output 3, digital
11	Reserved	Do not use!
12	Reserved	Do not use!
13	OUTA	Output Analog (4 mA ... 20 mA)
14	GND OUTA	Ground Output Analog
15	OUT4	Output 4, digital
Housing	–	Shield/earth

Tab. 21: Pin assignment of the “I/O” connection (D-Sub-HD female connector)

Notes

- The maximum frequency on the inputs IN2 and IN4 (encoder) is 10 kHz.
- The maximum output current on the digital outputs 1 ... 4 is 400 mA. If a load is not connected to the outputs, the outputs will exhibit tristate behavior. Normal switching behavior is achieved on the termination with 10 kOhm.

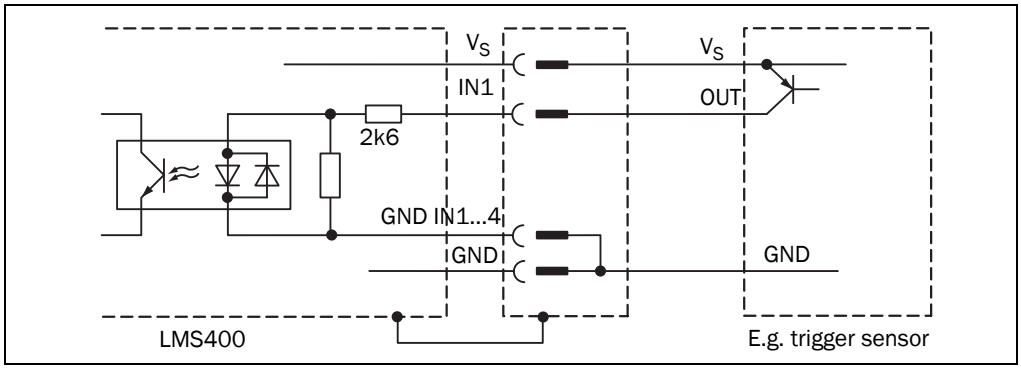


Fig. 30: Connection diagram for digital input

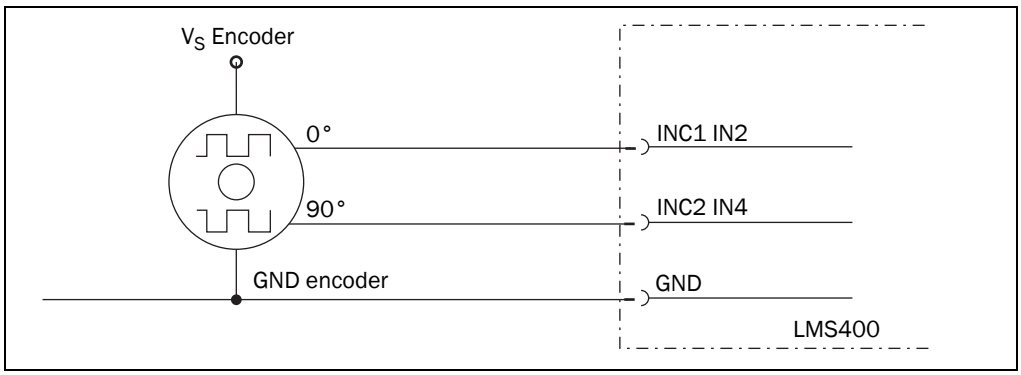
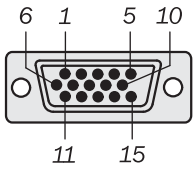


Fig. 31: Connection diagram for encoder inputs

5.3.4 “Serial” connection

You can choose whether to configure the pins 6 to 9 as RS-232 or as RS-422.

PROJECT TREE, LMS400-XX00, INTERFACE, SERIAL, area SERIAL HOST, option HARDWARE



Pin	Signal	Function
1	V_S	Supply voltage Sensor
2	RxD	Receiver RS-232 Aux
3	TxD	Sender RS-232 Aux
4	Reserved	Do not use!
5	GND	Ground Sensor
6	RD+	Receiver+ RS-422 Host
7	RD–/RxD	Receiver– RS-422/RS-232 Host
8	TD+	Transmitter+ RS-422 Host
9	TD–/TxD	Transmitter– RS-422/RS-232 Host
10	Reserved	Do not use!
11	Reserved	Do not use!
12	Reserved	Do not use!
13	Reserved	Do not use!
14	Reserved	Do not use!
15	Reserved	Do not use!
Housing	–	Shield/earth

Tab. 22: Pin assignment of the “Serial” connection (D-Sub-HD male connector)

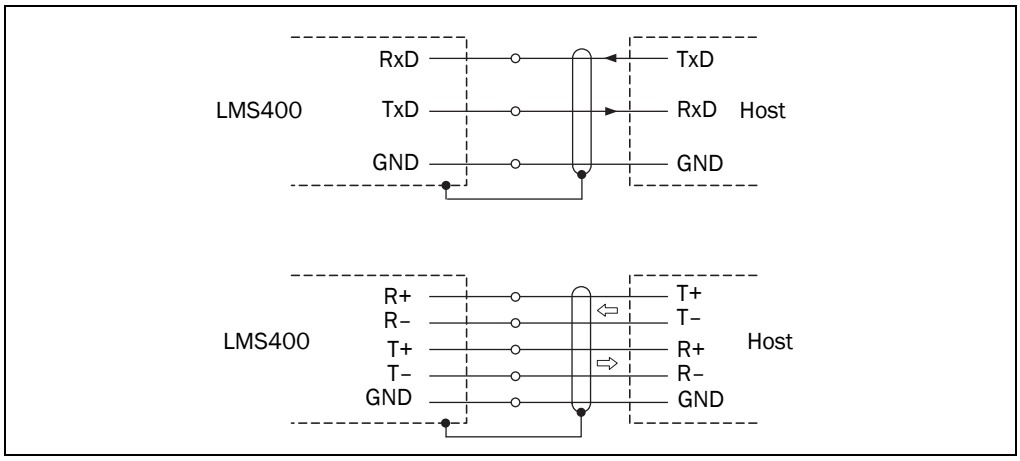


Fig. 32: Connection of RS-232 or RS-422 interface

5.4 Connection via connection module or plug cover

To quickly connect the LMS400 to a host or a PLC in a manner suitable for industrial use, the LMS400 can be connected using a connection module (CDM490) and/or a plug cover (e.g. No. 2030439 oder No. 2030535) can be connected. Since the plug cover covers the electrical connections of the LMS400, IP 65 degree of protection is achieved.

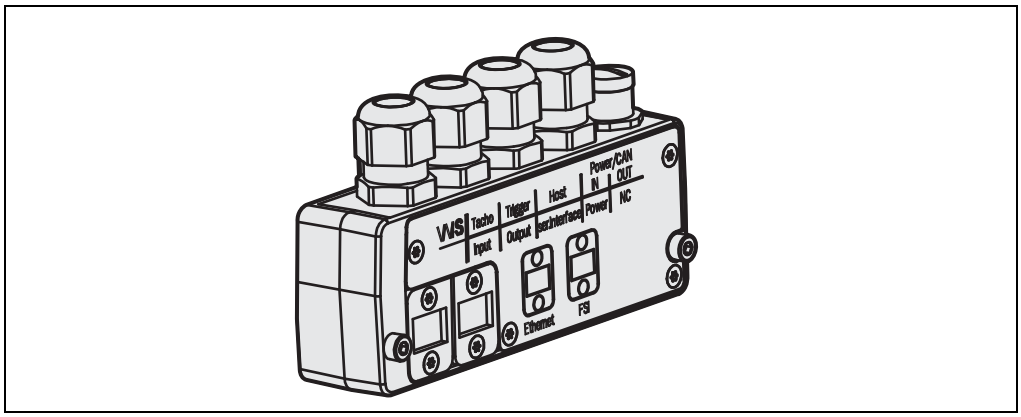


Fig. 33: Example of a plug cover

Parameter memory

The plug cover contains a parameter memory. The connection module can be extended using a parameter memory. The parameter memory makes it easy to replace an LMS400. The configuration is stored in this memory and transmitted to the newly connected LMS400 after replacement.

The **S2-Mode** rotary switch defines whether the parameter memory in the plug cover is used:

- ➢ Set the switch to the **F** position to use the parameter memory.
- ➢ Set the switch to the **0** position to not use the parameter memory.

Notes

- Only replace the LMS400 with a device with the same firmware version. Only then is it ensured that all parameters are transferred to the new device connected.
- If the parameter memory contains settings for the Ethernet interface, these settings are not effective immediately on transfer to the new LMS400. The transfer of the Ethernet parameters to the LMS400 is similar to changing the parameters. For this reason the LMS400 must first be reset (see *2.5 “Quick stop and Quick restart” on page 18*).
- The parameter set stored in the parameter memory always overwrites the configuration of the LMS400. An already configured device, for example, is reset to factory settings by a new plug cover.
- Special Ethernet cables are required to connect the plug cover (see *11.15 “Ordering information” on page 138*).

A detailed description of the electrical connections of the plug cover can be found in Document No. 8010817 — “Plug Covers for Volume Measurement System VMS400/500 and Laser Measurement Sensor LMS400”.

5.5 Performing the electrical installation

5.5.1 Overview of the connection steps

- Connecting the voltage supply.
- Connect host interface.
- Connect PC (connect aux interface).

5.5.2 Equipment

- tool set
- digital multimeter (current/voltage measurement)

5.5.3 Connecting the voltage supply

Note You can connect the supply voltage alternatively at the “I/O” connection or the “Serial” connection.

1. Ensure that the supply voltage is switched off.
2. Connect the 24 V input “Supply voltage” on the LMS400 to the corresponding connection on the voltage supply using a cable.

5.5.4 Connect PC

The LMS400 is operated and configured using the SOPAS ET configuration software.

1. Switch off PC and supply voltage.
2. Connect the PC and the aux interface of the LMS400 using a three-core RS-232 data cable (null modem cable).
3. Switch on PC and supply voltage.
4. Configure (see chapter *7 “Configuration (parameterisation)” on page 64*).

6 Commissioning

Commissioning requires a thorough check by qualified personnel!

Before you operate a system equipped with the LMS400 laser measurement sensor for the first time, make sure that the system is first checked and released by qualified personnel. On this issue, observe the notes in chapter *2 "For your safety" on page 13*.

Overview of the commissioning steps

1. Mounting the LMS400 (see chapter *4 "Mounting" on page 54*).
2. Performing the electrical installation (see chapter *5 "Electrical installation" on page 57*).
3. Configuring LMS400 using the SOPAS ET configuration software (see chapter *7 "Configuration (parameterisation)" on page 64*).

Note The RS-232 and RS-422 host interfaces cannot be used simultaneously during operation.

7 Configuration (parameterisation)

Note Software access to the LMS400 is password protected. Following completion of the configuration, you should change the password so that it can perform its protective function.

Firmware	User level	Password
V 1.20 or higher	Maintenance personnel	main
	Authorized client	client

Tab. 23: Passwords

You can configure the LMS400 laser measurement sensor in two ways:

- interactively using the provided SOPAS ET configuration software
  This section describes the interactive configuration.
- using configuration telegrams
  On this topic please read section *3.12 “Data communication using telegrams” on page 49* and refer to section *11.2 “Overview of the telegrams” on page 80*.

The interactive configuration is carried out using the provided SOPAS ET configuration software. Using this configuration software, you can configure and test the measurement properties, the analysis behaviour and the output properties of the sensor as required.

Help for the program user interface as well as for the different options can be found in SOPAS ET:

- menu HELP, HELP: Comprehensive online help for the program interface and the different options
- HELP window (on the bottom left in the program user interface): Context sensitive help for the visible dialog
- tool tips: Move the mouse pointer over an input field. A short text (“tool tip”) with information about valid entries appears.

7.1 Configuration for measured value output

To configure the LMS400 laser measurement sensor, you need:

- SOPAS ET configuration software (not included in scope of delivery)
- PC/notebook (Windows 98, NT 4.0, 2000, XP, Vista, 7, 8 or 10) with Ethernet interface (RJ45) and optional serial interface (RS-232).
  Device not included in scope of delivery.
- Optional for configuration via serial interface (RS-232): a three-core RS-232 connection cable (null modem cable) for connecting PC and LMS400.
  Cable not included in scope of delivery.

How to prepare the configuration:

➢ Make sure that the LMS400 laser measurement sensor has been correctly mounted and that the electrical connections are correct and in place.

➢ Plan all required settings (operating modes, beam coding, resolution, measured value filtering, trigger etc.).

➢ Connect the PC/notebook with the aux interface of the LMS400. The connection of the PC is described in section *5.5.4 “Connect PC” on page 62*.

➢ Install the provided SOPAS ET configuration software.

Note The second generation of SOPAS ET (version 2.38.3) can continue to be used, although support for it is no longer being provided. To receive updates or support, please use the latest version, i.e., the third generation of SOPAS ET (version 3.xx).

Download and installation of SOPAS ET

The configuration software SOPAS ET, the current system prerequisites for the PC, and the instructions for downloading the software and the device description file(s) can be found in the Web at: *www.sick.com/sopas*.

Procedure for SOPAS V3.x:

1. Start PC.
2. Download and install version V3.x of the SOPAS ET configuration software from the online product page for the software by following the instructions provided there. Administrator rights may be required on the PC to install the software.
3. Start the “SOPAS ET” program option after completing the installation.
   Path: C:\Program Files (x86)\SICK\SOPAS ET\SopasET.exe or via Windows search.
4. Install the device driver (SDD) in the device catalog using the wizard (gear symbol). The *.jar file can be obtained from the online repository if an Internet connection is present.
5. In the device search list, establish a connection between SOPAS ET and the LMS400 using the search settings. To do this, select the LMS4xx family of devices and select the default IP address 192.168.0.1 when connecting for the first time. The device is detected and can now be integrated into a project for configuration purposes.

Note

- The installation file for SOPAS ET and the *.jar file for the device driver can also be exported to a data card in order to install them on another PC when there is no Internet connection.
- The LMS4xx does not support Automatic Private IP Addressing (auto-IP). For this reason, the device is not detected by the default search. To ensure the device search is successful, the correct IP address and a sufficiently precise address range must be specified. It is possible to save a search setting that has been set up specifically for this purpose under a separate name (default: IP address = 192.168.0.1, subnet mask = 255.255.255.0).

7.1.1 Performing the configuration

Use the project tree in SOPAS ET to configure the parameters necessary for your application.

1. Start the SOPAS ET configuration software user interface.
2. On the file card SCAN ASSISTANT click on CONFIGURATION; Select interface and configure (use CoLa-A for CoLa protocol).
3. Click on SCAN DEVICES, select device from SUITABLE DEVICE TYPES.
4. From the OPTIONS menu select the LOGIN DEVICE command and log in to the system using the password “client” as AUTHORIZED CLIENT.

ATTENTION

Do not switch off the voltage supply during configuration!

Switching off the voltage supply during configuration causes all parameters already configured to be lost.

5. Configure the LMS400 for the required application with the aid of the parameters in SOPAS ET.

Help for the program user interface as well as for the different options can be found in SOPAS ET.

7.1.2 Connection and test measurement

Use the graphic scan view in SOPAS ET to verify the measured values generated and to verify the measurement area online.

1. In the LMS400-XX00 project tree, choose MONITOR, SCAN VIEW.
2. In order to start the measurement, click on PLAY.
3. Compare the measurement line with the desired result.

Notes

- The SCAN VIEW in the MONITOR is dependent on the available computing power of the PC and is **not** output in real-time. For this reason not all measured values are displayed. The same limitation also applies when saving measured values displayed in a file.
- The monitor displays the measured values **unfiltered**, i.e. the action of filters can **not** be checked with the aid of the monitor.

4. After completing the test measurement successfully, save the configuration permanently to the LMS400: Menu LMS400_XX00, PARAMETER, SAVE PERMANENT.

7.2 Configuration and adjustment for Level Control

For the configuration and adjustment of the LMS400 for the Level Control application you require:

- plumb line
- carpenter's square
- measuring tape (up to 3000 mm resp. 118.11 in)
- felt-tip pen (the colour should be quite different from that of the transporting surface)
- tool set
- white adhesive tape for black transporting surfaces
- cuboid body

How to prepare the adjustment:

➢ Switch off the laser control of the LMS400.
PROJECT TREE, LMS400-XX00, PARAMETER, DIGITAL INPUTS/TRIGGER, area LASER CONTROL

This switches the laser of the LMS400 permanently on; the scan line of the LMS400 is now visible and the LMS400 can be adjusted.

7.2.1 Adjustment of the angle γ

➢ Adjust the mounting kit such that the middle of the sensor is exactly over the middle of the conveyor system. For this purpose, a plumb line can be attached to the adjusting screw used to adjust angle γ.

