SICK.COM



OPERATING INSTRUCTIONS

LMS1xx

2D LiDAR sensor

SICK Sensor Intelligence

Described product

LMS1xx

Manufacturer

SICK AG
Erwin-Sick-Str. 1
79183 Waldkirch
Germany

Legal information

This work is protected by copyright. Any rights derived from the copyright shall be reserved for SICK AG. Reproduction of this document or parts of this document is only permissible within the limits of the legal determination of Copyright Law. Any modification, abridgment or translation of this document is prohibited without the express written permission of SICK AG.

The trademarks stated in this document are the property of their respective owner.

© SICK AG. All rights reserved.

Original document

This document is an original document of SICK AG.

Contents

1 About this document ... 6
1.1 Information on the operating instructions ... 6
1.2 Scope ... 6
1.3 Explanation of symbols ... 6
1.4 Further information ... 7

2 Safety information ... 8
2.1 Intended use ... 8
2.2 Improper use ... 8
2.3 Cybersecurity ... 8
2.4 Limitation of liability ... 9
2.5 Modifications and conversions ... 9
2.6 Requirements for skilled persons and operating personnel ... 9
2.7 Operational safety and specific hazards ... 10

3 Product description ... 11
3.1 Scope of delivery ... 11
3.2 Device variants ... 11
3.2.1 Outdoor variants with ventilation element ... 13
3.3 Type code ... 14
3.4 Type label ... 15
3.5 Status indicators ... 16
3.6 Principle of operation ... 17
3.6.1 Measurement principle ... 17
3.6.2 Distance measurement ... 18
3.6.3 Direction measurement ... 18
3.6.4 Impact of object surfaces on the measurement ... 18
3.6.5 Scanning range ... 20
3.6.6 Beam diameter and measuring point distance ... 22
3.6.7 Minimum object size ... 25
3.6.8 Contamination measurement ... 26
3.6.9 Calculation of the field size for mobile applications ... 27
3.7 Fields of application ... 29
3.7.1 Object measurement ... 29
3.7.2 Field evaluation ... 37
3.8 Inputs and outputs ... 42
3.9 Data interfaces ... 44
3.10 Communicating data via telegrams ... 46

4 Transport and storage ... 47
4.1 Transport ... 47
4.2 Unpacking ... 47
4.3 Transport inspection ... 47
4.4 Storage ... 47

5 Mounting 48
5.1 Mounting instructions 48
5.2 Mounting the device 48
5.3 Mounting multiple devices 48

6 Electrical installation 50
6.1 Wiring instructions 50
6.2 Overview of the installation steps 50
6.3 Prerequisites for safe operation of the device 51
6.4 Connections 55
6.4.1 Connections of the LMS10x 55
6.4.2 Connections of the LMS11x, LMS15x and LMS16x 57
6.5 Preparing electrical installation 58
6.5.1 SELV supply voltage 58
6.5.2 Conductor cross-sections 59
6.5.3 General Requirements of the Data Interfaces 60
6.5.4 Cable reserve on system plug 60
6.5.5 General conditions for data interfaces 60
6.6 Performing the electrical installation 61
6.6.1 Connection to AUX and to the Ethernet interface of the device 61
6.6.2 LMS10x/LMS12x/LMS173/LMC12x: Wiring the system plug 61
6.6.3 LMS11x/LMS13x/LMS15x/LMS16x/LMC13x: Connecting M12 round connectors 62
6.6.4 Wiring the inputs and outputs 62

7 Commissioning 66
7.1 Overview of the commissioning steps 66
7.2 SOPAS ET configuration software 66
7.3 Establishing communication with the device 67
7.4 Initial commissioning 68
7.5 Connecting and performing a test measurement 70

8 Maintenance 71
8.1 Cleaning 71
8.2 Maintenance plan 71
8.3 Device replacement 71
8.3.1 Parameter cloning (LMS10x/LMS12x/LMS17x) 72
8.3.2 Replacing the device 73

9 Troubleshooting 75
9.1 Repairs 75
9.2 Returns 75
9.3 General faults, warnings, and errors 75
9.3.1 LED fault indicators 75
9.3.2 Displays in the 7-segment display 76
9.3.3 Detailed fault analysis 76

9.3.4 Resetting the password for the **Service** user level ... 77
9.4 Disposal ... 78

10 Technical data ... 79
10.1 Features ... 79
10.2 Mechanics/Electronics ... 83
10.3 Performance ... 84
10.4 Interfaces ... 84
10.5 Ambient data ... 85
10.6 Dimensional drawings ... 85

11 Accessories ... 87

12 Annex ... 88
12.1 Declarations of conformity and certificates ... 88
12.2 Telegram listing (EN) ... 88

1 About this document

1.1 Information on the operating instructions

These operating instructions provide important information on how to use devices from SICK AG.

Prerequisites for safe work are:

- Compliance with all safety notes and handling instructions supplied.
- Compliance with local work safety regulations and general safety regulations for device applications

The operating instructions are intended to be used by qualified personnel and electrical specialists.

NOTE
Read these operating instructions carefully to familiarize yourself with the device and its functions before commencing any work.

The operating instructions are an integral part of the product. Store the instructions in the immediate vicinity of the device so they remain accessible to staff at all times. Should the device be passed on to a third party, these operating instructions should be handed over with it.

These operating instructions do not provide information on the handling and safe operation of the machine or system in which the device is integrated. Information on this can be found in the operating instructions for the machine or system.

1.2 Scope

The operating instructions are valid for the following variants:

Table 1: Variants of the LMS1xx

LMS1xx variants
LMS100
LMS111
LMS151/LMS153
LMS161
LMS121 / LMS122 / LMS123 / LMS173 Security, additional information in the Technical Information (Security Installer Instructions), Part no.: 8014486
LMS131 / LMS132 / LMS133 / LMS141 core/LMS141 prime
LMC131/LMC132/LMC133 VdS, additional information in the Technical Information (VdS Installer Instructions), part no.: 8013748

Simplified device designation in the document

In the following instructions, the LMS1xx 2D LiDAR sensor is referred to simply as “device” or “LMS1xx”. Except in cases where it is necessary to make a distinction between device variants due to different technical features or functions. In this case the complete type designation according to the type code (e.g., LMS10x) is used.

1.3 Explanation of symbols

Warnings and important information in this document are labeled with symbols. Signal words introduce the instructions and indicate the extent of the hazard. To avoid accidents, damage, and personal injury, always comply with the instructions and act carefully.

DANGER
... indicates a situation of imminent danger, which will lead to a fatality or serious injuries if not prevented.

WARNING
... indicates a potentially dangerous situation, which may lead to a fatality or serious injuries if not prevented.

CAUTION
... indicates a potentially dangerous situation, which may lead to minor/slight injuries if not prevented.

NOTICE
... indicates a potentially harmful situation, which may lead to material damage if not prevented.

NOTE
... highlights useful tips and recommendations as well as information for efficient and trouble-free operation.

1.4 Further information

More information can be found on the product page.

The product page can be accessed via the **SICK Product ID**: **pid.sick.com/{P/N}/{S/N}**

{P/N} corresponds to the part number of the product, see type label.

{S/N} corresponds to the serial number of the product, see type label (if indicated).

The following information is available depending on the product:

- Data sheets
- This document in all available language versions
- CAD files and dimensional drawings
- Certificates (e.g., declaration of conformity)
- Other publications
- Software
- Accessories

2 Safety information

2.1 Intended use

The LMS1xx is a non-contact optical distance measurement sensor in standalone or network operation based on a 2D-LiDAR sensor. It is suitable for applications which demand precise, non-contact optical measuring contours and environment perception. It can also be used to implement systems for collision protection, object protection or access monitoring, for example.

The device may only be put into operation by authorized staff and only in industrial environments.

SICK AG assumes no liability for losses or damage arising from the use of the product, either directly or indirectly. This applies in particular to use of the product that does not conform to its intended purpose and is not described in this documentation.