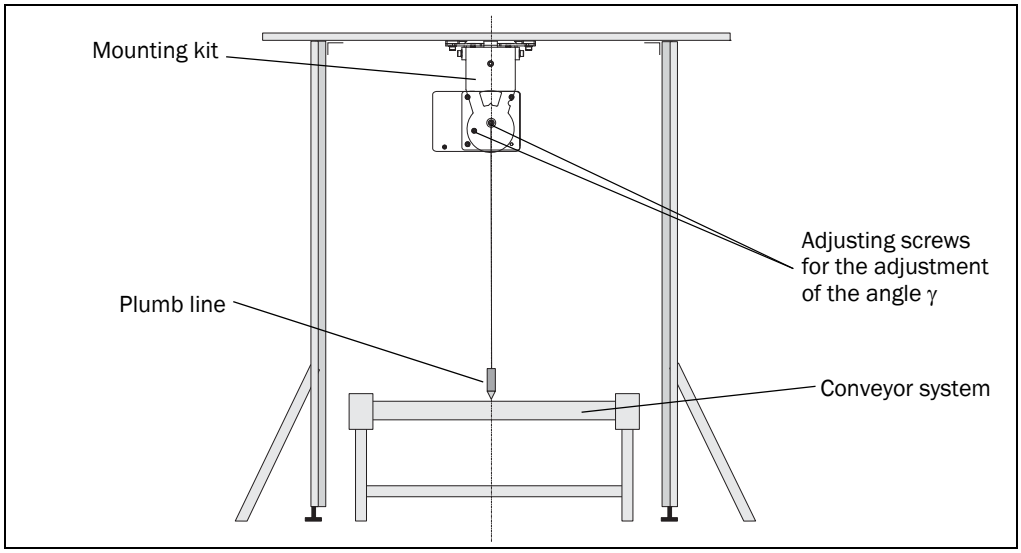


Fig. 34: Adjustment above a conveyor system

➢ Adjust the angle γ to 0°.

7.2.2 Adjustment of the angle β

1. Place a cubic object on the outside edge of the conveyor system. The scan line must be visible on the side of the cube.
2. Adjust the LMS400 with the help of the adjusting screw ① so that the scan line is perpendicular to the conveyor system.

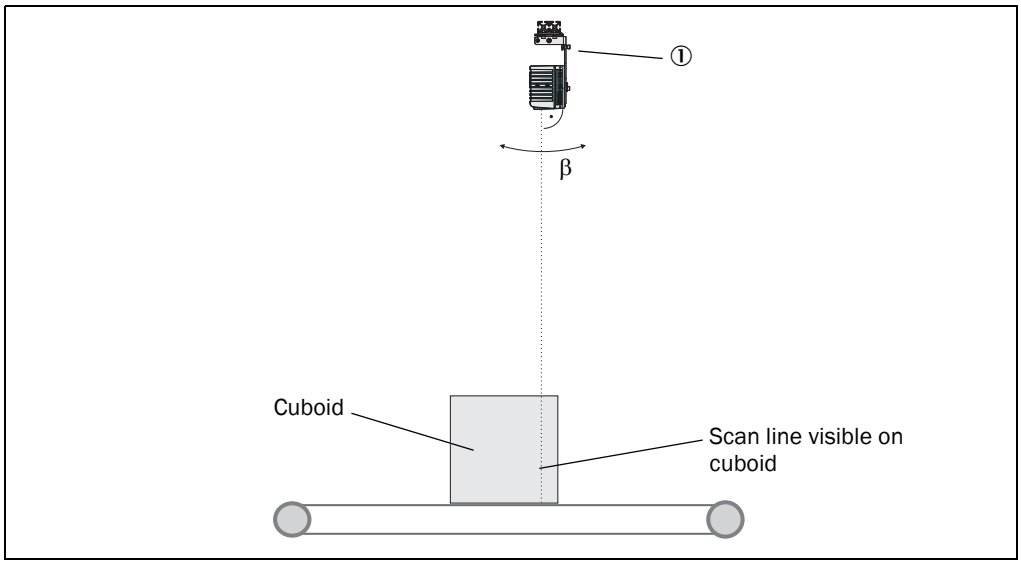


Fig. 35: Adjustment of the angle β

7.2.3 Adjustment of the angle α

- Using a felt-tip pen draw a line (direction vector) in the transporting direction. For this purpose start the conveyor system and hold the felt-tip pen on the moving surface.
- Align the scan line of the LMS400 at a right angle with the line on the conveyor system. A carpenter's square can be placed along the line on the conveyor system for this purpose.

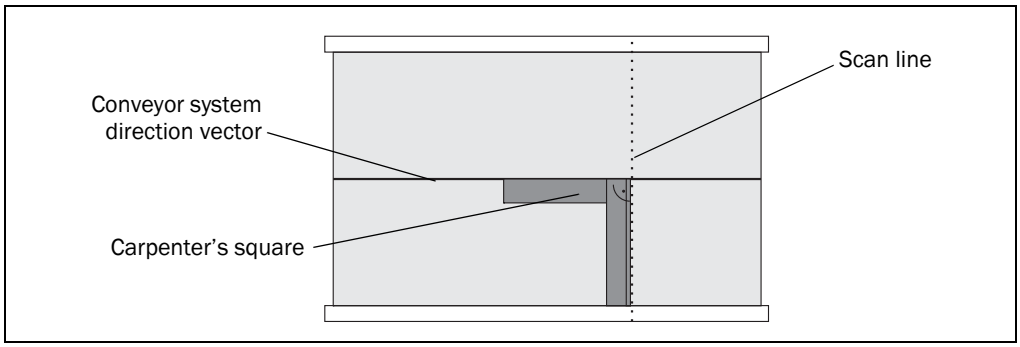


Fig. 36: Adjustment of the angle α

7.2.4 Determining the coordinates using the installation setup

There is an installation setup for the configuration of the position of the LMS400. There you will find detailed step-by-step instructions on commissioning.

Note To be able to use the installation setup, the LMS400 must be able to scan the transporting surface. Apply, for example, a piece of white tape along the scan line on a black surface.

7.2.5 Defining the angle γ, the y and the z coordinates

- Start the assistant on the LMS400_XX00 menu, COMMISSIONING, START INSTALLATION HELP.
- Enter the GAMMA, Y COORDINATE AND Z COORDINATE parameters (see *Fig. 37*). A rough entry is sufficient. The assistant will determine the exact parameters during the subsequent steps.

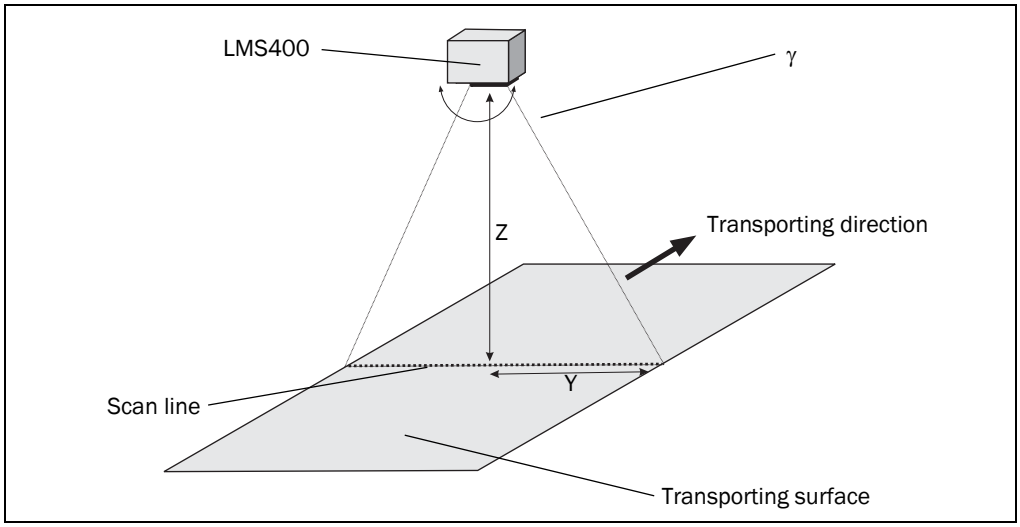


Fig. 37: Angle γ and y coordinate and z coordinate

7.2.6 Defining the transporting surface

In the next step the assistant scans the surrounding contour and displays the scan line seen. However, the assistant cannot yet identify which part of the scan line corresponds to the transporting surface.

You therefore define the position of the transporting surface by marking it.

➢ Position two marks using the right mouse button (M1 and M2) (see *Fig. 38* ①). Ensure the marks are as close as possible to the edges of the surface of the conveyor but not on the side limits of the conveyor system.

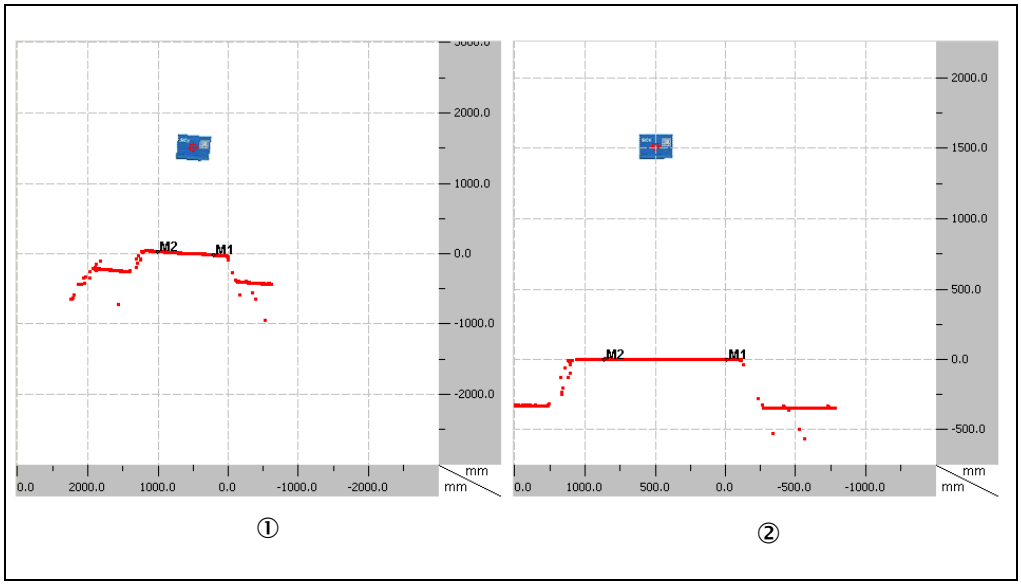


Fig. 38: Example of a scanned conveyor system

➢ In the assistant, click on CALCULATE.
The angle γ and the z coordinate are calculated and the scan line drawn horizontally in the diagram (see *Fig. 38* ②).

7.2.7 Defining the y coordinate

➢ Place the test object on end on the left side of the transporting surface.

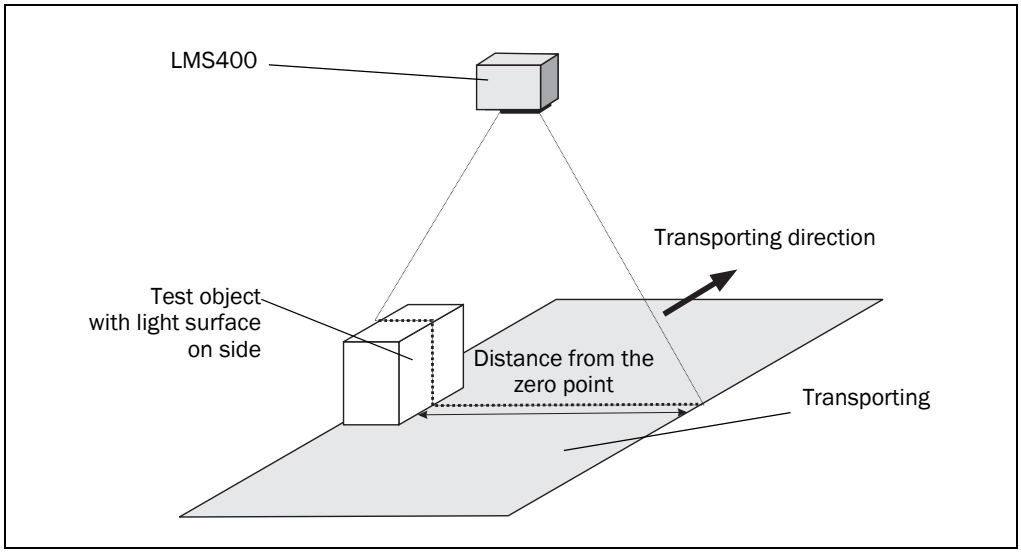


Fig. 39: Entry of the distance from the zero point

In the next step in the assistant enter the DISTANCE FROM THE ZERO POINT, the WIDTH OF OBJECT and the HEIGHT OF OBJECT.

Note The test object is 200 mm (7.87 in) wide and 300 mm (11.81 in) high. You must measure the distance from the zero point.

The assistant scans the surrounding contour and displays the scan line seen. However, the assistant cannot identify which part of the scan line corresponds to the test object.

➢ Define on the scan line which part of the surrounding contour on the right side corresponds to the object. For this purpose position two marks using the right mouse button (see *Fig. 40*, part 1).

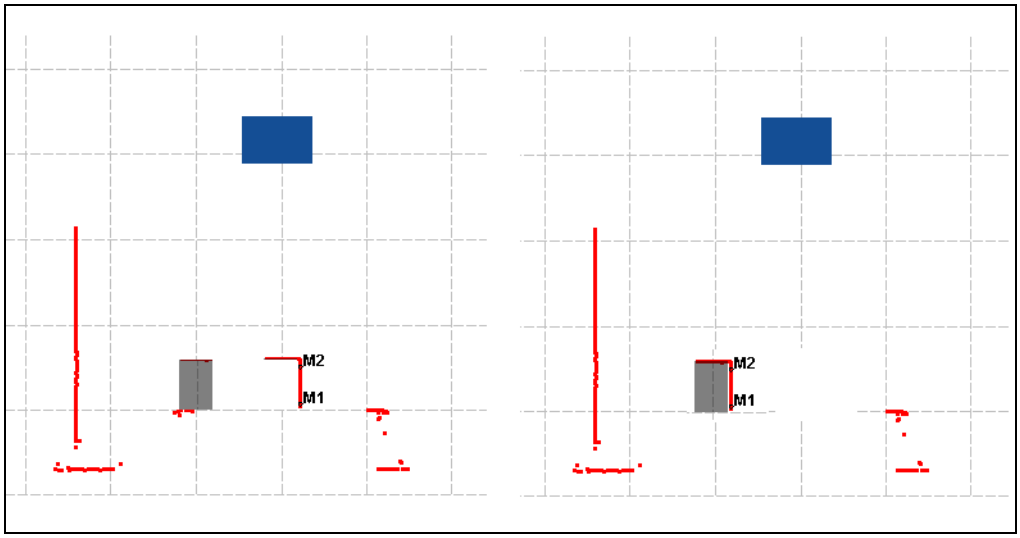


Fig. 40: Scanned test object

➢ In the assistant, click on CALCULATE.
The assistant now wraps the scan line around the test object (see *Fig. 40*, part 2) and calculates the distance from the LMS400 to the zero point and therefore its y coordinate.

7.2.8 Applying the parameters

Finally, the assistant displays the new parameters. If you apply the new parameters, they will be saved in the device.

Note The values are not yet displayed on the SOPAS ET user interface in PARAMETER, POSITION. The values are only displayed after the data have been uploaded from the device.

COMMUNICATION menu, command UPLOAD ALL PARAMETERS FROM DEVICE

8 Maintenance

8.1 Maintenance during operation

The LMS400 laser measurement sensor is maintenance-free apart from the maintenance measures listed below. No maintenance is necessary to ensure the retention of laser class 2.

Recommendation To preserve the full optical power of the LMS400, the front screen and any additional front screen should be regularly checked for contamination. This applies particularly in harsh operating environments (dust, powder, moisture, finger marks).

WARNING

Damage to the eye from laser radiation!

The LMS400 uses a red laser of class 2. On extended beam exposure, the retina in the eye may be damaged.

The entire front screen serves as the laser output aperture.

Warning — inappropriate use of the LMS400 can result in hazardous exposure to radiation and the laser class may be exceeded.

- Never look directly into the beam (similar to sunlight).
- Do not point the device laser beam at people.
- During mounting and adjustment of the LMS400, pay attention to possible reflections of the laser beam on reflective surfaces.
- Do not open the housing. (Opening the housing does not interrupt the power to the laser diode during the read cycle.)
- Observe the latest valid version of the laser safety regulations.

ATTENTION

Front screen damaged!

The front screen is made of glass. The optical power is reduced by scratches and smearing on the front screen.

- Do not use aggressive detergents.
- Do not use abrasive cleaning agents.
- Avoid scratching and scouring movements on the front screen.

Note Static charges cause dust particles to be attracted to the front screen. You can prevent this effect by using the antistatic plastic cleaner (SICK Part No. 5600006) and the SICK lens cloth (Part No. 4003353).

How to clean the front screen:

- Use a clean and soft brush to remove dust from the front screen.
- Then wipe the front screen with a clean and damp cloth.

8.2 Disposal

After de-commissioning, dispose of unusable or irreparable devices in an environmentally correct manner:

1. Observe national waste disposal regulations.
2. Dismantle the housing of the LMS400.
3. Remove electronics assemblies.
4. Send chassis and cover for die-cast aluminium recycling.
5. Dispose of all electronic assemblies as hazardous waste.

8.3 Replacement of a sensor

If the sensor has to be replaced, proceed as follows:

1. Switch off the voltage supply for the LMS400 and undo the connection.
2. Remove the connection cables from the LMS400.
3. Undo the three M6 screws (see *Fig. 28 on page 55*) and replace the LMS400.
4. Mounting the replacement device (see chapter *4 “Mounting” on page 54*).
5. Configuring the replacement device (see chapter *7 “Configuration (parameterisation)” on page 64*).

9 Troubleshooting

This chapter describes how to identify and rectify errors and malfunctions during the operation of the LMS400 laser measurement sensor.

9.1 In the event of faults or errors

ATTENTION

Cease operation if the cause of the malfunction has not been clearly identified!

Stop the machine/system if you cannot clearly identify or allocate the error and if you cannot safely rectify the malfunction.

9.2 Error on beam generation

The LMS400 automatically monitors the generation of the beam and automatically shuts down the laser diode in case of irregularities. In this case:

- The LED “Device Ready” turns red.
- The scanner transmits no more measured values.

To cancel error status:

➢ Switch the LMS400 off and back on again.

➢ If the error is still present when the device is switched on again, check the device status using SOPAS ET (see *9.3.1*). If errors are listed there, please contact SICK service.

9.3 Detailed error analysis

The LMS400 outputs occurring errors in various ways. Errors are output in stages and always permit detailed analysis:

- Communication errors can occur on the transfer of telegrams to the LMS400. The LMS400 then returns an error code.
- In case of status errors occurring during a scan, error codes are written to a status log. The sensor status is then set to 1 in the measured value telegram (see *11.3.1 on page 83*), so that your application can react appropriately.

9.3.1 Querying status log

Notes

- The status log is retained also after the device is switched off and on again.
- The LMS400 differentiates between four error types: “Information”, “Warning”, “Error” and “Serious error” (see *11.13 “Error codes” on page 131*). For each error type, the system saves only the last five occurrences.

Displaying log with the aid of SOPAS ET

You can display this logfile using SOPAS ET:

➢ Connect SOPAS ET to the device.

➢ Open the project tree LMS400-XX00, SERVICE, SYSTEM STATUS, area SYSTEM STATUS.

Reading status log with the aid of telegrams

You can also read the status log with the aid of telegrams. One telegram is available for each of the four different types of error (see *11.12 “Reference status log telegrams” on page 128*).

1. The terminal sends a telegram, comprising the command **sRN** (SOPAS read by name) and the telegram name.
2. The LMS400 sends a reply comprising **sRA** (SOPAS write answer) the telegram name, an error code (00000000 = no error) for the information, warnings, errors or serious errors.

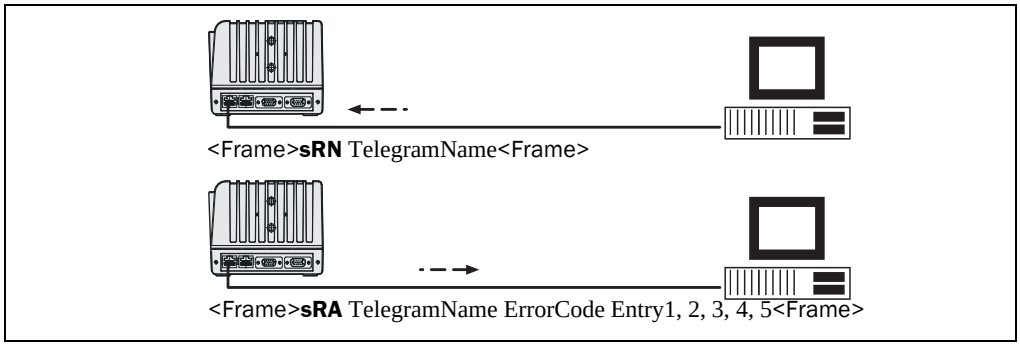


Fig. 41: Use of variables for configuration

10 Technical specifications

10.1 Data sheet LMS400 laser measurement sensor

Note The information in the data sheet relate to the measured value quality 7 (see *"Measured value quality" on page 28*). If the measured value quality determined from the parameters is <7, the sensor is no longer compliant with the values given in the technical data. If the measured value quality is ≥7, the technical data apply.

Type	**LMS400-1000**	**LMS400-2000**
Version	1 scanner solution	
Detectable object shape	Almost any, for restrictions see section *3.3.2 "Object specifications" on page 20*	
Useful field of view	Max. 70°	
Working area	0.7 to 3 m (2.3 to 9.8 ft)	
Laser output aperture	On front	
Laser diode (wavelength)	Visible light (λ = 650 nm ... 670 nm)	
Laser power	Max. 7.5 mW	Max. 10 mW
Laser class of the device	2	
Scanning frequency [1)]	230 Hz ... 500 Hz	300 Hz ... 500 Hz
Resonse time	≥ 2 ms	
Angular resolution	Choosable from 0.1333 to 1°	
Angular error	±0.1°	
Typical measuring error [2)] Systematic measuring error Statistical measuring error (1σ)	 ±4 mm (±0.16 in) 3 to 10 mm (0.12 to 0.39 in) (depending on remission and distance, see section *3.6.1 "Measuring accuracy of the distance measurement" on page 26*)	
Object remission	6.5 % to 200%	4.5 % to 100%
External light tolerance	2,000 lx	
Optical indicators	6 LEDs	
Switching inputs	4 x digital, encoder inputs IN2 and IN4: V_{in} = 24 V, F_{max}= 10 kHz	
Switching outputs [3)]	4 x digital, I_{out} = max. 400 mA 1 x analog, I_{out} = 4 mA ... 20 mA	
Aux interface	RS-232	
Host interfaces	1 x RS-232, RS-422 (data output format can be adjusted) 1 x Ethernet (10Base-T, half duplex)	
Electrical connections	2 × 8-pin RJ-45 female connector 1 × 15-pin D-Sub female connector 1 × 15-pin D-Sub male connector	
Supply voltage/power consumption	24 V DC ± 15%/max. 25 W	
Housing	Aluminium die-cast (on the exterior no materials that make usage of silicone)	

Tab. 24: Data sheet LMS400 laser measurement sensor

Type	**LMS400-1000**	**LMS400-2000**
Housing colour	Light blue (RAL 5012)	
Material of front screen	Float glass LMS400-1000S02: Polycarbonate [4)]	Float glass
EMC test	As per EN 61000-6-2:2001, EN 61000-6-4:2001	
Vibration/shock test	As per EN 60068-2-6, -27, -29, -64	
Electrical safety	As per EN 61010-1-3:2001	
Protection class	III, as per EN 61040-3:2002	
Enclosure rating	IP 20 (as per EN 60529-10-1991); with plug cover IP 65	
Weight	Approx. 2.3 kg (5.1 lb)	
Ambient temperature	Operation: 0 °C ... +40 °C (32 °F ... +104 °F) Storage: –20 °C ... +70 °C (–4 °F ... +158 °F)	
Air humidity	Max. 90%, non-condensing	

Tab. 24: Data sheet LMS400 laser measurement sensor

1) The frequency range in firmware V2.51 has been increased due to the specifications of the currently valid laser standard. Up to V2.50 the LMS400 operates with 180 Hz ... 500 Hz (LMS400-1000) and 270 Hz ... 500Hz (LMS400-2000). A firmware update is not required for these devices.
2) The information applies with the following boundary conditions:
   Room temperature 20 °C
   The LMS400 h as been switched on for at least two hours.
   Measuring distance, operating ambient temperature and object remission must be inside the specified range.
   The intensity of light from external sources is ≤2 kLux.
3) If a load is not connected to the outputs, the outputs will exhibit tristate behavior. Normal switching behavior is achieved on the termination with 10 kOhm.
4) Used in order to prevent fragments of glass being produced in the event of mechanical damage (in food applications, for instance). Plastic panes have a higher optical attenuation value than glass panes. This may result in a reduction in the measurement accuracy and detection capacity of the device as compared with the standard variant.

10.2 Dimensional drawings

10.2.1 Dimensional drawing LMS400 laser measurement sensor

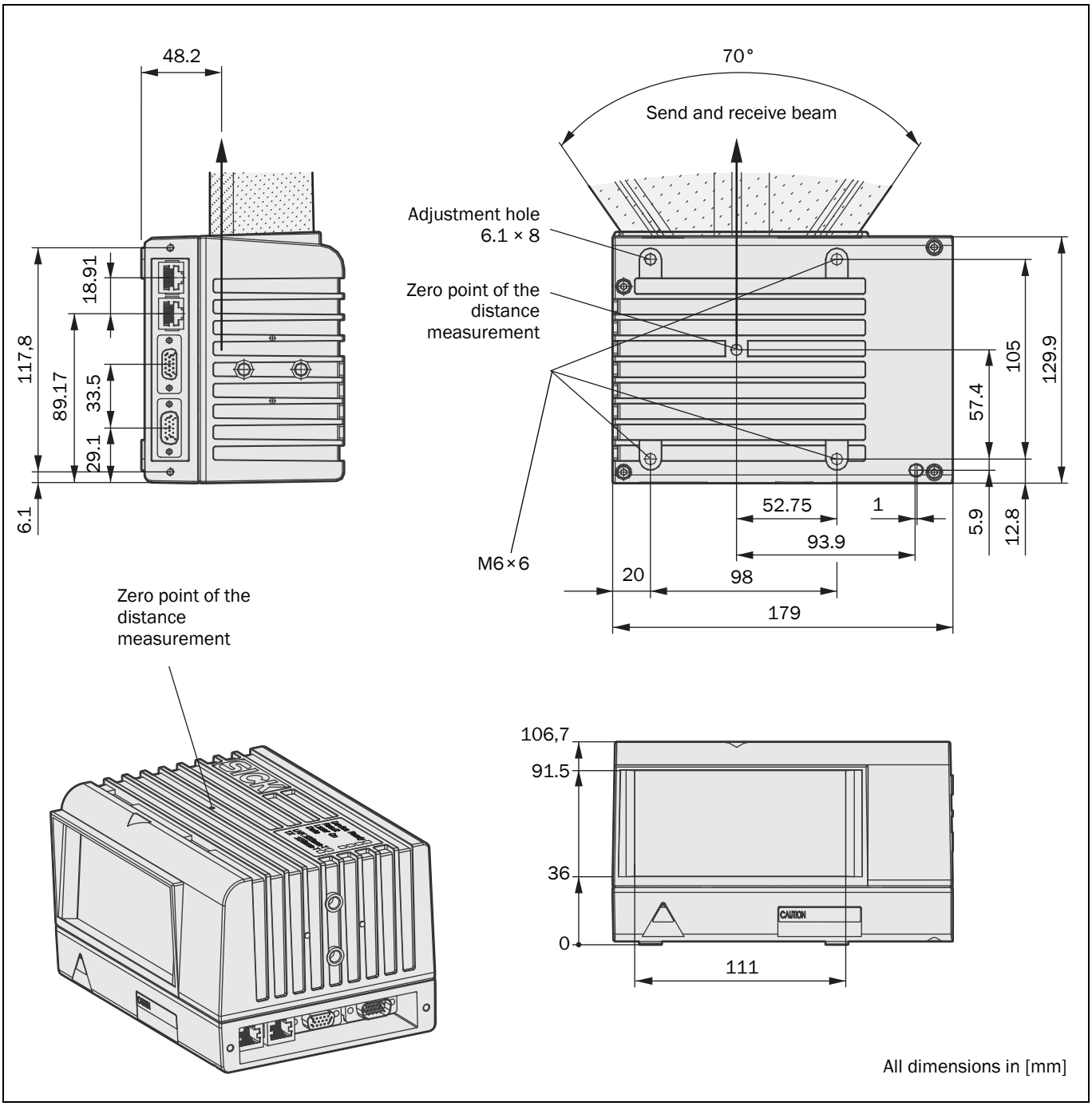


Fig. 42: Dimensional drawing LMS400

Other sensors on request.

10.2.2 Dimensional drawing mounting kit No. 2030421 for LMS400

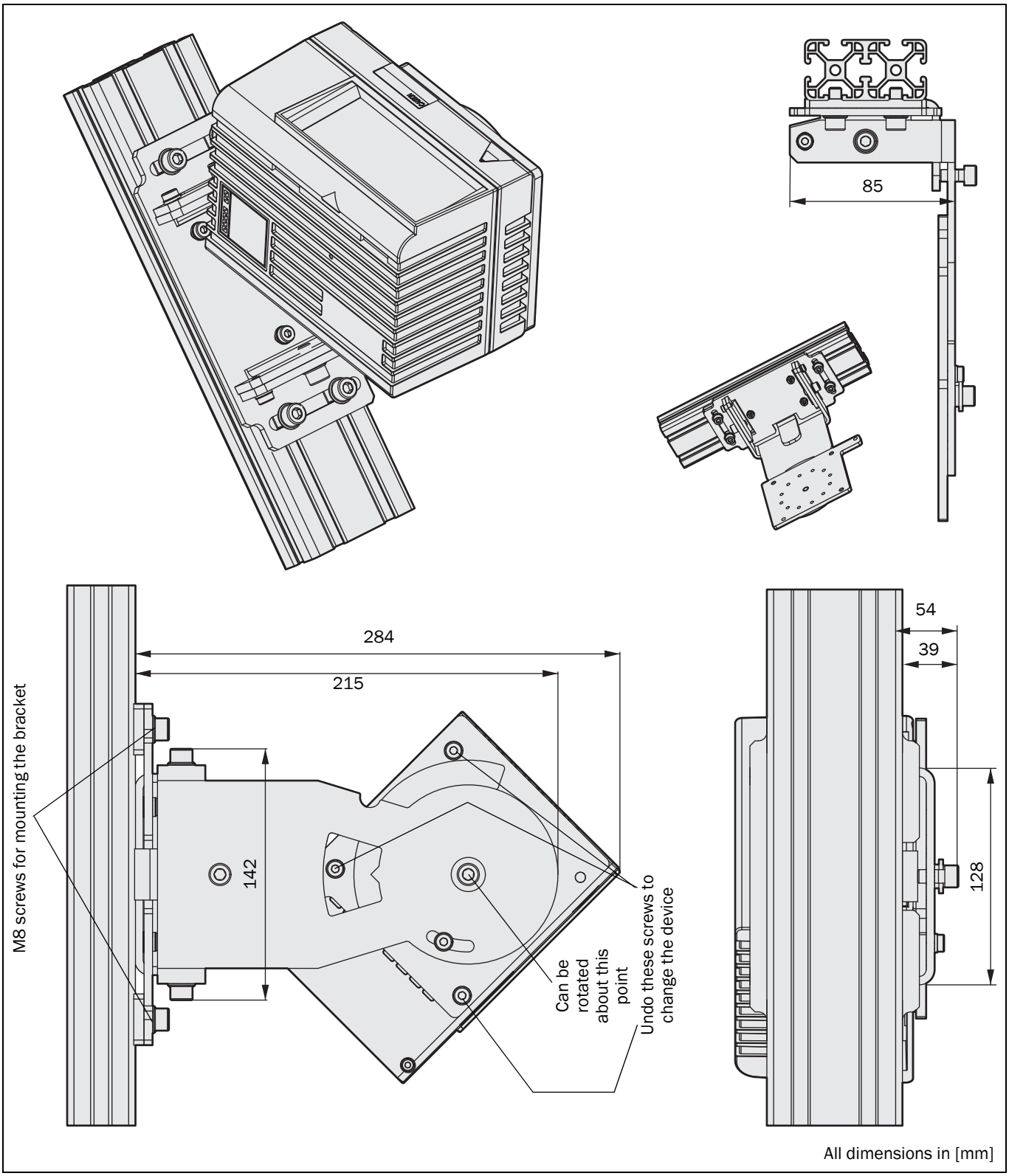


Fig. 43: Dimensional drawing mounting kit for LMS400 laser measurement sensor

11 Annex

11.1 Overview of the annexes

The annex contains the following supplementary information:

- Overview of the telegrams
- Reference measured value telegrams
- Reference configuration telegrams for the basic measurement parameters
- Reference general configuration telegrams
- Configuration telegrams for master/slave operation
- Reference configuration telegrams for filter setting
- Reference configuration telegrams for triggering
- Configuration telegrams for the host interface
- Configuration telegrams for the Ethernet interface
- Reference status log telegrams
- Reference measured value telegrams
- Error codes
- Ordering information
- Hints on EU Declaration of Conformity
- Glossary

11.2 Overview of the telegrams

The following table provides an overview of the telegrams that the LMS400 can use to communicate. In the User level column you will find the lowest user level required to use the telegram:

0 = No user level
1 = Machine operator
2 = Maintenance personnel
3 = Authorized client