2.2 Improper use

Any use outside of the stated areas, in particular use outside of the technical specifications and the requirements for intended use, will be deemed to be incorrect use.

- The device does not constitute a safety component in accordance with the respective applicable safety standards for machines.
- The device must not be used in explosion-hazardous areas, in corrosive environments or under extreme environmental conditions.
- Any use of accessories not specifically approved by SICK AG is at your own risk.

WARNING
Danger due to improper use!

Any improper use can result in dangerous situations.

Therefore, observe the following information:

- Product should be used only in accordance with its intended use.
- All information in the documentation must be strictly observed.
- Shut down the product immediately in case of damage.

2.3 Cybersecurity

Overview

To protect against cybersecurity threats, the operator must have a comprehensive cybersecurity concept, which must be continuously monitored and maintained. A suitable concept consists of organizational, technical, procedural, electronic, and physical levels of defense and considers suitable measures for different types of risks. The measures implemented in this product can only support protection against cybersecurity threats if the product is used as part of such a concept.

You will find further information at www.sick.com/psirt, e.g.:

- General information on cybersecurity
- Contact option for reporting vulnerabilities
- Information on known vulnerabilities (security advisories)

2.4 Limitation of liability

Relevant standards and regulations, the latest technological developments, and our many years of knowledge and experience have all been taken into account when compiling the data and information contained in these operating instructions. The manufacturer accepts no liability for damage caused by:

- Non-adherence to the product documentation (e.g., operating instructions)
- Incorrect use
- Use of untrained staff
- Unauthorized conversions or repair
- Technical modifications
- Use of unauthorized spare parts, consumables, and accessories

2.5 Modifications and conversions

NOTICE
Modifications and conversions to the device may result in unforeseeable dangers.

Interrupting or modifying the device or SICK software will invalidate any warranty claims against SICK AG. This applies in particular to opening the housing, even as part of mounting and electrical installation.

2.6 Requirements for skilled persons and operating personnel

WARNING
Risk of injury due to insufficient training.
Improper handling of the device may result in considerable personal injury and material damage.

- All work must only ever be carried out by the stipulated persons.

The following qualifications are required for various activities:

Table 2: Activities and technical requirements

Activities	Qualification
Mounting, maintenance	• Basic practical technical training • Knowledge of the current safety regulations in the workplace
Electrical installation, device replacement	• Practical electrical training • Knowledge of current electrical safety regulations • Knowledge of the operation and control of the devices in their particular application
Commissioning, configuration	• Basic knowledge of the computer operating system used • Basic knowledge of the design and setup of the described connections and interfaces • Basic knowledge of data transmission
Operation of the device for the particular application	• Knowledge of the operation and control of the devices in their particular application • Knowledge of the software and hardware environment for the particular application

2.7 Operational safety and specific hazards



CAUTION
Optical radiation: Class 1 Laser Product

The accessible radiation does not pose a danger when viewed directly for up to 100 seconds. It may pose a danger to the eyes and skin in the event of incorrect use.

- Do not open the housing. Opening the housing may increase the level of risk.
- Current national regulations regarding laser protection must be observed.

Caution – Use of controls or adjustments or performance of procedures other than those specified herein may result in hazardous radiation exposure.

It is not possible to entirely rule out temporary disorienting optical effects, particularly in conditions of dim lighting. Disorienting optical effects may come in the form of dazzle, flash blindness, afterimages, photosensitive epilepsy, or impairment of color vision, for example.



WARNING
Electrical voltage!

Electrical voltage can cause severe injury or death.

- Work on electrical systems must only be performed by qualified electricians.
- The power supply must be disconnected when attaching and detaching electrical connections.
- The product must only be connected to a voltage supply as set out in the requirements in the operating instructions.
- National and regional regulations must be complied with.
- Safety requirements relating to work on electrical systems must be complied with.



WARNING
Risk of injury and damage caused by potential equalization currents!

Improper grounding can lead to dangerous equipotential bonding currents, which may in turn lead to dangerous voltages on metallic surfaces, such as the housing. Electrical voltage can cause severe injury or death.

- Work on electrical systems must only be performed by qualified electricians.
- Follow the notes in the operating instructions.
- Install the grounding for the product and the system in accordance with national and regional regulations.

3 Product description

3.1 Scope of delivery

The delivery of the device includes the following components:

Table 3: Scope of delivery

No. of units	Components	Remarks
1	Device in the version ordered	Type depends on order Put plastic protective caps on all M12 round connectors and cable entries.
1	LMS: Printed safety notes, multilingual LMC: Printed device information, multilingual	Brief information and general safety notes

3.2 Device variants

Table 4: Device variants

Type	Purpose	Heating	Ventilation element	Enclosure rating
LMS10x	Indoor variant	No	No	IP65
LMS11x	Outdoor variant	Yes	Yes	IP67 [1]
LMS15x	Outdoor variant with extended functionality	Yes	Yes	IP67 [1]
LMS16x	Outdoor variant	Yes	Yes	IP67 [1]
LMS12x LMC12x LMS173	Indoor variant, optimized for use in object protection systems	No	No	IP65
LMS13x LMS14x	Outdoor variant, optimized for use in object protection systems	Yes	Yes	IP67 [1]
LMC13x	Semi-outdoor variant, optimized for use in object protection systems	Yes	Yes	IP67 [1]

1) Under the following conditions: The SICK cables attached to the M12 connections must be screwed tight. Any electrical connections that are not being used must be fitted with protective caps or plugs that are screwed tight (as in the delivery condition). The M8 connection for the AUX interface on the front of the device is fastened.

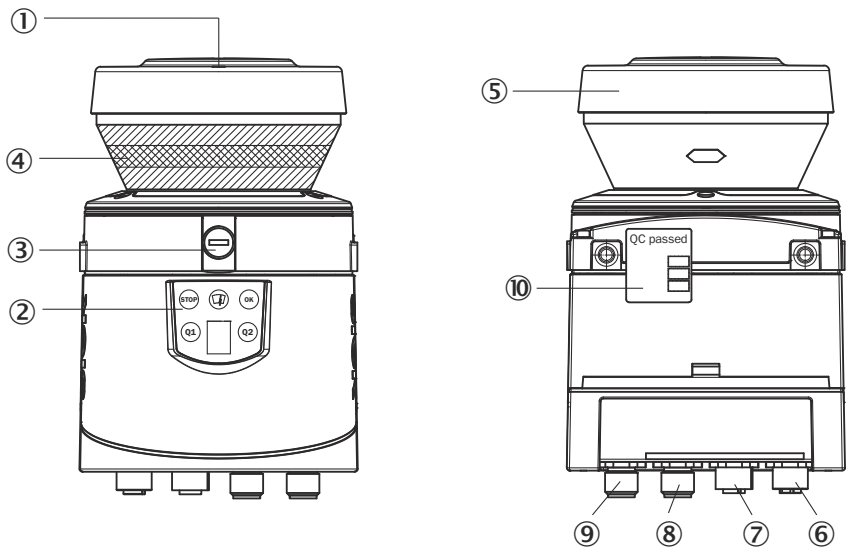


Figure 1: View of the LMS11x/LMS15x/LMS16x

① Mark of measurement origin

② Display elements

③ AUX connection (behind cover)

④ Viewing window (laser output aperture)

⑤ Optics cover

⑥ “Ethernet” connection

⑦ "I/O" connection (for LMS1xx Security Outdoor and LMS1xx VdS Semi Outdoor: "Inputs" connection with different assignment)

⑧ “Data” connection (for LMS1xx Security Outdoor and LMS1xx VdS Semi Outdoor: “Alarm" connection with different assignment)