Telegram name	Telegram type	User level	See
Telegrams for querying measured values			
Continuous measured value output	Procedure	0	*11.3.1 on page 83*
Triggered measured value output	Procedure	0	*11.3.2 on page 86*
Output defined number of scans	Variable	0	*11.3.3 on page 87*
Stop measurement	Procedure	0	*11.3.4 on page 88*
Telegrams for setting the basic parameters for the measurement			
Configuration of the scanning frequency and angular resolution	Procedure	0	*11.4.1 on page 89*
Configuration by defining the scanning frequency	Procedure	0	*11.4.2 on page 91*
Configuration by defining the angular resolution	Procedure	0	*11.4.3 on page 93*
Enable extended RIS detectivity	Variable	0	*11.4.4 on page 94*
Telegrams for general device settings			
Select user level	Procedure	0	*11.5.1 on page 95*
Query user level	Procedure	0	*11.5.2 on page 97*
Set password	Procedure	2 to 3	*11.5.3 on page 98*
Set device name	Variable	2	*11.5.4 on page 99*
Save parameters permanently	Procedure	3	*11.5.5 on page 100*
Terminate configuration	Procedure	0	*11.5.6 on page 101*
Reset device	Procedure	3	*11.5.7 on page 102*
Place device in delivery status	Procedure	3	*11.5.8 on page 103*
Telegrams for master/slave operation			
Select synchronisation type	Variable	3	*11.6.1 on page 104*
Define phase	Variable	3	*11.6.2 on page 104*
Telegrams to set filters			
Enable filter	Variable	2	*11.7.1 on page 105*
Define median filter	Variable	2	*11.7.2 on page 106*
Define range filter	Variable	2	*11.7.3 on page 106*
Define mean filter	Variable	2	*11.7.4 on page 107*
Telegrams for triggering			
Time or distance controlled	Variable	4	*11.8.1 on page 108*
Define digital inputs	Variable	3	*11.8.2 on page 109*
Define the gate settings	Variable	3	*11.8.3 on page 110*
Enable laser control	Variable	3	*11.8.4 on page 111*
Select the encoder settings	Variable	3	*11.8.5 on page 112*

Telegram name	Telegram type	User level	See
Telegrams for the configuration of the outputs			
Enable outputs	Procedure	4	*11.9.1 on page 113*
Telegrams for the configuration of the host interface			
Select hardware setting	Variable	3	*11.10.1 on page 114*
Select baud rate	Variable	3	*11.10.2 on page 115*
Select number of stop bits	Variable	3	*11.10.3 on page 116*
Select data and parity bits	Variable	3	*11.10.4 on page 117*
Select block check byte setting	Variable	3	*11.10.5 on page 118*
Select receive start character	Variable	3	*11.10.6 on page 119*
Select receive stop character	Variable	3	*11.10.7 on page 120*
Select send start character	Variable	3	*11.10.8 on page 121*
Select send stop character	Variable	3	*11.10.9 on page 122*
Telegrams for the configuration of the Ethernet interface			
Define IP address	Variable	3	*11.11 on page 123*
Define gateway address	Variable	3	*11.11.2 on page 124*
Define subnet mask	Variable	3	*11.11.3 on page 124*
Define port for the TCP/IP communication	Variable	3	*11.11.4 on page 125*
Read MAC address	Variable	0	*11.11.5 on page 126*
Select format for the CoLa protocol	Variable	3	*11.11.6 on page 127*
Telegrams for querying status			
Query warnings	Variable	0	*11.12.1 on page 128*
Query errors	Variable	0	*11.12.2 on page 129*
Query serious errors	Variable	0	*11.12.3 on page 130*
Telegrams Level Control			
Start measured value output for Level Control	Variable	0	*11.14.1 on page 131*
Stop measured value output for Level Control	Variable	0	*11.14.2 on page 137*

Tab. 25: Overview of the measured value and configuration telegrams

Notation

The individual telegram sections are each to be separated by a space (ASCII code 32, hex 20). The LMS400 interprets the parameters transferred as follows:

- Parameters with a leading “+” or “–” are interpreted as a decimal value (ASCII notation).
- Parameters without a leading “+” or “–” are interpreted as a hexadecimal value (ASCII notation).
- The LMS400 interprets each parameter individually, i.e. the different notations can be mixed within a telegram.
- All the examples used in the following telegram lists refer to the Cola-A protocol.

Variable types

The variable types are given in the telegram syntax, the following variable types are possible:

Variable type	Length (byte)	Value range	Sign
bool_1	1	0 or 1	No
uint_8	1	0 ... 255	No
int_8	1	–128 ... 127	Yes
uint_16	2	0 ... 65535	No
int_16	2	–32768 ... 32767	Yes
uint_32	4	0 ... 4294967295	No
int_32	4	–2147483648 ... 2147483647	Yes
float_32	4	$\pm$~$10^{-44.85}$ ... $10^{38.53}$	Yes
string	Context-dependent	Note: Strings are terminated with non-zero characters	

Tab. 26: Variable types

Notes

- The information in the “Length” column of the table refers to the binary transfer of the numeric parameters.
- The information in the “Value range” column in the table refers to the value range mathematically possible for the variable type. The actual value ranges for the parameters may be different. You will find these in the telegram syntax that follows.

Syntax error

If the LMS400 detects an error in the syntax of a received telegram, it outputs an error telegram with an error code.

Telegram structure: **sFA** ErrorCode

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Syntax error or logical error	string	3	**sFA**
ErrorCode	Contains the error type (see *Tab. 27*)	uint_16	2	**FF00h** ... **FFFFh**

Telegram syntax 1: Syntax error or logical error

ErrorCode	Possible cause	Solution
FF79h	Unknown name	The procedure or parameter name used is unknown. Check for spelling mistakes.
FFC8h FFC9h	User level too low	A higher user level is required to access the procedure or parameter. Change to required user level (see *11.5.1 on page 95*).
FF??h	General syntax error	Check telegram syntax: Type of command, command, parameter number and parameter value range

Tab. 27: Syntax error or logical errors

11.3 Reference measured value telegrams

11.3.1 Continuous measured value output

Continuous measured value output is started using this telegram (see *Fig. 5 on page 22*). Distance and remission values, only distance values or only remission values can be queried from the LMS400. Additional, expanded information on the measured values as well as the state of the I/O is transferred in the resulting cyclic data (measured values).

Request

Telegram structure: **sMN mLRreqdata** Format

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS method by name)	string	3	**sMN**
Command	Data request	string	10	**mLRreqdata**
Format	Defines content and size of the measured values telegram	uint_16	2	**0020h** distance and remission **0021h** distance only **0022h** remission only

Telegram syntax 2: Request “Continuous measured value output”

Confirmation

Telegram structure: **sMA mLRreqdata**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Confirmation (SOPAS method acknowledge)	string	3	**sMA**
Command	Data request	string	10	**mLRreqdata**

Telegram syntax 3: Confirmation of the “Continuous measured value output” request

Answer

Telegram structure: **sAN mLRreqdata** ErrorCode

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sAN**
Command	Data request	string	10	**mLRreqdata**
ErrorCode	The command has been accepted if the error code 0 is returned.	uint_32	4	**00000000h** no error ... **FFFFFFFFh** ErrorCode

Telegram syntax 4: Answer to the “Continuous measured value output” request

Cyclic data (measured values)

Note The contents of this telegram are sent in Little Endian format. Every telegram contains a complete scan line with all measured values.

Telegram structure: Format DistanceScaling StartingAngle AngularStepWidth NumberMeasuredValues ScanningFrequency RemissionScaling RemissionStartValue RemissionEndValue Distance_1 RemissionValue_1 [Distance_2 RemissionValue_2 … Distance_n RemissionValue_n] DigitalInputs ReservedBytesA ReservedBytesB EncoderPosition ReservedBytesC ReservedBytesD ScanCounter TelegramCounter SystemCounter

Telegram part		**Description**	**Variable type**	**Length (byte)**	**Value range**
Definition block	Format	Defines content and size of the measured values telegram	uint_16	2	**0020h** distance and remission **0021h** distance only **0022h** remission only
	DistanceScaling	Scaling of the distance values. The distance values are to be multiplied by this factor.	uint_16	2	**1**
	StartingAngle	Information 1/10 000 degree	int_32	4	**550000** ... **1250000**
	AngularStepWidth	Information 1/10 000 degree	uint_16	2	**1000** ... **10000**
	NumberMeasuredValues	Number of measured values in telegram	uint_16	2	**0** ... **700**
	ScanningFrequency	Information in Hertz	uint_16	2	**100** ... **500**
	RemissionScaling	Scaling for the remission value. The remission values are to be multiplied by this factor.	uint_16	2	**2**
	RemissionStartValue	Lower or upper limit for the remission value without scaling. Information in percent	uint_16	2	**0**
	RemissionEndValue		uint_16	2	**255**
Measured values	Distance_1	Information in millimetres. The distance value is to be multiplied with the DistanceScaling. Output is dependent on the “Format” parameter.	uint_16	2	**0000h** invalid **01F4h** 500 mm ... **0BB8h** 3000 mm
	RemissionValue_1	The remission value is to be multiplied with RemissionScaling. Output is dependent on the “Format” parameter.	uint_8	1	**0** ... **254** remission **255** glare
	Distance_n	See above	uint_16	2	See above
	RemissionValue_n	See above	uint_8	1	See above

Telegram part		Description	Variable type	Length (byte)	Value range
I/O status	DigitalInputs	The least significant byte reflects the state of the digital inputs by bit. The least significant bit corresponds to input 1.	uint_16	2	**0000h** all inputs off **000Fh** all inputs on
	ReservedBytesA	Reserved	uint_16	2	**0000h**
	ReservedBytesB	Reserved	uint_16	2	**0000h**
	EncoderPosition	Information in ticks	uint_16	2	**0000h** ... **FFFFh**
	ReservedBytesC	Reserved	uint_16	2	**0000h**
Sensor status	ReservedBytesD	Reserved	uint_16	2	**0000h**
	ScanCounter	Counter, starting with the first scan after confirmation of the measured value telegram. When the upper limit is reached, the counter starts again at 0 (= 1. scan).	uint_16	2	**0000h** 0 **0FFFh** 4095
	TelegramCounter	Counter, starting at the first measured value telegram (cyclic data) after confirmation of the measured value telegram. When the upper limit is reached, the counter starts again at 0 (= 1. telegram).	uint_16	2	**0000h** 0 **FFFFh** 65535
	SystemCounter	Enables the relative time difference between two measured value telegrams to be calculated. Information in 1/327.68 µs. When the upper limit is reached, the counter starts again at 0.	uint_16	2	**0000h** 0 µs **FFFFh** 21.4745 s

Telegram syntax 5: Cyclic data (measured values) in the maximum telegram

Note If the measured value quality is <7, then you are operating the sensor outside the specification. On this subject please read section *3.6.4 "Measured value quality" on page 28*.

Example

Request: sMN mLRreqdata 0020

Confirmation: sMA mLRreqdata

Answer: sAN mLRreqdata 00000000

Cyclic data: (see *Tab. 5*)

11.3.2 Triggered measured value output

The measurement/output of data only starts when a trigger is present (see *3.5.2 "Control of the measurement process using a gate" on page 23*). The following types of triggering are possible:

- hardware triggering using digital inputs
- software triggering using telegram with defined number of scans (see *11.3.3*)

The measured values output ends automatically when the trigger is no longer present or the cyclic output is stopped using the "Stop measurement" command (see *11.3.4*).

Distance and remission values, only distance values or only remission values can be queried from the LMS400. Additional, expanded information on the measured values as well as the state of the I/O is transferred in the resulting cyclic data (measured values).

Request

Telegram structure: **sMN mLRreqtrigdata** Format

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS method by name)	string	3	**sMN**
Command	Data request	string	14	**mLRreqtrigdata**
Format	Defines content and size of the measured values telegram	uint_16	2	**0020h** distance and remission **0021h** distance only **0022h** remission only

Telegram syntax 6: Request "Triggered measured value output"

Confirmation

Telegram structure: **sMA mLRreqtrigdata**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Confirmation (SOPAS method acknowledge)	string	3	**sMA**
Command	Data request	string	14	**mLRreqtrigdata**

Telegram syntax 7: Confirmation of the "Triggered measured value output" request

Answer

Telegram structure: **sAN mLRreqtrigdata** ErrorCode

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sAN**
Command	Data request	string	14	**mLRreqtrigdata**
ErrorCode	The command has been accepted if the error code 0 is returned.	uint_32	4	**00000000h** no error ... **FFFFFFFFh** ErrorCode

Telegram syntax 8: Answer to "Triggered measured value output" request

Cyclic data (measured values)

See *"Cyclic data (measured values)" on page 84*.

Example

Request: sMN mLRreqtrigdata 0020

Confirmation: sMA mLRreqtrigdata

Answer: sAN mLRreqtrigdata 00000000

Cyclic data: (see *Tab. 5 on page 85*)

11.3.3 Output defined number of scans

If you have initiated triggered measured value output (see *11.3.2 on page 86*), this telegram results in the activation of measured data output for a specific number of scans. For this purpose the telegram is transferred with a parameter that defines how many measured value telegrams are to be output.

Note A prerequisite is that software trigger is selected in the trigger settings as the trigger source (see *11.8.3 on page 110*).

As an alternative to configuration with telegrams, you can also make the trigger settings in SOPAS. PROJECT TREE, LMS400-XX00, PARAMETER, DIGITAL INPUTS, area GATE SETTINGS

Request

Telegram structure: **sWN LRscnt** Number

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS write by name)	string	3	**sWN**
Command	Output defined number of scans	string	6	**LRscnt**
Number	Number of measured values that are to be output	uint_16	2	**0001h** 1 scan ... **000Ah** 10 scans

Telegram syntax 9: Request "Output defined number of scans"

Answer

Telegram structure: **sWA LRscnt**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sWA**
Command	Output defined number of scans	string	6	**LRscnt**

Telegram syntax 10: Answer to the "Output defined number of scans" request

Example

Request: sWN LRscnt 0001

Answer: sWA LRscnt

11.3.4 Stop measurement

Request

Telegram structure: **sMN mLRstopdata**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS method by name)	string	3	**sMN**
Command	Stop measurement	string	11	**mLRstopdata**

Telegram syntax 11: Request “Stop measurement”

Answer 1 (confirmation)

Telegram structure: **sMA mLRstopdata**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Confirmation (SOPAS method acknowledge)	string	3	**sMA**
Command	Stop measurement	string	11	**mLRstopdata**

Telegram syntax 12: Confirmation of the “Stop measurement” request

Answer 2 (result)

Telegram structure: **sAN mLRstopdata**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sAN**
Command	Stop measurement	string	11	**mLRstopdata**
ErrorCode	The command has been accepted if the error code 0 is returned.	uint_32	4	**00000000h** no error ... **FFFFFFFFh** ErrorCode

Telegram syntax 13: Answer to the “Stop measurement” request

Example

Request: sMN mLRstopdata

Confirmation: sMA mLRstopdata

Answer: sAN mLRstopdata 00000000

11.4 Reference configuration telegrams for the basic measurement parameters

There are three ways of defining the angular resolution and scanning frequency:

- configuration of scanning frequency and angular resolution
- configuration by defining the scanning frequency
- configuration by defining the angular resolution

ATTENTION

The configuration of the scanner can be different from the parameters transmitted!

The scanner does not apply the basic parameters of scanning frequency and angular resolution exactly, the values must be technically optimised. For this reason, in further calculations during data analysis use the parameters contained in the related response from the scanner.

11.4.1 Configuration of scanning frequency and angular resolution

Configures the scanner by means of the exact definition of scanning frequency and angular resolution. The LMS400 calculates the technically possible values based on these parameters for supplies the parameters actually used in response 2 (result).

Note

- The required user level is “Authorized client” (see *11.5.1 on page 95*).
- Only transfer suitable scanning frequencies and angular resolutions. You can find the right combination in SOPAS ET with the aid of the scanning frequency wizard.

  PROJECT TREE, LMS400-XX00, PARAMETER, BASIC PARAMETER, CURRENT DEVICE PARAMETERS

Request

Telegram structure: **sMN mSCsetscanconfig** ScanningFrequency AngularResolution BeginMeasurementArea LengthMeasurementArea

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS method by name)	string	3	**sMN**
Command	Configuration of scanning frequency and angular resolution	string	16	**mSCsetscanconfig**
ScanningFrequency	Is rounded by the LMS400 to an integer multiple of 10. Information in Hertz	float_32	4	**+360** … **+500**
AngularResolution	Is rounded to a valid value. Information in degrees	float_32	4	**+0.1** … **+1**
BeginMeasurementArea	Information in degrees	float_32	4	**+55.00** … **+124.00**
LengthMeasurementArea	Information in degrees	float_32	4	**+0.00** … **+70.00**

Telegram syntax 14: Request “Configuration of scanning frequency and angular resolution”

Confirmation

Telegram structure: **sMA mSCsetscanconfig**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Confirmation (SOPAS method acknowledge)	string	3	**sMA**
Command	Configuration of scanning frequency and angular resolution	string	16	**mSCsetscanconfig**

Telegram syntax 15: Confirmation of the “Configuration of scanning frequency and angular resolution” request

Answer

Telegram structure: **sAN mSCsetscanconfig** ErrorCode ScanningFrequency AngularResolution MeasuredValueQuality

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sAN**
Command	Configuration of scanning frequency and angular resolution	string	16	**mSCsetscanconfig**
ErrorCode	The command has been accepted if the error code 0 is returned.	uint_32	4	**00000000h** no error ... **FFFFFFFFh**
ScanningFrequency	Scanning frequency calculated by the LMS400. Information in Hertz. Displayed as hex value	float_32	4	**42C80000h** 100.0 ... **43FA0000h** 500.0
AngularResolution	Angular resolution calculated by the LMS400. Information in degrees. Displayed as hex value	float_32	4	**3DCCCCCDh** 0.1 ... **3F800000h** 1.0
MeasuredValueQuality	Measured value quality calculated by the LMS400. Displayed as hex value	uint_8	1	**05h** 5 ... **0Ah** 10

Telegram syntax 16: Answer to the “Configuration of scanning frequency and angular resolution” request

Note If the measured value quality is <7, then you are operating the sensor outside the specification. On this subject please read section *3.6.4 “Measured value quality” on page 28*.

Example

Request: sMN mSCsetscanconfig +300 +0.25 +55.0 +70.0

Confirmation: sMA mSCsetscanconfig

Answer: sAN mSCsetscanconfig 00000000 4395C78F 3E800000 07

11.4.2 Configuration by defining the scanning frequency

Configures the scanner by means of the exact definition of a scanning frequency and rough definition of the angular resolution. The LMS400 calculates the technically possible values based on these parameters for supplies the parameters actually used in response 2 (result).

Note It is not necessary to log in to the device **first**, as the login is part of this telegram. To enable the parameters to be applied by the LMS400, as a minimum login as “Maintenance personnel” is necessary.

Request

Telegram structure: **sMN mSCconfigbyfreq** UserLevel Password ScanningFrequency RoughSelectionAngularResolution BeginMeasurementArea LengthMeasurementArea

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS method by name)	string	3	**sMN**
Command	Configuration via scanning frequency	string	15	**mSCconfigbyfreq**
UserLevel	A valid user level must be included in the transmission. Otherwise the LMS400 rejects the command.	uint_8	1	**02h** maintenance personnel **03h** authorized client **04h** service
Password	Hash value of the password (see *11.5.1 on page 95*)	uint_32	4	**00000000h** ... **FFFFFFFFh**
ScanningFrequency	Is rounded by the LMS400 to an integer multiple of 10. Information in Hertz	float_32	4	**+360** ... **+500**
RoughSelectionAngularResolution	Corresponds to the possible selections available in SOPAS	uint_8	1	**00h** coarse **01h** medium **02h** fine
BeginMeasurementArea	Information in degrees	float_32	4	**+55.00** ... **+124.00**
LengthMeasurementArea	Information in degrees	float_32	4	**+0.00** ... **+70.00**

Telegram syntax 17: Request “Configuration by defining the scanning frequency”

Confirmation

Telegram structure: **sMA mSCconfigbyfreq**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Confirmation (SOPAS method acknowledge)	string	3	**sMA**
Command	Configuration via scanning frequency	string	15	**mSCconfigbyfreq**

Telegram syntax 18: Confirmation of the “Configuration by defining the scanning frequency” request

Answer

Telegram structure: **sAN mSCconfigbyfreq** ErrorCode ScanningFrequency AngularResolution MeasuredValueQuality

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sAN**
Command	Configuration via scanning frequency	string	15	**mSCconfigbyfreq**
ErrorCode	The command has been accepted if the error code 0 is returned.	uint_32	4	**00000000h** no error ... **FFFFFFFFh** ErrorCode
ScanningFrequency	Scanning frequency calculated by the LMS400. Information in Hertz. Displayed as hex value	float_32	4	**42C80000h** 100.0 ... **43FA0000h** 500.0
AngularResolution	Angular resolution calculated by the LMS400. Information in degrees. Displayed as hex value	float_32	4	**3DCCCCCDh** 0.1 ... **3F800000h** 1.0
MeasuredValueQuality	Measured value quality calculated by the LMS400. Displayed as hex value	uint_8	1	**05h** 5 ... **0Ah** 10

Telegram syntax 19: Answer to the “Configuration by defining the scanning frequency” request

Note If the measured value quality is <7, then you are operating the sensor outside the specification. On this subject please read section *3.6.4 “Measured value quality” on page 28*.

Example

Request: sMN mSCconfigbyfreq 03 B18244B6 +300 +0 +55.0 +70.0

Confirmation: sMA mSCconfigbyfreq

Answer: sAN mSCconfigbyfreq 00000000 4395C78F 3ECCCCCD 08

11.4.3 Configuration by defining the angular resolution

Configures the scanner by means of the exact definition of an angular resolution and rough definition of the scanning frequency. The LMS400 calculates the technically possible values based on these parameters for supplies the parameters actually used in response 2 (result).

Note It is not necessary to log in to the device **first**, as the login is part of this telegram. To enable the parameters to be applied by the LMS400, as a minimum login as “Maintenance personnel” is necessary.

Request

Telegram structure: **sMN mSCconfigbyang** UserLevel Password AngularResolution RoughSelectionScanningFrequency BeginMeasurementArea LengthMeasurementArea

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS method by name)	string	3	**sMN**
Command	Configuration via angular resolution	string	14	**mSCconfigbyang**
UserLevel	A valid user level must be included in the transmission. Otherwise the LMS400 rejects the command.	uint_8	1	**02h** maintenance personnel **03h** authorized client **04h** service
Password	Hash value of the password (see *11.5.1 on page 95*)	uint_32	4	**00000000h** ... **FFFFFFFFh**
AngularResolution	Is rounded to a valid value. Information in degrees	float_32	4	**+0.1** ... **+1**
RoughSelectionScanningFrequency	Corresponds to the possible selections available in SOPAS	uint_8	1	**00h** fast **01h** medium **02h** slow
BeginMeasurementArea	Information in degrees	float_32	4	**+55.00** ... **+124.00**
LengthMeasurementArea	Information in degrees	float_32	4	**+0.00** ... **+70.00**

Telegram syntax 20: Request “Configuration by defining the angular resolution”

Confirmation

Telegram structure: **sMA mSCconfigbyang**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Confirmation (SOPAS method acknowledge)	string	3	**sMA**
Command	Configuration via angular resolution	string	14	**mSCconfigbyang**

Telegram syntax 21: Confirmation of the “Configuration by defining the angular resolution” request

Answer

Telegram structure: **sAN mSCconfigbyang** ErrorCode ScanningFrequency AngularResolution MeasuredValueQuality

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sAN**
Command	Configuration via angular resolution	string	14	**mSCconfigbyang**
ErrorCode	The command has been accepted if the error code 0 is returned.	uint_32	4	**00000000h** no error ... **FFFFFFFFh**
ScanningFrequency	Scanning frequency calculated by the LMS400. Information in Hertz. Displayed as hex value	float_32	4	**42C80000h** 100.0 ... **43FA0000h** 500.0
AngularResolution	Angular resolution calculated by the LMS400. Information in degrees. Displayed as hex value	float_32	4	**3DCCCCCDh** 0.1 ... **3F800000h** 1.0
MeasuredValueQuality	Measured value quality calculated by the LMS400. Displayed as hex value	uint_8	1	**05h** 5 ... **0Ah** 10

Telegram syntax 22: Answer to the “Configuration by defining the angular resolution” request

Note If the measured value quality is <7, then you are operating the sensor outside the specification. On this subject please read section *3.6.4 “Measured value quality” on page 28*.

Example

Request: sMN mSCconfigbyang 03 B18244B6 +0.25 01 +55.0 +70.0

Confirmation: sMA mSCconfigbyang

Answer: sAN mSCconfigbyang 00000000 4340FF1D 3E800000 08

11.4.4 Enable extended RIS detectivity

If you want to measure objects with remission values <10%, you can extend the so-called Remission Information System (RIS) on the LMS400 (see *“Measurement area expansion” on page 27*).

Request

Telegram structure: **sWN MDblex** ExtendedRISDetectivity

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS write by name)	string	3	**sWN**
Command	Enable extended RIS detectivity	string	6	**MDblex**
ExtendedRISDetectivity	Extended RIS detectivity is active	bool_1	1	**00h** false **01h** true

Telegram syntax 23: Request “Enable extended RIS detectivity”

Answer

Telegram structure: **sWA MDblex**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sWA**
Command	Enable extended RIS detectivity	string	6	**MDblex**

Telegram syntax 24: Answer to the “Enable extended RIS detectivity” request

Example

Request: sWN MDblex 01

Answer: sWA MDblex

11.5 Reference general configuration telegrams

11.5.1 Select user level

By means of the selection of a user level and transfer of the corresponding password, permits further configuration. The LMS400 expects the password in the telegram in coded form (hash value).

How to determine the hash value for the password:

- Open or create a SOPAS ET project which contains the LMS400.
- Mark in the project tree the LMS400 or a lower level branch.
- Login to the device using OPTIONS, LOGIN DEVICE with the required user level.
- In the menu LMS400_XX00 open the command PASSWORD, CALCULATE HASH VALUE.
- Enter the password in the PLAIN TEXT field and click on START CALCULATION.
- Use the hash value determined **without** the prefix “0x”.

Note Factory setting for devices with software version up to 1.13 is the hash value B18244B6

Request

Telegram structure: **sMN SetAccessMode** UserLevel Password

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS method by name)	string	3	**sMN**
Command	Select user level	string	13	**SetAccessMode**
UserLevel	A valid user level must be included in the transmission. Otherwise the LMS400 rejects the command.	int_8	1	**02h** maintenance personnel **03h** authorized client **04h** service
Password	E.g. encoded value for “LMS_400”	uint_32	4	**00000000h** ... **FFFFFFFFh**

Telegram syntax 25: Request “Select user level”

Confirmation

Telegram structure: **sMA SetAccessMode**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Confirmation (SOPAS method acknowledge)	string	3	**sMA**
Command	Select user level	string	13	**SetAccessMode**

Telegram syntax 26: Confirmation of the “Select user level” request

Answer

Telegram structure: **sAN SetAccessMode** ChangeUserLevel

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sAN**
Command	Select user level	string	13	**SetAcessMode**
ChangeUserLevel	A new user level is opened.	bool_1	1	**00h** error **01h** user change successful

Telegram syntax 27: Answer to the “Select user level” request

Example

Request: sMN SetAccessMode 03 B18244B6

Confirmation: sMA SetAccessMode

Answer: sAN SetAccessMode 01

11.5.2 Query user level

Returns the current user level.

Request

Telegram structure: **sMN GetAccessMode**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS method by name)	string	3	**sMN**
Command	Query user level	string	13	**GetAccessMode**

Telegram syntax 28: Request “Query user level”

Confirmation

Telegram structure: **sMA GetAccessMode**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Confirmation (SOPAS method acknowledge)	string	3	**sMA**
Command	Query user level	string	13	**GetAccessMode**

Telegram syntax 29: Confirmation of the “Query user level” request

Answer

Telegram structure: **sAN GetAccessMode** UserLevel

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sAN**
Command	Query user level	string	13	**GetAccessMode**
UserLevel	The current user level is output.	uint_8	1	**00h** nobody logged in **01h** machine operator **02h** maintenance personnel **03h** authorized client **04h** service

Telegram syntax 30: Answer to the “Query user level” request

Example

Request: sMN GetAccessMode

Confirmation: sMA GetAccessMode

Answer: sAN GetAccessMode 03

11.5.3 Set password

Sets a new password for a specific user level. The LMS400 must be in the appropriate user level to perform this action (see *11.5.1 on page 95*). The telegram can then change the password for the same user level or a lower user level.

Request

Telegram structure: **sMN SetPassword** UserLevel NewPassword

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS method by name)	string	3	**sMN**
Command	Set password	string	11	**SetPassword**
UserLevel	User level for which the password is to be changed	int_8	1	**02h** maintenance personnel **03h** authorized client
NewPassword	Hash value for the new password (see *11.5.1 on page 95*)	uint_32	4	**00000000h** ... **FFFFFFFFh**

Telegram syntax 31: Request “Set password”

Confirmation

Telegram structure: **sMA SetPassword**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Confirmation (SOPAS method acknowledge)	string	3	**sMA**
Command	Set password	string	11	**SetPassword**

Telegram syntax 32: Confirmation of the “Set password” request

Answer

Telegram structure: **sAN SetPassword** PasswordSet

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sAN**
Command	Set password	string	11	**SetPassword**
PasswordSet	The new password has been set.	bool_1	1	**00h** error **01h** password has been set

Telegram syntax 33: Answer to the “Set password” request

Example

Request: sMN SetPassword 03 B8F9E6C

Confirmation: sMA SetPassword

Answer: sAN SetPassword 01

11.5.4 Set device name

Sets a device name for the LMS400.

Note The required user level is “Maintenance personnel” (see *11.5.1 on page 95*).

Request

Telegram structure: **sWN LocationName** Length Device name

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS write by name)	string	3	**sWN**
Command	Set device name	string	12	**LocationName**
Length	Number of characters in the device name	uint_16	2	**0000h** 0 ... **0010h** 16
Device name	Flexible range from 0 to 16 characters (20h ... FFh)	string	0 ... 16	**.....**

Telegram syntax 34: Request “Set device name”

Answer

Telegram structure: **sWA LocationName**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS write answer)	string	3	**sWA**
Command	Set device name	string	12	**LocationName**

Telegram syntax 35: Answer to the “Set device name” request

Example

Request: sMN LocationName 0006 ROB003

Answer: sWA LocationName

11.5.5 Save parameters permanently

All parameters transferred using telegrams are first saved in an interim memory in the LMS400. The “Save parameters permanently” telegram permanently saves in the device the scanner parameters that are in the intermediate memory. The parameters are then not lost when the LMS400 is switched off.

Note The LMS400 must be in the user level “Authorized client” (see *11.5.1 on page 95*).

Request

Telegram structure: **sMN mEEwriteall**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS method by name)	string	3	**sMN**
Command	Save parameters permanently	string	11	**mEEwriteall**

Telegram syntax 36: Request “Save parameters permanently”

Confirmation

Telegram structure: **sMA mEEwriteall**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Confirmation (SOPAS method acknowledge)	string	3	**sMA**
Command	Save parameters permanently	string	11	**mEEwriteall**

Telegram syntax 37: Confirmation of the “Save parameters permanently” request

Answer

Telegram structure: **sAN mEEwriteall** ErrorCode

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sAN**
Command	Save parameters permanently	string	11	**mEEwriteall**
ErrorCode	The command has been accepted if the error code 0 is returned.	uint_32	4	**00000000h** no error ... **FFFFFFFFh**

Telegram syntax 38: Answer to the “Save parameters permanently” request

Example

Request: sMN mEEwriteall

Confirmation: sMA mEEwriteall

Answer: sAN mEEwriteall 00000000

11.5.6 Terminate configuration

The LMS400 changes to user level 0.

Request

Telegram structure: **sMN Run**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS method by name)	string	3	**sMN**
Command	Terminate configuration	string	3	**Run**

Telegram syntax 39: Request “Terminate configuration”

Confirmation

Telegram structure: **sMA Run**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Confirmation (SOPAS method acknowledge)	string	3	**sMA**
Command	Terminate configuration	string	3	**Run**

Telegram syntax 40: Confirmation of the “Terminate configuration” request

Answer

Telegram structure: **sAN Run** UserLevel0

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Acknowledgement of receipt (SOPAS method acknowledge)	string	3	**sAN**
Command	Terminate configuration	string	3	**Run**
UserLevel0	User level 0 is opened.	bool_1	1	**00h** false **01h** true

Telegram syntax 41: Answer to the “Terminate configuration” request

Example

Request: sMN Run

Confirmation: sMA Run

Answer: sAN Run 01

11.5.7 Reset device

This telegram resets the LMS400 (see also *2.5 “Quick stop and Quick restart” on page 18*). It retains parameters stored in the internal, non-volatile memory. Measured values on the interface are lost. The LMS400 restarts operation with the last saved parameters.

Notes

- It takes approx. 5 seconds to reset the LMS400. During this period the device cannot receive or process further telegrams.
- The required user level is “Maintenance personnel” (see *11.5.1 on page 95*).

Request

Telegram structure: **sMN mDCreset**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS method by name)	string	3	**sMN**
Command	Reset device	string	8	**mDCreset**

Telegram syntax 42: Request “Reset device”

Confirmation

Telegram structure: **sMA mDCreset**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Confirmation (SOPAS method acknowledge)	string	3	**sMA**
Command	Reset device	string	8	**mDCreset**

Telegram syntax 43: Confirmation of the “Reset device” request

Answer

Telegram structure: **sAN mDCreset** ErrorCode

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Acknowledgement of receipt (SOPAS method acknowledge)	string	3	**sAN**
Command	Reset device	string	8	**mDCreset**
ErrorCode	The command has been accepted if the error code 0 is returned.	uint_32	4	**00000000h** no error ... **FFFFFFFFh**

Telegram syntax 44: Answer to the “Reset device” request

Example

Request: sMN mDCreset

Confirmation: sMA mDCreset

Answer: sAN mDCreset 00000000

11.5.8 Place device in delivery status

All parameters are set to the default delivery status and saved permanently in the EEPROM. All parameters configured previously are lost.

Notes

- It takes approx. 5 seconds to reset the LMS400. During this period the device cannot receive or process further telegrams.
- The required user level is “Authorized client” (see *11.5.1 on page 95*).

Request

Telegram structure: **sMN mMDsetdefault**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS method by name)	string	3	**sMN**
Command	Place device in delivery status	string	13	**mMDsetdefault**

Telegram syntax 45: Request “Place device in delivery status”

Confirmation

Telegram structure: **sMA mMDsetdefault**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Confirmation (SOPAS method acknowledge)	string	3	**sMA**
Command	Place device in delivery status	string	13	**mMDsetdefault**

Telegram syntax 46: Confirmation of the “Place device in delivery status” request

Answer

Telegram structure: **sAN mMDsetdefault** ErrorCode

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Acknowledgement of receipt (SOPAS method acknowledge)	string	3	**sAN**
Command	Place device in delivery status	string	13	**mMDsetdefault**
ErrorCode	The command has been accepted if the error code 0 is returned.	uint_32	4	**00000000h** no error ... **FFFFFFFFh**

Telegram syntax 47: Answer to the “Place device in delivery status” request

Example

Request: sMN mMDsetdefault

Confirmation: sMA mMDsetdefault

Answer: sAN mMDsetdefault 00000000

11.6 Configuration telegrams for master/slave operation

11.6.1 Select type of synchronisation

Defines how an LMS400 synchronises with another LMS400 (see *3.8 "Master/slave operation" on page 33*).