⑨ “Power” connection

⑩ Quality assurance seal

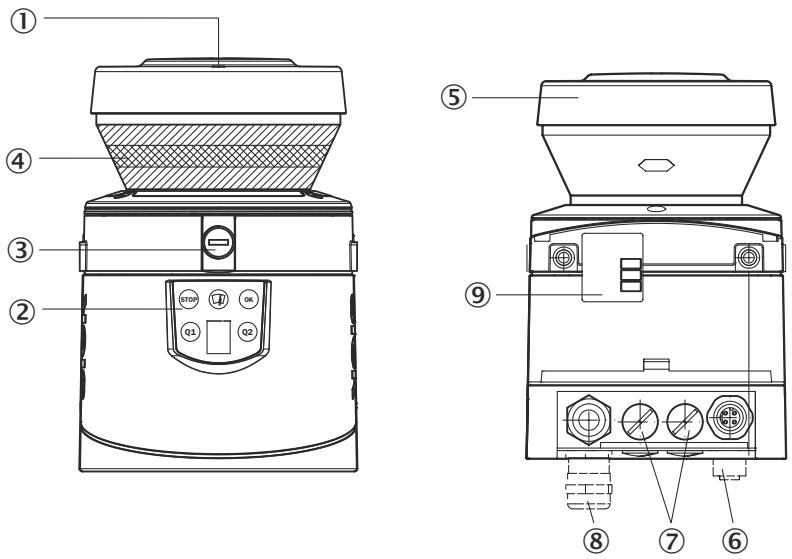
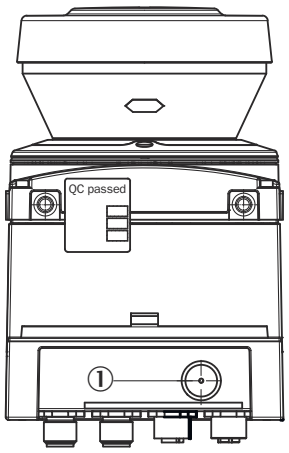


Figure 2: View of the LMS10x

① Mark of measurement origin
② Display elements
③ AUX connection (behind cover)
④ Viewing window (laser output aperture)
⑤ Optics cover
⑥ “Ethernet” connection
⑦ Blind plug
⑧ Cable entry
⑨ Quality assurance seal

3.2.1 Outdoor variants with ventilation element

The ventilation element ensures an improved pressure equalization and allows the exchange of air and heat between the sensor housing and the environment. The breathable membrane allows ambient air to either penetrate into the device, or escape again depending on the prevailing ambient conditions. In particular for applications with frequently changing environmental influences (e.g., large temperature fluctuations or fast temperature changes) or with standing water, the membrane ensures a reliable pressure equalization and thereby relieves the seals and adhesive joints of the sensor housing. This can improve the expected service life of the device in the application.



① Ventilation element

Notes and recommendations:

- Do not affix any labels or stickers to the ventilation element or paint over the membrane. Longer moist periods and rapid temperature changes may briefly reduce the availability of the device. This applies in particular to previously switched off devices that are started in the application and were previously exposed to the aforementioned environmental influences.
- In some circumstances, therefore, a period of time should be allowed before measurement readiness of the device because any moisture in the housing must first be taken up by the air in the housing, which is heated up through the operation of the device, so that it can then escape via the ventilation element. Depending on the nature of the precipitated moisture, this time period might be several minutes or even up to hours. It is recommended that the LMS1xx variants with a ventilation element be operated continuously during the application (continuous operation).
- It is recommended to connect the integrated sensor heating according to the specifications in the operating instructions to ensure additional heat input at low ambient temperatures.

3.3 Type code

Structure of LMS1xx type code

LMS a b c - dd e

Table 5: Type code

Position	Description	Characteristic
a	Device family	1: LMS1xx
b	Type	0: Indoor housing without heating, IP65 1: Outdoor housing with heating, IP65, IP67 2: Security outdoor housing with heating, IP65 3: Security outdoor housing with heating, IP65, IP67 4: Security outdoor housing with heating, IP67, scanning range max. 40 m, high sensitivity 5: Outdoor housing with heating, IP65, IP67, scanning range max. 50 m 6: Outdoor housing with heating, IP67, scanning range max. 50 m, high sensitivity 7: Security outdoor housing with heating, IP65, special 8: Security outdoor housing with heating, IP65, IP67, special
c	Color	0: Blue 1: Gray 2: Black 3: White

Position	Description	Characteristic
–		
dd	Performance	05: Core, with extended ambient temperature range -40 °C ... + 60 °C 10: Standard/Prime 15: Standard/Prime, with extended ambient temperature range -40 °C ... + 60 °C
e	Connection technology	0: System plug/PG 1: M12 standard

Structure of LMC type code

LMC a b c - dd e ff

Table 6: Type code

Position	Description	Characteristic
a	Device family	1: LMC1xx
b	Type	0: Security indoor housing without heating, IP65 1: Security outdoor housing with heating, IP65, IP67
c	Color	1: Gray 2: Black 3: White 4: Black (device) / mounting kit unpainted
–		
dd	Performance	11: VdS certified
e	Connection technology	0: System plug/PG 1: M12 standard
ff	Mounting types	00: Mounting kit VdS 1 90: Mounting kit VdS 2

3.4 Type label

The type label gives information for identification of the product variant.

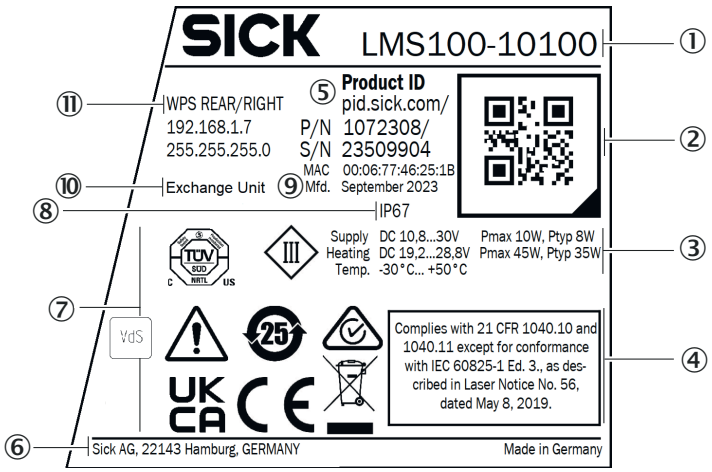


Figure 3: Elements of the type label (example)

① Type code

② Data Matrix code with product data and link to product page

③ Voltage supply to device, typical power, max. power/voltage supply to heater, typical power, max. power/temperature range

④ Laser Notice

⑤ Product ID with part number (P/N) and serial number (S/N)

⑥ Manufacturer and production site

⑦ Conformity mark/certification mark, symbol: Observe the operating instructions!

⑧ Enclosure rating

⑨ Production date

⑩ Replacement device

⑪ WPS REAR/RIGHT

3.5 Status indicators

Operator interface

The device works fully automatically in normal operation and requires no operator intervention.

The SOPAS ET configuration software allows interactive configuration. For this, the software runs on a computer which is connected to the device with one of the data interfaces.

The graphical scan display in SOPAS ET is used to verify the generated measured values and the measuring range online.

NOTE

Scans in SOPAS ET are not displayed in real time, but at a limited speed. Not all measured values are displayed for this reason.

Status indicators

The LEDs and 7-segment display indicate the operational status of the device.



Display	Description
①	Device is either not in measuring mode (stopped by the user) or it is running in measurement mode, but errors have occurred
②	Off: No contamination On (lit up) Contamination warning On (flashing): Contamination error
③	Device running in measurement mode without any errors
④	Device in teach-in mode
⑤	The 7-segment display is used for diagnostics when errors or faults occur see "Displays in the 7-segment display", page 76.
⑥	At least one field is breached see "Linking of evaluation cases at the output", page 42.

NOTE
As well as the standard indicators described below, the indicator functions of the LEDs and the 7-segment display can be configured in SOPAS ET for the device (SOPAS ET: **Project tree**, **LMS...**, **Parameters**, **Network/Interfaces/IOs**, **Display**).

For additional information, see "LED fault indicators", page 75.

NOTE
LMC12x/LMC13x VdS (object protection) Further information on the display elements of these devices is provided in the document “Technical Information (VdS Installer Instructions)“, Part no.: 8013748

LMS12x/LMS13x/LMS14x Security Further information on the display elements of these devices is provided in the document “Technical Information (Security Installer Instructions)“, Part no.: 8014486

3.6 Principle of operation

3.6.1 Measurement principle

The device is an opto-electronic LiDAR sensor that scans the outline of its surroundings with the help of laser beams without making contact. The device measures its surroundings in two-dimensional polar coordinates, relative to its measurement origin. This is indicated by small markings on the sides of the optics cover. If a laser beam strikes an object, the position of that object is determined in terms of distance and angle.

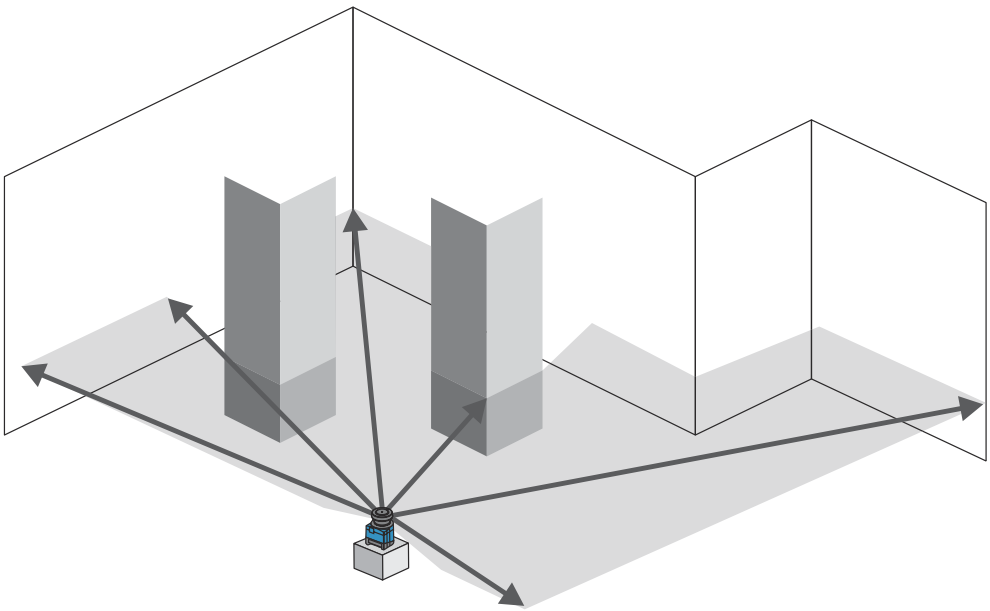
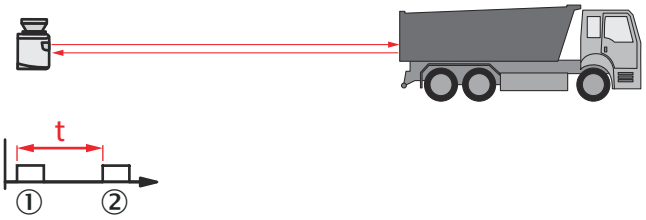


Figure 4: LiDAR sensor with one scan plane

3.6.2 Distance measurement

The device emits beams pulsed by a laser diode. If the laser beam is reflected by an object, the reflected beam is received by the sensor.

The distance to the object is calculated on the basis of the time that the pulsed light beam requires to be reflected and received by the sensor.



① Emitted pulse

② Receive pulse

3.6.3 Direction measurement

The emitted laser beams are deflected by an internal rotating mirror and scan the environment in a circular motion. The measurements are triggered by an encoder in regular angle increments.

The device scans with an adjustable scanning frequency of 25 Hz or 50 Hz. A laser pulse and thus a measurement is triggered continuously after each angle increment of 0.25° or 0.50°.

3.6.4 Impact of object surfaces on the measurement

The received signal from a perfectly diffuse reflection from a white surface (diffuse Lambertian reflector) corresponds to a remission of 100%. By this definition, surfaces that reflect the light in bundles (specular surfaces, reflectors) have remissions of over 100%.

Reflection

Most surfaces produce a diffuse reflection of the laser beam in all directions. The structure (smooth or rough), shape (flat or curved), and color (light or dark) of the surface determine how well the laser beam is reflected.

On very rough surfaces, a large proportion of the energy is lost due to absorption. Curved surfaces produce a higher diffusion. Dark surfaces reflect the laser beam worse than light ones (brilliant white plaster reflects approx. 100% of the light, while black foam rubber reflects approx. 2.4%). The aforementioned surface characteristics can reduce the scanning range of the device, in particular for surfaces with low remission values.

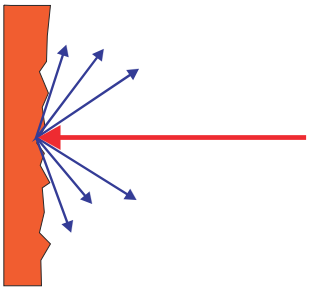


Figure 5: Reflection of light on the surface of the object

Angle of reflection

The angle of reflection corresponds to the angle of incidence. If the laser beam hits a surface at right angles, the energy is optimally reflected. If the laser beam hits a surface at an oblique angle, energy and range are lost accordingly.

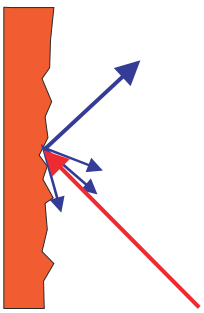


Figure 6: Angle of reflection

Retroreflection

If the reflective energy is greater than 100%, the beam is not reflected diffusely in all directions; instead it is reflected in a targeted way (retroreflection). Thus a large part of the emitted energy can be received by the laser distance measurer. Plastic reflectors (cat's eyes), reflective tape, and triple prisms have these properties.

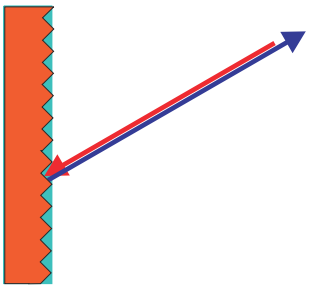


Figure 7: Retroreflection

Reflective surfaces

The laser beam is almost completely deflected on reflective surfaces. This means that an object hit by the deflected beam may be detected instead of the reflective surface.

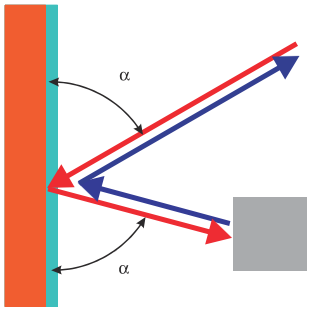


Figure 8: Specular surfaces

Small objects

Objects that are smaller than the diameter of the laser beam cannot reflect the laser light's full energy. The portion of the light beam that does not reach the object is lost. If all of the light reflected to the sensor is insufficient, the object may not be detected.

The portion of the light that does not reach the front object can be reflected by a larger object in the background. If all of the light reflected to the sensor is sufficient, this object is detected. This can lead to a corruption of the measured value.

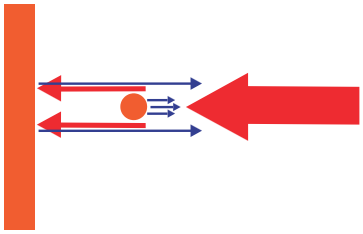


Figure 9: Object smaller than the laser beam diameter

3.6.5 Scanning range

The scanning range of the device depends on the remission of the object to be detected. The better a surface reflects the incident beam back to the device, the greater the scanning range of the device.