Note The required user level is "Authorized client" (see *11.5.1 on page 95*).

Request

Telegram structure: **sWN SYtype** Synchronisation

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS write by name)	string	3	**sWN**
Command	Select synchronisation	string	6	**SYtype**
Synchronisation	Corresponds to the possible selections available in SOPAS	uint_8	1	**00h** none **01h** master **02h** slave

Telegram syntax 48: Request "Select synchronisation"

Answer

Telegram structure: **sWA SYtype**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS write answer)	string	3	**sWA**
Command	Select synchronisation	string	6	**SYtype**

Telegram syntax 49: Answer to the "Select synchronisation" request

Example

Request: sWN SYtype 01

Answer: sWA SYtype

11.6.2 Define phase

Defines the phase offset of the rotating mirrors on two synchronised LMS400 (see *3.8.1 "Phase offset of the rotating mirrors" on page 33*).

Note The required user level is "Authorized client" (see *11.5.1 on page 95*).

Request

Telegram structure: **sWN SYphas**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS write by name)	string	3	**sWN**
Command	Determine phase	string	6	**SYphas**
Phase	Information on the angle in degrees	uint_32	4	**+0** ... **+60**

Telegram syntax 50: Request "Determine phase"

Answer

Telegram structure: **sWA SYphas**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sWA**
Command	Determine phase	string	6	**SYphas**

Telegram syntax 51: Answer to the “Determine phase” request

Example

Request: sWN SYphas +35

Answer: sWA SYphas

11.7 Reference configuration telegrams for filter setting

11.7.1 Enable filter

Sets one or more filters for the determination of measured values (see *3.7 “Filtering measured values” on page 28*).

Request

Telegram structure: **sWN FLsel** Filter type

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS write by name)	string	3	**sWN**
Command	Enable filter	string	5	**FLsel**
Filter type	Defines which of the filters are enabled. Several filters can be set by the addition of values. 5 = Median + range 9 = Median + mean, etc.	uint_8	1	**+1** median filter **+2** edge filter **+4** range filter **+8** mean filter

Telegram syntax 52: Request “Enable filter”

Answer

Telegram structure: **sWA FLsel**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sWA**
Command	Enable filter	string	5	**FLsel**

Telegram syntax 53: Answer to the “Enable filter” request

Example

Request: sWN FLsel +9

Answer: sWA FLsel

11.7.2 Define median filter

Defines the median filter (see *3.7.2 “Median filter” on page 29*).

Request

Telegram structure: **sWN FLmed** ReservedByte

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS write by name)	string	3	**sWN**
Command	Define median filter	string	5	**FLmed**
ReservedByte	Reserved	uint_8	1	**00h**

Telegram syntax 54: Request “Define median filter”

Answer

Telegram structure: **sWA FLmed**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sWA**
Command	Define median filter	string	5	**FLmed**

Telegram syntax 55: Answer to the “Define median filter” request

Example

Request: sWN FLmed 00

Answer: sWA FLmed

11.7.3 Define range filter

Defines a specific distance range within which measured values are valid and are output (see *3.7.3 “Range filter” on page 31*).

Request

Telegram structure: **sWN FLrang** BottomLimit TopLimit

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS write by name)	string	3	**sWN**
Command	Define range filter	string	6	**FLrang**
BottomLimit	Information in mm	float_32	4	**+700.0000** ... **+3000.0000**
TopLimit	Information in mm	float_32	4	<bottom limit>... **+3000.0000**

Telegram syntax 56: Request “Define range filter”

Answer

Telegram structure: **sWA FLrang**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sWA**
Command	Define range filter	string	6	**FLrang**

Telegram syntax 57: Answer to the “Define range filter” request

Example

Request: sWN FLrang +1000.0000 +2200.0000

Answer: sWA FLrang

11.7.4 Define mean filter

Defines the number of means for the mean filter (see *3.7.4 “Mean filter” on page 31*).

Request

Telegram structure: **sWN FLmean** ReservedByte Mean

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS write by name)	string	3	**sWN**
Command	Define mean filter	string	6	**FLmean**
ReservedByte	Reserved	uint_8	1	**00h**
Mean	Number of means	uint_16	2	**0002h** 2 ... **00C8h** 200

Telegram syntax 58: Request “Define mean filter”

Answer

Telegram structure: **sWA FLmean**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sWA**
Command	Define mean filter	string	6	**FLmean**

Telegram syntax 59: Answer to the “Define mean filter” request

Example

Request: sWN FLmean 0 0014

Answer: sWA FLmean

11.8 Reference configuration telegrams for triggering

11.8.1 Define time or distance controlled triggering

Defines whether the de-bounce on the inputs as well as the delay and expansion of the gate is time or distance controlled (see *11.8.2 on page 109* and *11.8.3 on page 110*).

Note The required user level is “Service” (see *11.5.1 on page 95*).

Request

Telegram structure: **sWN IObase** Control

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS write by name)	string	3	**sWN**
Command	Determines the type of control	string	6	**IObase**
Control	Time or distance based	uint_8	1	**00h** time based **01h** distance based

Telegram syntax 60: Request “Define time or distance controlled triggering”

Answer

Telegram structure: **sWA IObase**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sWA**
Command	Determines the type of control	string	6	**IObase**

Telegram syntax 61: Answer to the “Define time or distance controlled triggering” request

Example

Request: sWN IObase 01

Answer: sWA IObase

11.8.2 Define settings for the digital inputs

Defines the input conditions for the digital inputs 1 and 3 (see *3.5.2 “Control of the measurement process using a gate” on page 23*).

Note The required user level is “Authorized client” (see *11.5.1 on page 95*).

Request

Telegram structure: **sWN IOpins** Parameter IN1 (= DebounceDist DebounceTime Logic) Parameter IN2 Parameter IN3 Parameter IN4

Telegram part		**Description**	**Variable type**	**Length (byte)**	**Value range**	
Type of command		Request (SOPAS write by name)	string	3	**sWN**	
Command		Settings for the digital inputs	string	6	**IOpins**	
Is repeated for each input (IN1 to IN4)	DebounceDist	De-bounce over distance, information in millimetres (see also *11.8.1*)	int_16	2	**FC13h** **03E8h**	–1000 mm ... 1000 mm
	DebounceTime	De-bounce over time, information in milliseconds (see also *11.8.1*)	uint_16	2	**0000h** **03E8h**	0 ms ... 1000 ms
	Logic	Defines the input logic	uint_8	1	**00h** **01h**	active low active high

Telegram syntax 62: Request “Settings for the digital inputs”

Answer

Telegram structure: **sWA IOpins**

Telegram part	**Description**	**Variable type**	**Length (byte)**	**Value range**
Type of command	Answer (SOPAS answer)	string	3	**sWA**
Command	Settings for the digital inputs	string	6	**IOpins**

Telegram syntax 63: Answer to the “Settings for the digital inputs” request

Example

Request: sWN IOpins 0000 0000 01 0000 0000 01 0000 0000 01 0000 0000 01

Answer: sWA IOpins

11.8.3 Define the gate settings

Defines how the gate is triggered (see *3.5.2 “Control of the measurement process using a gate” on page 23*).

Note The required user level is “Authorized client” (see *11.5.1 on page 95*).

Request

Telegram structure: **sWN lOgcfg** StopLikeStart ParameterGATEON (Source TimeDelay ExpansionTime DistanceDelay ExpansionDistance) ParameterGATEOFF

Telegram part		Description	Variable type	Length (byte)	Value range
Type of command		Request (SOPAS write by name)	string	3	**sWN**
Command		Trigger settings	string	6	**lOgcfg**
StopLikeStart		The same parameters are used for stopping the gate as for the start	bool_1	1	**00h** false **01h** true
Is repeated for starting and stopping the gate	Source	Input 1 or 3, if the trigger source is connected directly to the LMS400. Software trigger, if the trigger is applied via a telegram (see *11.3.3 on page 87*).	uint_8	1	**00h** input 1 **02h** input 3 **04h** software trigger **05h** CAN-BUS **06h** input 1 AND 3 **07h** input 1 OR 3 **08h** test trigger **09h** master **0Fh** none
	TimeDelay	Delays the start of the measurement, information in milliseconds	uint_16	2	**0000h** 0 ms ... **FFFFh** 65535 ms
	Expansion time	Enlarges the gate, information in milliseconds	int_16	2	**FC13h** –1000 mm ... **03E8h** 1000 mm
	DistanceDelay	Delays the start of the measurement, information in millimetres	uint_16	2	**0000h** 0 mm ... **FFFFh** 65535 mm
	Expansion distance	Enlarges the gate, information in millimetres	int_16	2	**F830h** –2000 mm ... **07D8h** 2000 mm

Telegram syntax 64: Request “Gate settings”

Answer

Telegram structure: **sWA IOgcfg**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sWA**
Command	Trigger settings	string	6	**IOgcfg**

Telegram syntax 65: Answer to the “Gate settings” request

Example

Request: sWN IOgcfg 01 08 07D0 0000 0014 0032 00 0000 0000 0000 0000

Answer: sWA IOgcfg

11.8.4 Enable laser control

Enables the laser control that is used to switch on the laser with a trigger and switch it off after a certain distance is reached.

Note The required user level is “Authorized client” (see *11.5.1 on page 95*).

Request

Telegram structure: **sWN IOlasc** TriggerSource LaserSwitchOffDistance LaserSwitchOffDelay LaserControl

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS write by name)	string	3	**sWN**
Command	Enable laser control	string	6	**IOlasc**
TriggerSource	Defines the trigger used to switch on the laser.	uint_8	1	**00h** input 1 **02h** input 3 **04h** software trigger **05h** CAN-BUS **06h** input 1 AND 3 **07h** input 1 OR 3 **08h** test trigger **09h** master
LaserSwitchOffDistance	Defines the distance after which the laser is switched off. Maximum value	uint_16	2	**0000h** 0 mm ... **1770h** 6000 mm (For an encoder resolution of 0.2 mm/incr.)
LaserSwitchOffDelay	Defines the time after which the laser is switched off.	uint_16	2	**0000h** 0 min ... **FFFFh** 65535 min
LaserControl	The laser control function is used.	uint_8	1	**00h** deactivated **01h** own source **02h** gate controlled (see *11.8.3*)

Telegram syntax 66: Request “Enable laser control”

Answer

Telegram structure: **sWA lOlasc**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sWA**
Command	Enable laser control	string	6	**lOlasc**

Telegram syntax 67: Answer to the “Enable laser control” request

Example

Request: sWN lOlasc 02 0DAC 0005 01

Answer: sWA lOlasc

11.8.5 Select the encoder settings

Defines the type of encoder used (see *3.5.4 “Connection of encoders” on page 25*).

Note The required user level is “Authorized client” (see *11.5.1 on page 95*).

Request

Telegram structure: **sWN lOencm Encoder type**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS write by name)	string	3	**sWN**
Command	Encoder settings	string	6	**lOencm**
Encoder type	Defines the encoder used	uint_8	1	**00h** no encoder **01h** DIn 2 **02h** direction detection (phase) DIn2/DIn4 **03h** direction detection (level) DIn2/DIn4 **04h** constant velocity

Telegram syntax 68: Request “Encoder settings”

Answer

Telegram structure: **sWA lOencm**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sWA**
Command	Encoder settings	string	6	**lOencm**

Telegram syntax 69: Answer to the “Encoder settings” request

Example

Request: sWN lOencm 02

Answer: sWA lOencm

11.9 Configuration telegrams for the outputs

11.9.1 Enable outputs

Activates the outputs 1 to 5.

Notes

- The required user level is “Authorized client” (see *11.5.1 on page 95*).
- The telegram cannot be used if the Level Control application is used (see *11.14.1 on page 131*).

Request

Telegram structure: **sMN mMSsetoutput** OutputNumber OutputValue

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS method by name)	string	3	**sMN**
Command	Enable output	string	12	**mMSsetoutput**
OutputNumber	Defines the output that is to be configured.	uint_8	1	**01h** digital output 1 **02h** digital output 2 **03h** digital output 3 **04h** digital output 4 **05h** analog output
OutputValue	Activates/deactivates the output defined with the OutputNumber	uint_16	2	digital outputs: **0** inactive **1** active Analog output: **0 ... 20000** (in mA)

Telegram syntax 70: Request “Enable outputs”

Confirmation

Telegram structure: **sMA mMSsetoutput**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Confirmation (SOPAS method acknowledge)	string	3	**sMA**
Command	Enable output	string	12	**mMSsetoutput**

Telegram syntax 71: Confirmation of the request “Enable outputs”

Answer

Telegram structure: **sAN mMSsetoutput** ErrorCode

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sAN**
Command	Enable output	string	12	**mMSsetoutput**
ErrorCode	The command has been accepted if the error code 0 is returned.	uint_32	4	**00000000h** no error ... **FFFFFFFFh**

Telegram syntax 72: Answer to the “Enable Outputs” request

Example

Request: sMN mMSsetoutput 21

Confirmation: sMA mMSsetoutput

Answer: sAN mMSsetoutput 00000000

11.10 Configuration telegrams for the host interface

11.10.1 Select hardware setting

Defines the host interface as RS-232 or as RS-422 interface (see *3.11 “Interface specification” on page 48*).

Note The required user level is “Authorized client” (see *11.5.1 on page 95*).

Request

Telegram structure: **sWN HIr422** Hardware

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS write by name)	string	3	**sWN**
Command	Hardware setting for the host interface	string	6	**HIr422**
Hardware	Defines whether RS-232 or RS-422	uint_8	1	**00h** RS-232 **01h** RS-422

Telegram syntax 73: Request “Hardware setting for the host interface”

Answer

Telegram structure: **sWA HIr422**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sWA**
Command	Hardware setting for the host interface	string	6	**HIr422**

Telegram syntax 74: Answer to the “Hardware setting for the host interface” request

Example

Request: sWN HIr422 01

Answer: sWA HIr422

11.10.2 Select baud rate

Defines the baud rate for the host interface (see *3.11 “Interface specification” on page 48*).

Note The required user level is “Authorized client” (see *11.5.1 on page 95*).

Request

Telegram structure: **sWN HIbaud** Baud rate

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS write by name)	string	3	**sWN**
Command	Baud rate of the host interface	string	6	**HIbaud**
Baud rate	Defines the baud rate for the host interface	uint_8	1	**00h** 300 Baud **01h** 600 Baud **02h** 1200 Baud **03h** 2400 Baud **04h** 4800 Baud **05h** 9600 Baud **06h** 19200 Baud **07h** 38400 Baud **08h** 57600 Baud **09h** 115200 Baud

Telegram syntax 75: Request “Baud rate for the host interface”

Answer

Telegram structure: **sWA HIbaud**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sWA**
Command	Baud rate of the host interface	string	6	**HIbaud**

Telegram syntax 76: Answer to the “Baud rate for the host interface” request

Example

Request: sWN HIbaud 09

Answer: sWA HIbaud

11.10.3 Select number of stop bits

Defines the number of stop bits for the host interface (see *3.11 “Interface specification” on page 48*).

Note The required user level is “Authorized client” (see *11.5.1 on page 95*).

Request

Telegram structure: **sWN HIstop** Stop bits

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS write by name)	string	3	**sWN**
Command	Number of stop bits for the host interface	string	6	**HIstop**
Stop bits	Defines the number of stop bits for the host interface	uint_8	1	**00h** 1 stop bit **01h** 2 stop bits

Telegram syntax 77: Request “Number of stop bits for the host interface”

Answer

Telegram structure: **sWA HIstop**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sWA**
Command	Number of stop bits for the host interface	string	6	**HIstop**

Telegram syntax 78: Answer to the “Number of stop bits for the host interface” request”

Example

Request: sWN HIstop 01

Answer: sWA HIstop

11.10.4 Select data and parity bits

Defines the combination of data and parity bits for the host interface (see *3.11 “Interface specification” on page 48*).

Note The required user level is “Authorized client” (see *11.5.1 on page 95*).

Request

Telegram structure: **sWN HIdpar** DataParityBit

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS write by name)	string	3	**sWN**
Command	Data and parity bits for the host interface	string	6	**HIdpar**
DataParityBit	Defines the baud rate for the host interface.	uint_8	1	**00h** 8 data bits, no parity **01h** 8 data bits, even **02h** 7 data bits, even **03h** 8 data bits, odd **04h** 7 data bits, odd **05h** 7 data bits, MarkP **06h** 7 data bits, SpaceP

Telegram syntax 79: Request “Data and parity bits for the host interface”

Answer

Telegram structure: **sWA HIdpar**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sWA**
Command	Data and parity bits for the host interface	string	6	**HIdpar**

Telegram syntax 80: Answer to the “Data and parity bits for the host interface” request

Example

Request: sWN HIdpar 01

Answer: sWA HIdpar

11.10.5 Select block check byte setting

Defines whether the block check byte is sent and, if so, whether it is sent before or after the stop bit.

Notes

- The required user level is “Authorized client” (see *11.5.1 on page 95*).
- The setting affects the telegram frame (see *Tab. 14 on page 50*).

Request

Telegram structure: **sWN HIblck** Block check

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS write by name)	string	3	**sWN**
Command	Block check byte setting for the host interface	string	6	**HIblck**
Block check	Defines whether a block check byte is sent and whether it is sent before or after the stop bit	uint_8	1	**00h** none **01h** before the stop bit **02h** after the stop bit

Telegram syntax 81: Request “Block check byte setting for the host interface”

Answer

Telegram structure: **sWA HIblck**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sWA**
Command	Block check byte setting for the host interface	string	6	**HIblck**

Telegram syntax 82: Answer to the “Block check byte setting for the host interface” request

Example

Request: sWN HIblck 02

Answer: sWA HIblck

11.10.6 Select receive start character

Defines which character is detected as the start character for the telegrams from the LMS400 (see *3.12.1 “Frame and coding for the telegrams” on page 50*).

Note The required user level is “Authorized client” (see *11.5.1 on page 95*).

Request

Telegram structure: **sWN HIrpre** ReceiveStartCharacter

Telegram part	Description	Variable type	Length (byte)	Value range	
Type of command	Request (SOPAS write by name)	string	3	**sWN**	
Command	Start character for telegrams to be received	string	6	**HIrpre**	
ReceiveStartCharacter	Hex value for the start character	uint_8	1	**00h** NUL **01h** SOH **02h** STX **03h** ETX **04h** EOT **05h** ENQ **07h** BEL **08h** BS **09h** HT **0Ah** LF **0Bh** VT **0Ch** FF **0Dh** CR **0Eh** SO **0Fh** SI	**10h** DLE **12h** DC2 **14h** DC4 **16h** SYN **17h** ETB **18h** CAN **19h** EM **1Ah** SUB **1Bh** ESC **1Ch** FSP **1Dh** GSP **1Eh** RSP **1Fh** USP **20h** SPC **7Fh** DEL

Telegram syntax 83: Request “Start character for telegrams to be received”

Answer

Telegram structure: **sWA HIrpre**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sWA**
Command	Start character for telegrams to be received	string	6	**HIrpre**

Telegram syntax 84: Answer to the “Start character for telegrams to be received” request

Example

Request: sWN HIrpre 02

Answer: sWA HIrpre

11.10.7 Select receive stop character

Defines which character is detected as the stop character for the telegrams from the LMS400 (see *3.12.1 “Frame and coding for the telegrams” on page 50*).

Notes

- The required user level is “Authorized client” (see *11.5.1 on page 95*).
- The setting affects the telegram frame (see *Tab. 14 on page 50*).

Request

Telegram structure: **sWN HIrpst** ReceiveStopCharacter1 ReceiveStopCharacter2

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS write by name)	string	3	**sWN**
Command	Stop character for telegrams to be received	string	6	**HIrpst**
ReceiveStopCharacter1	Hex value of stop character 1	uint_8	1	See “ReceiveStartCharacter” in *11.10.6 on page 119*
ReceiveStopCharacter2	Hex value of stop character 2 If only one stop character is to be used, then set the second byte to ZERO (00h)	uint_8	1	

Telegram syntax 85: Request “Stop character for telegrams to be received”

Answer

Telegram structure: **sWA HIrpst**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sWA**
Command	Stop character for telegrams to be received	string	6	**HIrpst**

Telegram syntax 86: Answer to the “Stop character for telegrams to be received” request

Example

Request: sWN HIrpst 0D 0A

Answer: sWA HIrpst

.

11.10.8 Select send start character

Defines which start character is sent by the LMS400 before the data in the telegram (see *3.12.1 “Frame and coding for the telegrams” on page 50*).

Note The required user level is “Authorized client” (see *11.5.1 on page 95*).

Request

Telegram structure: **sWN HItpre** SendStartCharacter

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS write by name)	string	3	**sWN**
Command	Start character for telegrams to be sent	string	6	**HItpre**
SendStartCharacter	Hex value for the start character	uint_8	1	See “ReceiveStartCharacter” in *11.10.6 on page 119*

Telegram syntax 87: Request “Start character for telegrams to be sent”

Answer

Telegram structure: **sWA HItpre**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sWA**
Command	Start character for telegrams to be sent	string	6	**HItpre**

Telegram syntax 88: Answer to the “Start character for telegrams to be sent” request

Example

Request: sWN HItpre 02

Answer: sWA HItpre

11.10.9 Select send stop character

Defines which stop character is sent by the LMS400 after the data in the telegram (see *3.12.1 “Frame and coding for the telegrams” on page 50*).

Notes

- The required user level is “Authorized client” (see *11.5.1 on page 95*).
- The setting affects the telegram frame (see *Tab. 14 on page 50*).

Request

Telegram structure: **sWN HItpst** SendStopCharacter1 SendStopCharacter2

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS write by name)	string	3	**sWN**
Command	Stop character for telegrams to be sent	string	6	**HItpst**
SendStopCharacter1	Hex value of stop character 1	uint_8	1	See “ReceiveStartCharacter” in *11.10.6 on page 119*
SendStopCharacter2	Hex value for stop character 2. If only one stop character is to be used, then set the second byte to ZERO (00h).	uint_8	1	

Telegram syntax 89: Request “Stop character for telegrams to be sent”

Answer

Telegram structure: **sWA HItpst**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sWA**
Command	Stop character for telegrams to be sent	string	6	**HItpst**

Telegram syntax 90: Answer to the “Stop character for telegrams to be sent” request

Example

Request: sWN HItpst 0D 0A

Answer: sWA HItpst

11.11 Configuration telegrams for the Ethernet interface

Note For the configuration telegram for the Ethernet interface to be effective, the LMS400 must be reset after successful transfer of the telegram (see *11.5.7 on page 102*).

11.11.1 Defining the IP address of the LMS400

Defines the IP address of the LMS400 (see *3.11 "Interface specification" on page 48*).

Note The required user level is "Authorized client" (see *11.5.1 on page 95*).

Request

Telegram structure: **sWN Elip** IPAddress

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS write by name)	string	3	**sWN**
Command	IP address of the Ethernet interface	string	4	**Elip**
IPAddress	Hex value for the IP address	Array of 4 × uint_8	4	**00h 00h 00h 00h** 0.0.0.0 … **FFh FFh FFh FFh** 255.255.255.255

Telegram syntax 91: Request "IP address for the Ethernet interface"

Answer

Telegram structure: **sWA Elip**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sWA**
Command	IP address of the Ethernet interface	string	4	**Elip**

Telegram syntax 92: Answer to the "IP address for the Ethernet interface" request

Example

Request: sWN EIip C6 A8 14 01

Answer: sWA EIip

11.11.2 Define gateway address for the Ethernet interface

Defines the gateway address for the Ethernet interface (see *3.11 “Interface specification” on page 48*).

Note The required user level is “Authorized client” (see *11.5.1 on page 95*).

Request

Telegram structure: **sWN EIgate** GatewayAddress

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS write by name)	string	3	**sWN**
Command	Gateway address for the Ethernet interface	string	6	**EIgate**
GatewayAddress	Hex value for the gateway address	Array of 4 × uint_8	4	**00h 00h 00h 00h** 0.0.0.0 … **FFh FFh FFh FFh** 255.255.255.255

Telegram syntax 93: Request “Gateway address for the Ethernet interface”

Answer

Telegram structure: **sWA EIgate**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sWA**
Command	Gateway address for the Ethernet interface	string	6	**EIgate**

Telegram syntax 94: Answer to the “Gateway address for the Ethernet interface” request

Example

Request: sWN EIgate C6 A8 73 01

Answer: sWA EIgate

11.11.3 Define the subnet mask for the Ethernet interface

Defines the gateway address for the Ethernet interface (see *3.11 “Interface specification” on page 48*).

Note The required user level is “Authorized client” (see *11.5.1 on page 95*).

Request

Telegram structure: **sWN EIMsak** Subnet mask

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS write by name)	string	3	**sWN**
Command	Subnet mask for the Ethernet interface	string	6	**EIMask**
Subnet mask	Hex value for the subnet mask	Array of 4 × uint_8	4	**00h 00h 00h 00h** 0.0.0.0 … **FFh FFh FFh FFh** 255.255.255.255

Telegram syntax 95: Request “Subnet mask for the Ethernet interface”

Answer

Telegram structure: **sWA EIMask**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sWA**
Command	Subnet mask for the Ethernet interface	string	6	**EIMask**

Telegram syntax 96: Answer to the “Subnet mask for the Ethernet interface” request

Example

Request: sWN EIMask FF FF FF 00

Answer: sWA EIMask

11.11.4 Define port for the TCP/IP communication

Defines the port for the TCP/IP communication (see *3.11 “Interface specification” on page 48*).

Note The required user level is “Authorized client” (see *11.5.1 on page 95*).

Request

Telegram structure: **sWN EIport** Port

Telegram part	Description	Variable type	Length (byte)	Value range	
Type of command	Request (SOPAS write by name)	string	3	**sWN**	
Command	Port for the TCP/IP communication	string	6	**EIport**	
Port	Hex value for the port	uint_16	2	**0000h** **FFFFh**	0 ... 65535

Telegram syntax 97: Request “Port for the TCP/IP communication”

Answer

Telegram structure: **sWA EIport**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sWA**
Command	Port for the TCP/IP communication	string	6	**EIport**

Telegram syntax 98: Answer to the “Port for the TCP/IP communication” request

Example

Request: sWN EIport 0461

Answer: sWA EIport

11.11.5 Read MAC address of the LMS400

Reads the MAC address (Media Access Control address) of the LMS400.

Request

Telegram structure: **sRN EImac**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS read by name)	string	3	**sRN**
Command	Define MAC address for the TCP/IP communication	string	5	**EImac**

Telegram syntax 99: Request “Define MAC address for TCP/IP communication”

Answer

Telegram structure: **sRA EImac** MACAddress

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sRA**
Command	Define MAC address for the TCP/IP communication	string	5	**EImac**
MACAddress	The MAC address of the LMS400 is output.	string	17	**00-06-77-00-00-00** … **00-06-77-FF-FF-FF**

Telegram syntax 100: Answer to the “Define MAC address for the TCP/IP communication” request

Example

Request: sRN EImac

Answer: sRA EImac 00-06-77-00-00-00

11.11.6 Select format for the CoLa protocol

Defines the transmission protocol of the LMS400.

Note The required user level is “Authorized client” (see *11.5.1 on page 95*).

Request

Telegram structure: **sWN EIcola** CoLaProtokol

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS write by name)	string	3	**sWN**
Command	Define format of the CoLa protocol	string	6	**EIcola**
CoLaProtokol	Defines whether transmission is in binary or ASCII	uint_8	1	**00h** ASCII (Cola-A) **01h** binary (Cola-B)

Telegram syntax 101: Request “Define format of the CoLa protocol”

Answer

Telegram structure: **sWA EIcola**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sWA**
Command	Define format of the CoLa protocol	string	6	**EIcola**

Telegram syntax 102: Answer to the “Define format of the CoLa protocol” request

Example

Request: sWN EIcola 01

Answer: sWA EIcola

11.12 Reference status log telegrams

11.12.1 Query warnings

Request

Telegram structure: **sRN MSwarn**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS write by name)	string	3	**sRN**
Command	Query warnings	string	6	**MSwarn**

Telegram syntax 103: Request “Query warnings”

Answer

Telegram structure: **sRA MSwarn** ParameterWARNING1 (ErrorCode FirstTimePwrOnCnt FirstTimeOpHours FirstTimeDailyOpHours LastTimePwrOnCnt LastTimeOpHours LastTimeDailyOpHours Number ReservedBytes) ParameterWARNING2 ParameterWARNING3 ParameterWARNING4 ParameterWARNING5

Telegram part		Description	Variable type	Length (byte)	Value range
Type of command		Answer (SOPAS answer)	string	3	**sRA**
Command		Query warnings	string	6	**MSwarn**
Parameters are repeated five times (per memory for warnings, independent of whether these are filled)	ErrorCode	Hex value of the error code saved	uint_32	4	**00000001h** … **FFFFFFFFh**
	FirstTimePwrOnCnt	Switch on counter reading at the time of first occurrence	uint_16	2	**0000h** … **FFFFh**
	FirstTimeOpHours	Overall operating hours counter reading at the time of first occurrence	uint_16	2	**0000h** … **FFFFh**
	FirstTimeDailyOpHours	Operating hours counter reading at the time of first occurrence	uint_16	2	**0000h** … **FFFFh**
	LastTimePwrOnCnt	Switch on counter reading at the time of last occurrence	uint_16	2	**0000h** … **FFFFh**
	LastTimeOpHours	Overall operating hours counter reading at the time of last occurrence	uint_16	2	**0000h** … **FFFFh**
	LastTimeDailyOpHours	Operating hours counter reading at the time of last occurrence	uint_16	2	**0000h** … **FFFFh**
	Number	Indicates a number of occurrences	uint_16	2	**0000h** … **FFFFh**
ReservedBytes		Reserved	uint_16	2	**0000h**

Telegram syntax 104: Answer to the “Query warnings” request

Example

Request: sWN MSwarn

Answer: sWA MSwarn 4C0ACC0A 00A4 0255 0005 00A4 0256 0006 0256 0000 4C0ACC0B 00A4 0255 0005 00A4 0256 0006 0256 0000 4C0ACC0C 00A4 0255 0005 00A4 0256 0006 0256 0000 4C0ACC0B 00A4 0255 0005 00A4 0256 0006 0256 0000 4C0ACC0C 00A4 0255 0005 00A4 0256 0006 0256 0000

11.12.2 Query errors

Request

Telegram structure: **sRN MSerr**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS write by name)	string	3	**sRN**
Command	Query errors	string	5	**MSerr**

Telegram syntax 105: Request “Query errors”

Answer

Telegram structure: **sRA MSerr** ParameterERROR1 (ErrorCode FirstTimePwrOnCnt FirstTimeOpHours FirstTimeDailyOpHours LastTimePwrOnCnt LastTimeOpHours LastTimeDailyOpHours Number ReservedBytes) ParameterERROR2 ParameterERROR3 ParameterERROR4 ParameterERROR5

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sRA**
Command	Query errors	string	5	**MSerr**
	For parameters see *Telegram syntax 104: on page 128*			
ReservedBytes	Reserved	uint_16	2	**0000h**

Telegram syntax 106: Answer to the “Query errors” request

Example

Request: sRN MSerr

Answer: sRA MSerr 4303C303 00A4 0255 0005 00A4 0256 0006 0256 0000 00000000 0000 0000 0000 0000 0000 0000 0000 0000 00000000 0000 0000 0000 0000 0000 0000 0000 0000 00000000 0000 0000 0000 0000 0000 0000 0000 0000 00000000 0000 0000 0000 0000 0000 0000 0000 0000

11.12.3 Query serious errors

Request

Telegram structure: **sRN MSfat**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS write by name)	string	3	**sRN**
Command	Query serious errors	string	5	**MSfat**

Telegram syntax 107: Request “Query serious errors”

Answer

Telegram structure: **sRA MSfat** ParameterSERIOUSERROR1 (ErrorCode FirstTimePwrOnCnt FirstTimeOpHours FirstTimeDailyOpHours LastTimePwrOnCnt LastTimeOpHours LastTimeDailyOpHours Number ReservedBytes) ParameterSERIOUSERROR2 ParameterSERIOUSERROR3 ParameterSERIOUSERROR4 ParameterSERIOUSERROR5

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sRA**
Command	Query serious errors	string	5	**MSfat**
	For parameters see *Telegram syntax 104: on page 128*			
ReservedBytes	Reserved	uint_16	2	**0000h**

Telegram syntax 108: Answer to the “Query serious errors” request

Example

Request: sRN MSfat

Answer: sRA MSfat 4912C912 00A4 0255 0005 00A4 0256 0006 0256 0000 00000000 0000 0000 0000 0000 0000 0000 0000 0000 00000000 0000 0000 0000 0000 0000 0000 0000 0000 00000000 0000 0000 0000 0000 0000 0000 0000 0000 00000000 0000 0000 0000 0000 0000 0000 0000 0000

11.13 Error codes

Value	Class	Possible cause	Comment
00000000h	No error		
XXXXC303h	Error	Motor speed too low • compare with frequency range (see *10.1 on page 75*)	Laser has been shutdown, as eye protection is no longer assured.
XXXXC304h	Error	Laser power is too low	Maybe malfunction due to external light
XXXXC612h	Info	The connection to the host has been lost.	The device is still continuing to attempt to send data. Initiate a new request or re-start the LMS400.
XXXXC614h	Info		
XXXXC912h	Serious error	EEPROM faulty	Please contact SICK service.
XXXXC913h	Error	The required motor velocity cannot be achieved.	Please contact SICK service.
XXXXCC03h	Info	One of the device self-tests has failed.	Not a critical state
XXXXCC0Ah	Warning	Self-test on a temperature sensor has failed	Not a critical state
XXXXCC0Bh	Warning		
XXXXCC0Ch	Warning		
XXXXCE01h	Info	Parameters outside the valid range	A parameter that has been transferred with the aid of a configuration telegram was outside its valid range.
XXXXCE02h	Info	Parameter too high	
XXXXCE03h	Info	Parameter too low	
XXXXCE04h	Info	Password transferred incorrect	Check the hash value transferred (see *11.5.1 on page 95*).
XXXXCE04h	Info	User level transferred too low	Check whether the user level with which the application has logged in is adequate for the telegram (see *11.2 on page 80*).
XXXXCF04h	Info	Device busy	The LMS400 was processing a previous command.
XXXXCF05h	Info	Command unknown	Check the syntax of the telegram transferred.