NOTE
The following diagrams are only valid if no filters are configured.

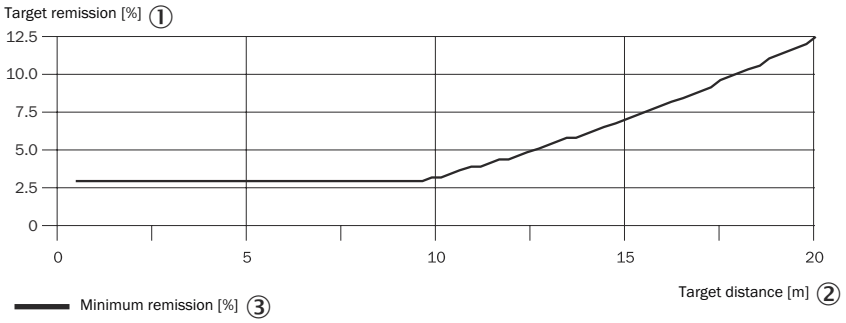


Figure 10: Scanning range LMS10x/LMS11x/LMS12x/LMS13x/LMS173 and LMC12x/LMC13x

① Remission factor of the target [%]

② Distance of the target [m]

③ Minimum remission factor [%]

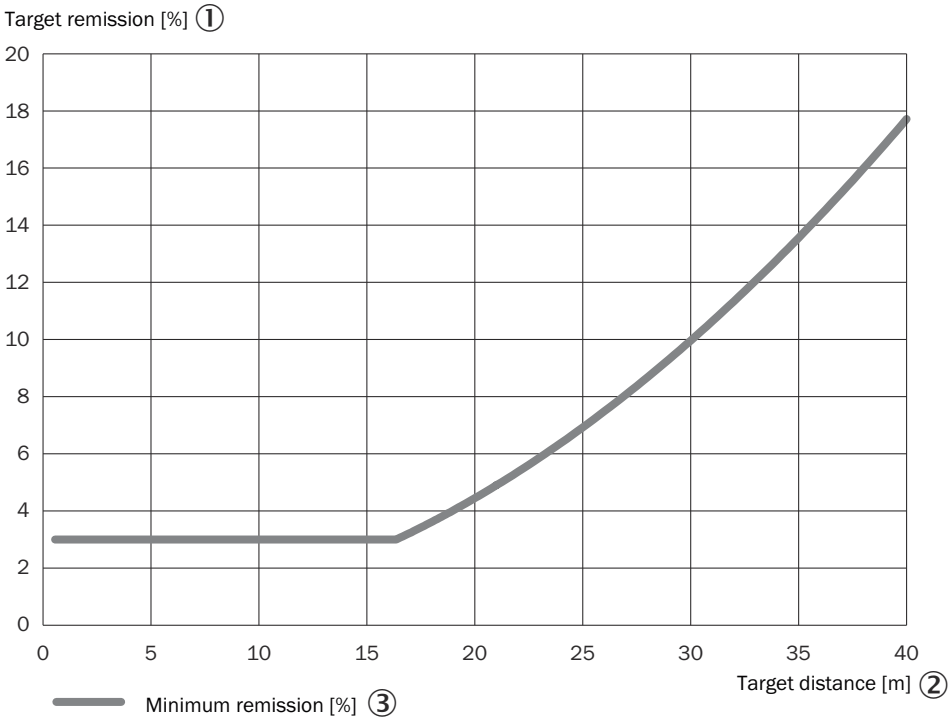


Figure 11: LMS14x/LMS16x: Scanning range depends on the remission factor of the target

① Remission factor of the target [%]

② Distance of the target [m]

③ Minimum remission factor [%]

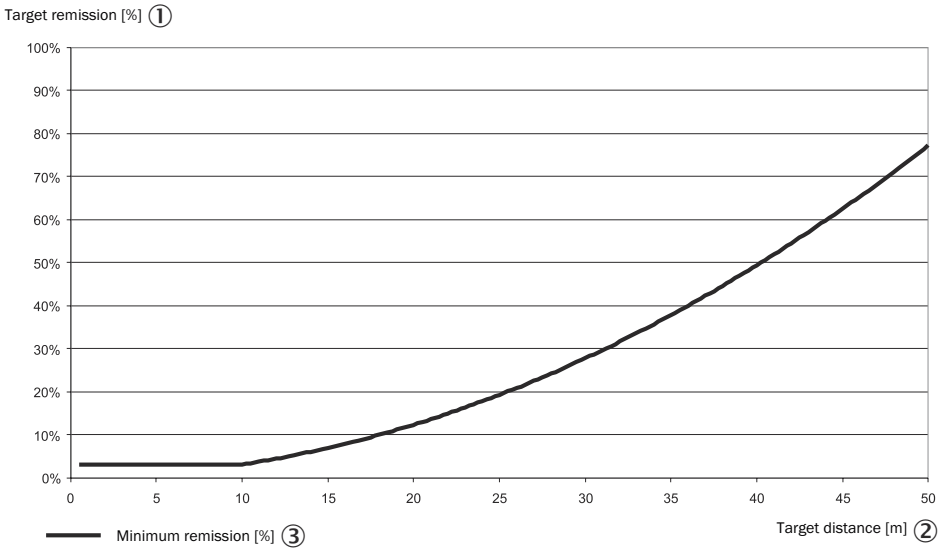


Figure 12: LMS15x: Scanning range depending on the remission factor of the target

① Remission factor of the target [%]

② Distance of the target [m]

③ Minimum remission factor [%]

The device can detect objects with a remission factor of 3% or more at a distance of up to 10 m. Up to a distance of 20 m, only objects with a remission factor of > 13% are reliably detected (see figure 10, page 21). For a remission factor of 3% or more, the LMS14x/LMS16x detects only up to 16 m (see figure 11, page 21), the LMS15x only up to 10 m (see figure 12, page 22).

3.6.6 Beam diameter and measuring point distance

As the distance from the device increases, the laser beam expands. As a result, the diameter of the measuring point increases on the surface of the object.

The range-dependent diameter of the measuring point corresponds to the distance (mm) x 0.015 rad + 8 mm.

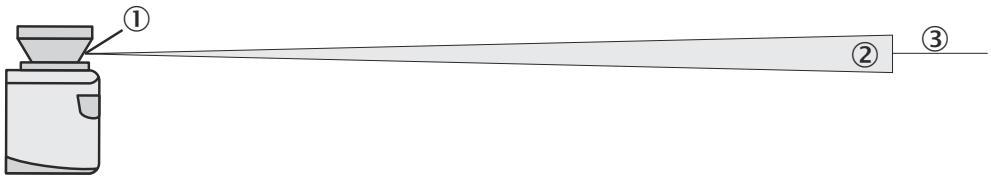


Figure 13: Beam expansion

① Beam diameter on the front screen = 8 mm

② Expanded laser beam

③ Optical axis

Similarly, as the distance from the device increases, the individual measuring points also grow further apart from one another. The distance between the measuring points also depends on the selected angular resolution. With coarser angular resolution (e.g. 0.5°), the distance is larger, with finer angular resolution (e.g. 0.25°), the distance is smaller. The range-dependent distance between the measuring points corresponds to the tangent of the angular resolution × distance.