Tab. 28: Error codes

Note The first four digits of the error code do not need to be used to evaluate the error, only the last four digits are relevant.

11.14 Telegrams for Level Control

11.14.1 Start measured value output for Level Control

Measured values are only output on request over the related interface. You can configure in SOPAS ET which measured value telegram is output as a reply (see *"Output of the column values" on page 133* or *"Output of the switching point states" on page 135*).

PROJECT TREE, LMS400-XX00, PARAMETER, LEVEL CONTROL, TELEGRAM, area SETTINGS

Request

Telegram structure: **sMN mLRreqlevelcontroldata** Number

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS method by name)	string	3	**sMN**
Command	Data request	string	22	**mLRreqlevelcontroldata**
Number	Number of results that are to be output	uint_16	2	**0000h** continuous **0001h** 1 result ... **000Ah** 10 results

Telegram syntax 109: Request “Data request”

Confirmation

Telegram structure: **sMA mLRreqlevelcontroldata**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Confirmation (SOPAS method acknowledge)	string	3	**sMA**
Command	Data request	string	22	**mLRreqlevelcontroldata**

Telegram syntax 110: Confirmation of the “Data request” request

Answer

Telegram structure: **sAN mLRreqlevelcontroldata** ErrorCode

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sAN**
Command	Data request	string	22	**mLRreqlevelcontroldata**
ErrorCode	The command has been accepted if the error code 0 is returned.	uint_32	4	**00000000h** no error ... **FFFFFFFFh** ErrorCode

Telegram syntax 111: Answer to the “Data request” request”

Output of the column values

Note The contents of this telegram are sent in Big Endian format.

Telegram structure: Format ResultScaling StartingAngle AngularStepWidth NumberMeasuredValues ScanningFrequency NumberColumns ResultColumn_1 StatusColumn_1[ResultColumn_2 StatusColumn_2 … ResultColumn_n StatusColumn_n] DigitalInputs DigitalOutputs AnalogueOutput EncoderPosition ReservedBytesA ReservedBytesB ScanCounter TelegramCounter SystemCounter

Telegram part		Description	Variable type	Length (byte)	Value range
Definition block	Format	Type of measured value output	uint_16	2	**0001h** Column values
	ResultScaling	Scaling of the results in column 1 to n. The values are to be multiplied by this factor.	uint_16	2	**0001h**
	StartingAngle	Information 1/10000 degree	uint_32	4	**550000** … **1250000**
	AngularStepWidth	Information 1/10000 degree	uint_16	2	**1000** … **10000**
	NumberMeasuredValues	Number of measured values in scan	uint_16	2	**1** … **700**
	ScanningFrequency	Information in Hertz	uint_16	2	**150** … **500**
	NumberColumns	Number of columns configured	uint_16	2	**0** … **50**
Results	ResultColumn_1	Value measured in the column in millimetres. The value is to be multiplied by ResultScaling.	int_16	2	**–32768** … **32767**
	StatusColumn_1	Status of the column Several statuses are output by using a logical operator on the values (example 0003 = “Quality not reached” and “no values”).	uint_16	2	**0000h** Ok **0001h** Quality not reached **0002h** no values **0004h** More than 3000 values in X direction (see *“Interval” on page 47*)
	ResultColumn_n	See above	int_16	2	See above
	StatusColumn_n	See above	uint_16	2	See above

Telegram part		Description	Variable type	Length (byte)	Value range
Status	DigitalInputs	The least significant byte reflects the state of the digital inputs by bit. The least significant bit corresponds to input 1.	uint_16	2	**0000h** all inputs off **0001h** input 1 on **0002h** input 2 on **0003h** inputs 1 and 2 on ... **000Fh** all inputs on
	DigitalOutputs	The least significant byte reflects the state of the digital outputs by bit. The least significant bit corresponds to output 1.	uint_16	2	**0000h** all outputs off **0001h** output 1 on **0002h** output 2 on **0003h** outputs 1 and 2 on ... **000Fh** all outputs on
	AnalogueOutput	Output of the state in 10 µA steps	uint_16	2	**0000h** 0 µA **000Ah** 10 µA **0014h** 20 µA ... **4E20h** 20 mA
	EncoderPosition	Information in ticks	uint_16	2	**0000h** ... **FFFFh**
	ReservedBytesA	Reserved	uint_16	2	**0000h**
	ReservedBytesB	Reserved	uint_16	2	**0000h**
	ScanCounter	Counter, starting with the first scan after confirmation of the measured value output. Starts again at 0 when the upper limit is reached (= 1. scan).	uint_16	2	**0000h** 0 ... **0FFFh** 4095
	TelegramCounter	Counter starting with the first telegram after confirmation of the measured value output. Starts again at 0 when the upper limit is reached (= 1. telegram).	uint_16	2	**0000h** 0 ... **FFFFh** 65535
	SystemCounter	Enables the relative time difference between two telegrams to be calculated. Information in 1/327.68 µs. Starts again at 0 when the upper limit is reached	uint_16	2	**0000h** 0 µs ... **FFFFh** 21.4745 s

Telegram syntax 112: Column values

Example

Request: sMN mLRreqlevelcontroldata 0000

Confirmation: sMA mLRreqlevelcontroldata

Answer: sAN mLRreqlevelcontroldata 00000000

Output of the column values: (see *Tab. 112*)

Output of the switching point states

Telegram structure: Format ResultScaling StartingAngle AngularStepWidth NumberMeasuredValues ScanningFrequency NumberColumns ResultColumn_1 StatusColumn_1[ResultColumn_2 StatusColumn_2 … ResultColumn_n StatusColumn_n] DigitalInputs DigitalOutputs AnalogueOutput EncoderPosition ReservedBytesA ReservedBytesB ScanCounter TelegramCounter SystemCounter

Telegram part		**Description**	**Variable type**	**Length (byte)**	**Value range**
Definition block	Format	Defines the type of measured value telegram	uint_16	2	**0101h** Switching point states
	ResultScaling	Scaling of the results in column 1 to n. The values are to be multiplied by this factor.	uint_16	2	**0001h**
	StartingAngle	Information 1/10000 degree	uint_32	4	**550000** ... **1250000**
	AngularStepWidth	Information 1/10000 degree	uint_16	2	**1000** ... **10000**
	NumberMeasuredValues	Number of measured values in scan	uint_16	2	**0** ... **700**
	ScanningFrequency	Information in Hertz	uint_16	2	**150** ... **500**
	NumberColumns	Number of columns configured	uint_16	2	**0** ... **50**
Results	ResultColumn_1	Status of the switching points configured	uint_8	1	**00h** not reached **01h** exceeded
	StatusColumn_1	Status of the column	uint_16	2	**0000h** OK **0001h** quality not reached **0002h** no values **0004h** more than 3000 values in X direction (see *"Interval" on page 47*)
	ResultColumn_n	See above	uint_8	1	See above
	StatusColumn_n	See above	uint_8	1	See above

Telegram part		Description	Variable type	Length (byte)	Value range
Status	DigitalInputs	The least significant byte reflects the state of the digital inputs by bit. The least significant bit corresponds to input 1.	uint_16	2	**0000h** all inputs off **0001h** input 1 on **0002h** input 2 on **0003h** inputs 1 and 2 on ... **000Fh** all inputs on
	DigitalOutputs	The least significant byte reflects the state of the digital outputs by bit. The least significant bit corresponds to output 1.	uint_16	2	**0000h** all outputs off **0001h** output 1 on **0002h** output 2 on **0003h** outputs 1 and 2 on ... **000Fh** all outputs on
	AnalogueOutput	Output of the state in 10 µA steps	uint_16	2	**0000h** 0 µA **000Ah** 10 µA **0014h** 20 µA ... **4E20h** 20 mA
	EncoderPosition	Information in ticks	uint_16	2	**0000h** ... **FFFF**
	ReservedBytesA	Reserved	uint_16	2	**0000h**
	ReservedBytesB	Reserved	uint_16	2	**0000h**
	ScanCounter	Counter, starting with the first scan after confirmation of the measured value output. Starts again at 0 when the upper limit is reached (= 1. scan).	uint_16	2	**0000h** 0 ... **0FFFh** 4095
	TelegramCounter	Counter starting with the first telegram after confirmation of the measured value output. Starts again at 0 when the upper limit is reached (= 1. telegram).	uint_16	2	**0000h** 0 ... **FFFFh** 65535
	SystemCounter	Enables the relative time difference between two telegrams to be calculated. Information in 1/327.68 µs. Starts again at 0 when the upper limit is reached.	uint_16	2	**0000h** 0 µs ... **FFFFh** 21.4745 s

Telegram syntax 113: Switching point states

Example

Request: sMN mLRreqlevelcontroldata 0000

Confirmation: sMA mLRreqlevelcontroldata

Answer: sAN mLRreqlevelcontroldata 00000000

Output of the switching point states: (see *Tab. 113*)

11.14.2 Stop measured value output for Level Control

The measured value output is stopped using a telegram.

Request

Telegram structure: **sMN mLRstoplevelcontroldata**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Request (SOPAS method by name)	string	3	**sMN**
Command	Stop measured value output	string	23	**mLRstoplevelcontroldata**

Telegram syntax 114: Request “Stop measured value output”

Confirmation

Telegram structure: **sMA mLRstoplevelcontroldata**

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Confirmation (SOPAS method acknowledge)	string	3	**sMA**
Command	Stop measured value output	string	23	**mLRstoplevelcontroldata**

Telegram syntax 115: Confirmation of the “Stop measured value output” request

Answer

Telegram structure: **sAN mLRstoplevelcontroldata** ErrorCode

Telegram part	Description	Variable type	Length (byte)	Value range
Type of command	Answer (SOPAS answer)	string	3	**sAN**
Command	Stop measured value output	string	23	**mLRstoplevelcontroldata**
ErrorCode	The command has been accepted if the error code 0 is returned.	uint_32	4	**00000000h** no error ... **FFFFFFFFh** ErrorCode

Telegram syntax 116: Answer to the “Stop measured value output” request

Example

Request: sMN mLRstoplevelcontroldata

Confirmation: sMA mLRstoplevelcontroldata

Answer: sAN mLRstoplevelcontroldata 00000000

11.15 Ordering information

Ordering information about the LMS400 and its optional accessories is available from the following web product page:

➢ *www.sick.com/lms4xx*

11.16 EU Declaration of Conformity

The EU declaration of conformity of the LMS400 with the listed available device variants (according to the type code) and the fullfiled standards is availabe in PDF format from the following product web page:

➢ *www.sick.com/lms4xx*

11.17 Glossary

Note For other terms, see also the online help for the SOPAS ET configuration software.

Aux interface

Auxiliary data interface (RS-232) on the LMS400 with fixed data output format. Using this interface it is always possible to access the LMS400 using the SOPAS ET configuration software. Is used, among other tasks, for the output of system messages and error messages. Can be used with various functions.

Download

Transmission of the parameter set that has been modified offline in the SOPAS ET configuration software from the PC to the LMS400. SOPAS ET transmits either always a complete copy to the memory (RAM) in the LMS400 (menu COMMUNICATION, DOWNLOAD ALL PARAMETERS TO DEVICE) or only the parameter that has just been edited (menu COMMUNICATION, DOWNLOAD MODIFIED PARAMETERS TO DEVICE). With the menu LMS_XX00, PARAMETER, SAVE PERMANENT, the parameter set is saved permanently in the EEPROM of the LMS400.

Field of view α

Angle that defines the limits to which the laser beam is deflected by the polygon mirror wheel. A v-shaped area is formed radially in the scan direction in front of the laser output aperture; this area must contain the objects to be measured.

Host interface

Primary data interface for the LMS400 with data output format that can be configured. Is used, among other tasks, for the output of the measuring result in telegram format to the host/the PLC. Used to integrate the LMS400 in the SICK network. Can be connected electrically as RS-232 or RS-422. Provides various transmission protocols.

Line scanner

Scanner that very rapidly deflects its focused laser beam with the aid of a polygon mirror wheel with mirrors parallel to the axis. In this way the scanner generates a dot of light in the measuring plane that repeatedly runs along a straight line and appears to be a “stationary” scan line due to the relative slow response of the human eye.

Parameter set

Data set using which the functions implemented in the LMS400 are initialised and activated. Is transmitted from the LMS400 to SOPAS ET and in the reverse direction using UPLOAD or DOWNLOAD respectively.

Remission

Remission is the quality of reflection at a surface. The basis is the Kodak standard, known worldwide in, among other areas, photography.

RIS

Remission Information System: The RIS value corresponds to the remission value without application of the scaling factor. It states the reflectivity of the object at the measurement point in percent determined by the system. A small RIS value signifies a low reflectivity (as a rule a dark object). Only distance values with RIS values ≥5 (= 10%) are inside the specified range of the LMS400. With RIS values <5 the reliability of the measured result is low.

Scan

A scan encompasses all measured values determined referred to the scanning angle and the speed of rotation of the mirror.

Scan line

See line scanner.

SOPAS ET

Configuration software, can be used with Windows 98/NT 4.0/2000/XP/Vista/7. Is used for the offline configuration (adaptation to the read situation on-site) and the online operation of the LMS400 in the dialog box.

Upload

Transmission of the parameter set from the LMS400 to the PC into the SOPAS ET configuration software. The values for the parameters are displayed on the file cards of the configuration software. Prerequisite for the modification of the current parameter set.

8016153/ZM55/2017-05-05 · 8M_DR · A4 4c int47

Australia
Phone +61 3 9457 0600
1800 334 802 – tollfree
E-Mail sales@sick.com.au

Austria
Phone +43 22 36 62 28 8-0
E-Mail office@sick.at

Belgium/Luxembourg
Phone +32 2 466 55 66
E-Mail info@sick.be

Brazil
Phone +55 11 3215-4900
E-Mail marketing@sick.com.br

Canada
Phone +1 905 771 14 44
E-Mail information@sick.com

Czech Republic
Phone +420 2 57 91 18 50
E-Mail sick@sick.cz

Chile
Phone +56 2 2274 7430
E-Mail info@schadler.com

China
Phone +86 20 2882 3600
E-Mail info.china@sick.net.cn

Denmark
Phone +45 45 82 64 00
E-Mail sick@sick.dk

Finland
Phone +358-9-2515 800
E-Mail sick@sick.fi

France
Phone +33 1 64 62 35 00
E-Mail info@sick.fr

Germany
Phone +49 211 5301-301
E-Mail info@sick.de

Hong Kong
Phone +852 2153 6300
E-Mail ghk@sick.com.hk

Hungary
Phone +36 1 371 2680
E-Mail office@sick.hu

India
Phone +91 22 6119 8900
E-Mail info@sick-india.com

Israel
Phone +972 4 6881000
E-Mail info@sick-sensors.com

Italy
Phone +39 02 274341
E-Mail info@sick.it

Japan
Phone +81 3 5309 2112
E-Mail support@sick.jp

Malaysia
Phone +6 03 8080 7425
E-Mail enquiry.my@sick.com

Mexico
Phone +52 (472) 748 9451
E-Mail mario.garcia@sick.com

Netherlands
Phone +31 30 2044 000
E-Mail info@sick.nl

New Zealand
Phone +64 9 415 0459
0800 222 278 – tollfree
E-Mail sales@sick.co.nz

Norway
Phone +47 67 81 50 00
E-Mail sick@sick.no

Poland
Phone +48 22 539 41 00
E-Mail info@sick.pl

Romania
Phone +40 356 171 120
E-Mail office@sick.ro

Russia
Phone +7 495 775 05 30
E-Mail info@sick.ru

Singapore
Phone +65 6744 3732
E-Mail sales.gsg@sick.com

Slovakia
Phone +421 482 901201
E-Mail mail@sick-sk.sk

Slovenia
Phone +386 591 788 49
E-Mail office@sick.si

South Africa
Phone +27 11 472 3733
E-Mail info@sickautomation.co.za

South Korea
Phone +82 2 786 6321
E-Mail info@sickkorea.net

Spain
Phone +34 93 480 31 00
E-Mail info@sick.es

Sweden
Phone +46 10 110 10 00
E-Mail info@sick.se

Switzerland
Phone +41 41 619 29 39
E-Mail contact@sick.ch

Taiwan
Phone +886 2 2375-6288
E-Mail sales@sick.com.tw

Thailand
Phone +66 2645 0009
E-Mail Ronnie.Lim@sick.com

Turkey
Phone +90 216 528 50 00
E-Mail info@sick.com.tr

United Arab Emirates
Phone +971 4 88 65 878
E-Mail info@sick.ae

United Kingdom
Phone +44 1727 831121
E-Mail info@sick.co.uk

USA
Phone +1 800 325 7425
E-Mail info@sick.com

Vietnam
Phone +84 945452999
E-Mail Ngo.Duy.Linh@sick.com

Further locations at **www.sick.com**

SICK
Sensor Intelligence.