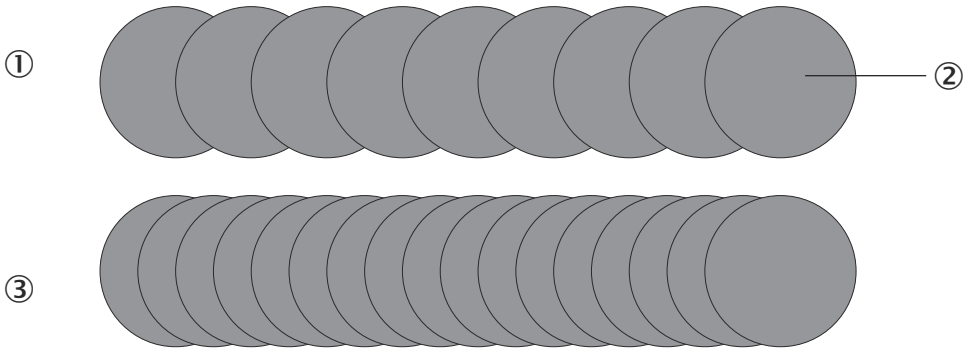


Figure 14: Schematic representation of the measuring point distance at different angular resolutions

① Scan with coarse angular resolution
② Measuring point
③ Scan with fine angular resolution

The graph in figure 15shows the beam diameter and the measuring point distance depending on the distance to the device.

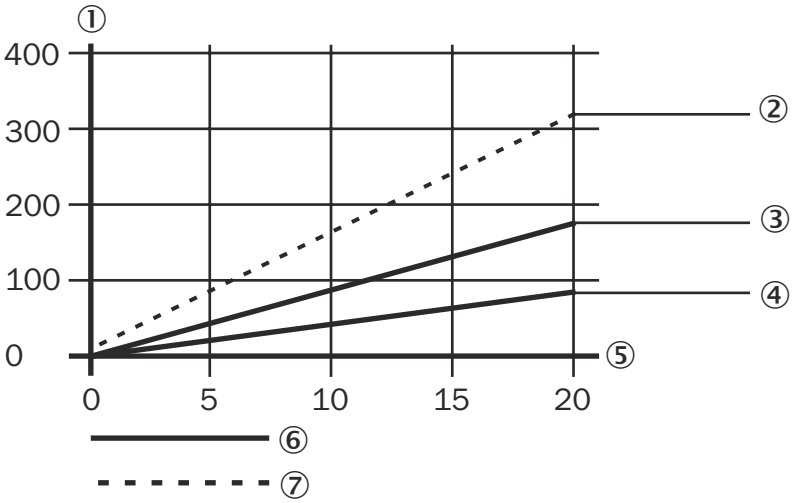


Figure 15: Beam diameter and measuring point distance at 0 to 20 m

① Size [mm]
② Beam diameter
③ 0.5° angular resolution
④ 0.25° angular resolution
⑤ Distance [m]
⑥ Measuring point distance
⑦ Beam diameter

Reading example for an angular resolution of 0.25° in figure 15

- Distance 10 m
- For a distance intersection point at 10 m, the measuring point distance is approx. 40 mm
- For a distance intersection point at 10 m and based on the characteristic curve for the beam diameter, the beam expansion is approx. 170 mm

Reading example for an angular resolution of 0.50° in figure 15

- Distance 20 m
- For a distance intersection point at 20 m, the measuring point distance is approx. 180 mm
- For a distance intersection point at 20 m and based on the characteristic curve for the beam diameter, the beam expansion is approx. 310 mm

LMS14x

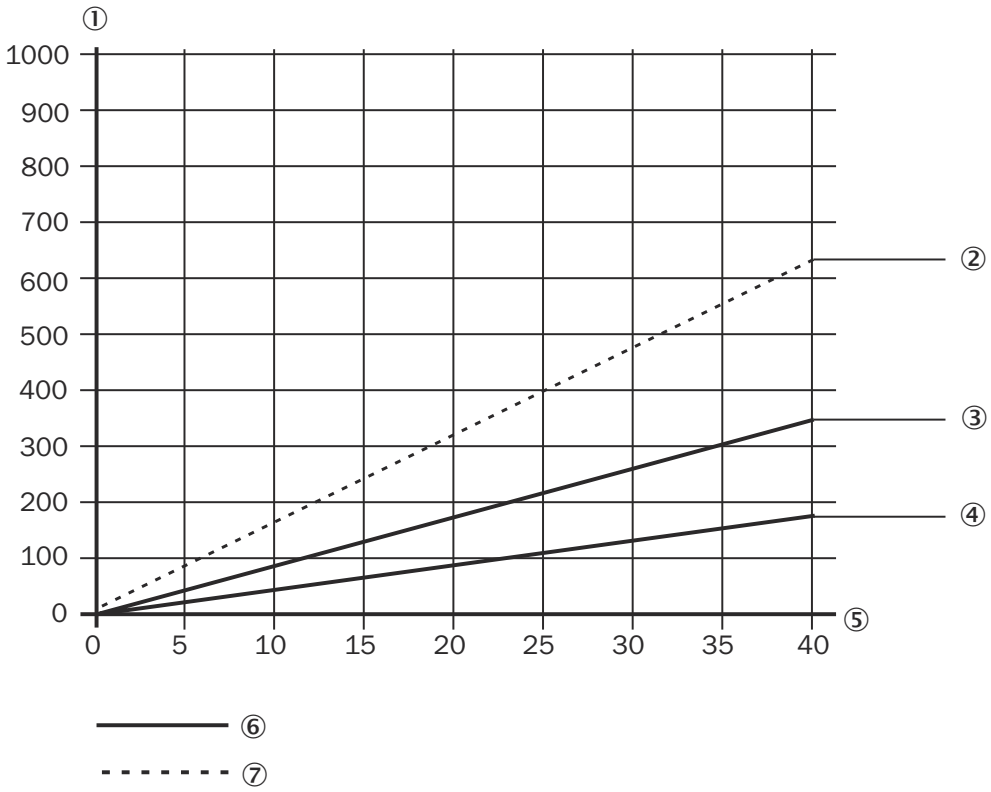


Figure 16: Beam diameter and measuring point distance at 0 to 40 m

① Size [mm]
② Beam diameter
③ 0.5° angular resolution
④ 0.25° angular resolution
⑤ Distance [m]
⑥ Measuring point distance
⑦ Beam diameter

Reading example for an angular resolution of 0.50° in figure 16

- Distance 40 m
- For a distance intersection point at 40 m, the measuring point distance is approx. 350 mm
- For a distance intersection point at 40 m and based on the characteristic curve for the beam diameter, the beam expansion is approx. 608 mm

LMS15x/LMS16x

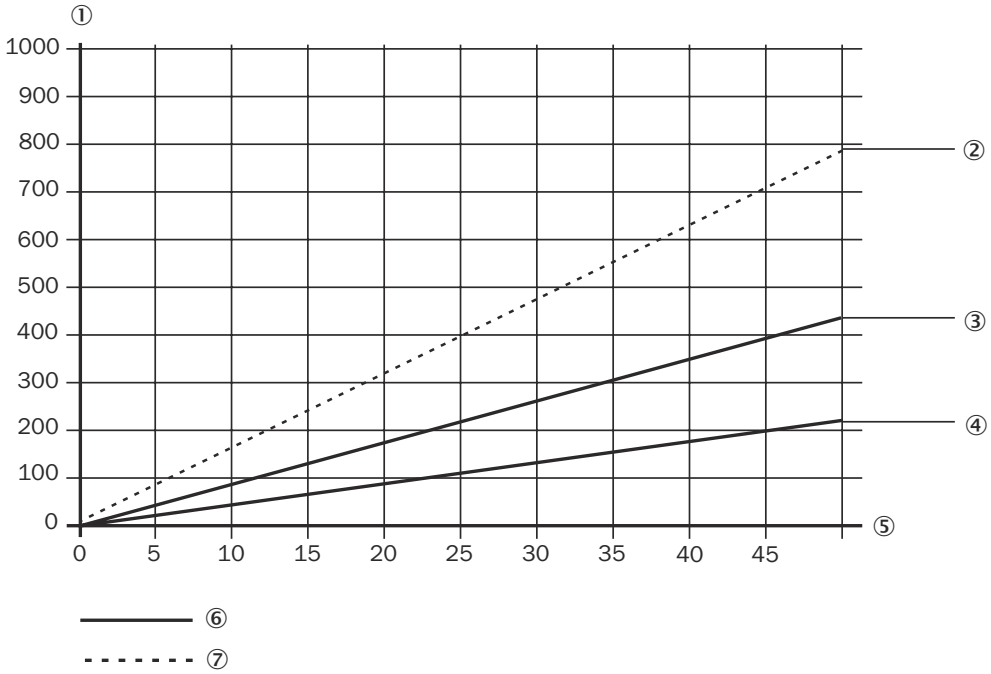


Figure 17: Beam diameter and measuring point distance at 0 to 50 m

① Size [mm]
② Beam diameter
③ 0.5° angular resolution
④ 0.25° angular resolution
⑤ Distance [m]
⑥ Measuring point distance
⑦ Beam diameter

Reading example for an angular resolution of 0.50° in figure 17

- Distance 35 m
- For a distance intersection point at 35 m, the measuring point distance is approx. 305 mm
- For a distance intersection point at 35 m and based on the characteristic curve for the beam diameter, the beam expansion is approx. 560 mm

3.6.7 Minimum object size

For an object to be detected reliably, it must be hit completely by a laser beam once. If the beam only partially hits, less energy may be reflected from the object see "Principle of operation", page 17.

An object is then reliably hit completely at any time if it is at least as large as the measuring point distance plus the beam diameter.

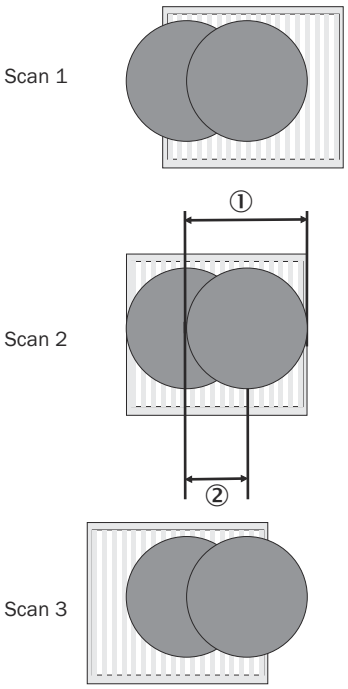


Figure 18: Minimum object size for detection

① Beam diameter

② Measuring point distance

In the example in figure 18, the object is fully hit at least once during each scan. It is therefore reliably detected if it has the necessary remission.

How to calculate the minimum object size:

Beam diameter + measuring point distance = minimum object size

- Beam diameter and measuring point distance as a function of the distance to the device can be seen in the diagram in see figure 15, page 23.

NOTE

For reliable measurement, in particular when using the device to output measured values, the laser needs to hit the object several times. Therefore, the object either needs to be larger than the minimum object size, or both the device and the object must not be moving.

3.6.8 Contamination measurement

The device has a viewing window that protects it. This viewing window can become contaminated. Contamination reduces the energy emitted and received by the laser beam. As a result, scanned objects appear to have a lower remission than they actually have and, with a certain level of contamination, it will no longer be possible to perform measurements. The device has 6 of its own contamination sensors. Depending on the selected contamination strategy, a certain number of these sensors is evaluated. The 6 contamination sensors are divided up into 2 groups (3 + 3).

The contamination is therefore constantly measured by a separate system during operation. A contamination warning is output first for the different degrees of contamination. If the viewing window is not cleaned and contamination increases, then a contamination error is output. The device turns off measurement mode.

Depending on the application in which the device is used, you can choose from different contamination measurement strategies.

SOPAS ET: **Project tree**, **LMS...**, **Parameters**, **Contamination Measurement**.

- Deactivated: Contamination measurement is not carried out.
- High availability: Contamination warning and contamination errors are only output when the viewing window is uniformly contaminated. This means that all 6 contamination sensors must have reached the limit value for a warning or error.
- Available: Contamination warning and contamination errors are output when the viewing window is partially contaminated.
- Sensitive: Contamination warnings and contamination errors are even output for isolated areas of contamination.
- Semi-sensitive: Contamination warning and contamination errors are output when the viewing window is partially and only slightly contaminated.
- Sensitive: Contamination warning and contamination errors are output even when there is just a little or local contamination.

NOTE

The cleaner the application environment is, the lower you can set the contamination measurement sensitivity. If a high precision of the measured values is required, the contamination measurement must be set to the most sensitive level.

Contamination warnings and contamination errors are displayed on the status indicators of the device see "LED fault indicators", page 75. You can also read these statuses via telegrams (available in the Telegram Listing, see "Telegram listing (EN)", page 88).

NOTE

The LMS14x Core has only two levels for the contamination display: No display / contamination measurement disabled; 80% contamination warning, 50% contamination error.

3.6.9 Calculation of the field size for mobile applications

In order to prevent collisions between vehicles, and between vehicles and fixed objects, the switching field must have sufficient length and width.

To calculate the switching field length, you need to take into consideration the stopping distance of the vehicle. This comprises the following:

- the braking distance, which can be found in the vehicle documentation
- the distance covered during the vehicle control's response time, which can be found in the vehicle documentation
- The distance covered during the response time of the LiDAR sensor, "Technical data", page 79.

NOTE

- We recommend adding a supplement of at least 100 mm to the protective field length in order to stop the vehicle before a possible collision.
- If retro-reflectors are situated in the path of the vehicles, or if you anticipate that the braking force of the vehicle will diminish over time, you may, under certain circumstances, need to increase the recommended supplement.
- The width of the switching field should cover the vehicle width. You should also configure a supplement of at least 100 mm on every side.

Mounting height

The recommended mounting height for mobile applications is at least 150 mm.

3.6.9.1 Switching field length

You must configure the switching field so that a minimum distance to the vehicle is maintained at all times. This ensures that a vehicle monitored by the LiDAR sensor stops before an object is reached. You can define multiple monitoring cases each with different switching fields. These can be switched over dynamically via static control inputs, for example to adjust the protective field size based on the vehicle speed.

In this kind of application, you must calculate the switching field sizes (in particular the switching field lengths) for all speeds.

The switching field length SL can be calculated using the following formula (guideline values based on a pixel calculation):

SL = SA + ZG + ZR + ZB

SA = Stopping distance

ZG = General supplement of the LiDAR sensor = 100 mm

ZR = Supplement for application-related influences or the selected application parameters

ZB = Supplement for the decreasing braking force of the vehicle. This can be obtained from the relevant vehicle documentation, or alternatively: 10% of the stopping distance.

Stopping distance SA

The stopping distance comprises the vehicle's braking distance and the distance covered during the response time of the LiDAR sensor and the vehicle control's response time.

NOTE
Please note that a vehicle's braking distance does not increase linearly with increasing speed, but rather in a squared relationship. This is particularly important if you switch between different-sized switching fields depending on the speed.

How to calculate the stopping distance SA:

$SA = S_{AnF} + S_{AnS} + S_{Br}$

S_{AnF} = The distance covered during the vehicle control's response time, which is specified in the vehicle documentation

S_{AnS} = The distance covered during the response time of the LiDAR sensor

S_{Br} = The braking distance, found in the vehicle documentation

The distance S_{AnS} covered during the response time of the LiDAR sensor depends on:

- the response time of the LiDAR sensor
- the maximum speed of the vehicle in your mobile application

For more information on the response time T_S of the LiDAR sensor, see "Technical data", page 79.

How to calculate the distance S_{AnS} covered during the response time of the LiDAR sensor:

$S_{AnS} = T_S \times V_{max}$

T_S = Response time of the LiDAR sensor

V_{max} Maximum speed of the vehicle, from the relevant vehicle documentation

The response time T_S of the LiDAR sensor depends on:

- the base response time of the LiDAR sensor
- whether multiple sampling is set
- Filter settings (e.g., particle filter)

ZR supplement

This supplement must be determined on an application-specific basis and taken into account appropriately. The following factors can make it necessary to use a supplement: reflectors or shiny objects on the scan plane, multi-echo analysis, blanking size, device filter (e.g., particle filter).

3.6.9.2 Switching field width

The width of the switching field must cover the width of the vehicle and take into account the supplements for the measurement error.

The switching field width SB can be calculated using the following formula (guideline values based on a pixel calculation):

SB = FB + 2 × (ZG + ZR)

FB = Vehicle width

ZG = General supplement of the LiDAR sensor = 100 mm

ZR = Supplement for application-related influences or the selected application parameters

3.7 Fields of application

In general, the device can be used for two purposes:

- Object measurement see "Object measurement", page 29
- Detection of objects using detection fields see "Field evaluation", page 37

That is why there is a wide range of possible applications. Some of these applications are listed below:

- Container loading/handling/positioning
- Traffic monitoring/management
- Robots/Pick-and-place applications
- Property/Building protection (low faulty alarm rate)

It is recommended to continuously check the correct output of the measurement data (measurement of objects) and switching states (field application) via plausibility tests during the application.

This can be done, for example, by analyzing the measurement data or a field evaluation for expected objects in the viewing range of the device (e.g., test target, fixed objects, regularly recurring objects).

3.7.1 Object measurement

3.7.1.1 Basic parameters

At a high scanning frequency or a finer angular resolution, the device delivers more measured values.

SOPAS ET: **Project tree**, **LMS...**, **Parameters**, **Basic settings**, **Current configuration** and **New configuration** areas.

NOTE

- The device outputs the data after the start of the measurement through the same interface via 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 the event of an error, the output of measured values is terminated immediately and an error code is output which can be evaluated by the connected application. The error code can also be queried from the device via SOPAS ET see "Detailed fault analysis", page 76.

NOTE

The LMS14x Core outputs the measured contour values via RS-232 / Ethernet, but only at a low refresh rate. They are used exclusively for visualization or configuration on the computer.

3.7.1.2 Multi-echo analysis

The distance between the device and an object is calculated via the time-of-flight of the emitted pulse. The device can evaluate up to two echo signals per emitted measuring beam to deliver reliable measurement results even under unfavorable ambient conditions.

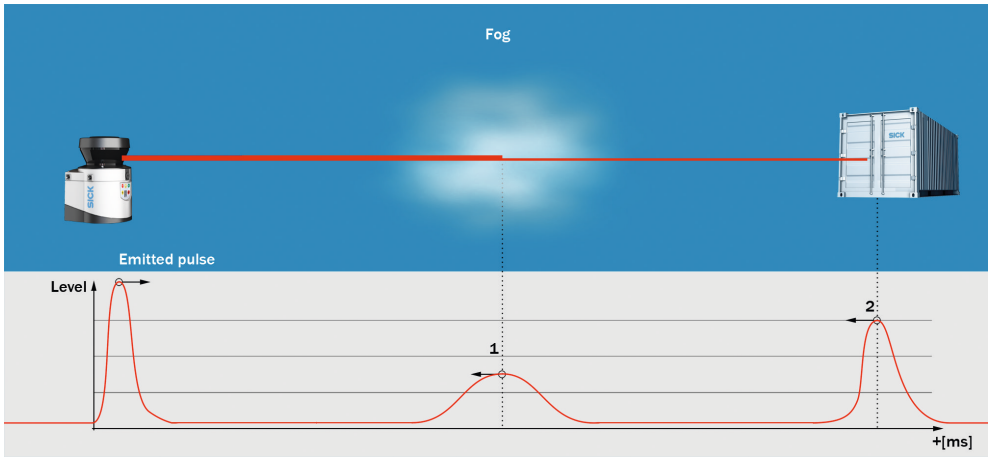


Figure 19: Multi-echo analysis: example industrial application for building management

3.7.1.3 Measuring through a pane

When measuring through a pane, one unique factor to remember is that different light reflections can occur, either from aspects directly on the pane (e.g., from contamination or scratches) or from reflections on the pane (i.e., caused by objects behind or next to the device). Reflections of objects on the pane must be shadowed.

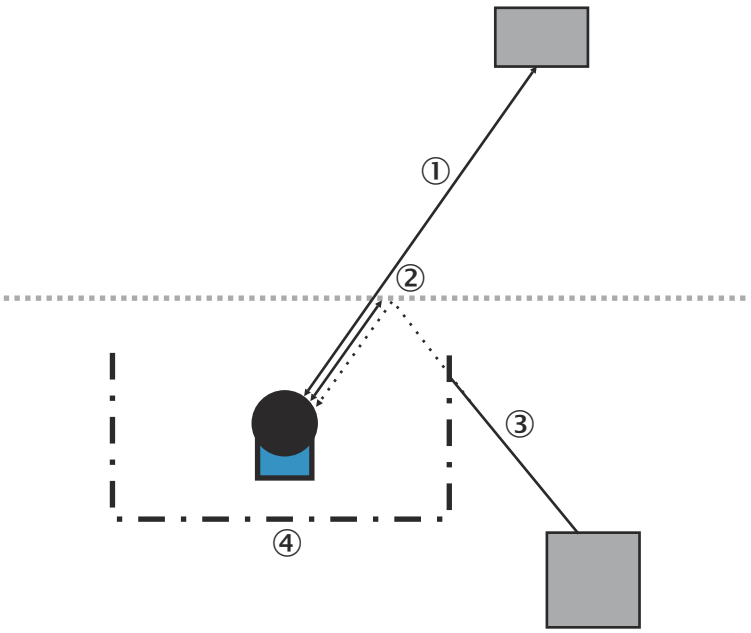


Figure 20: Shadowing reflections

① 2. Echo (object)

② 1. Echo (pane)

③ Shadowed echo (mirrored object)

④ Shading

NOTE

To avoid direct reflections back to the device (mirror effect), the laser pulses must not hit the pane perpendicularly. The pane should therefore be mounted slightly tilted relative to the horizontal axis of the device.

3.7.1.4 Filter

By using digital filters to pre-process and optimize the measured distance values, the device can be tailored to the specific requirements of the respective application. This makes it possible to prevent virtually all faults.

The active filter functions affect the outputted measured values. It is not possible to recalculate the original measured values from the filtered output values.

3.7.1.4.1 Fog filter

The fog filter enables the device to eliminate unwanted echoes at close range. This considerably lowers the probability of false activations at close range in fog.

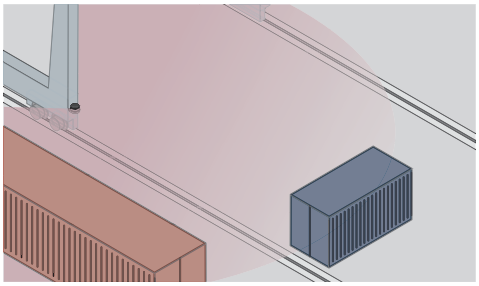


Figure 21: Without the fog filter: objects are difficult to detect through the fog due to reflections.

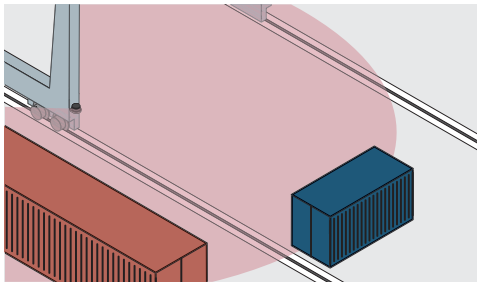


Figure 22: Using the fog filter: objects can be detected reliably because unwanted echoes are screened out.

3.7.1.4.2 Blanking window

Hardware blanking window

Due to a hardware blanking window, the device only provides measured values from a configured distance onwards. This means that an area in front of the device is completely hidden. You can configure a blanking window from 1 to 15 m.

The limits of the evaluation range are defined through the dimensioning of the respective evaluation fields. The limits defined in this way are quite coarse as the measurement data within the selected evaluation range are fully suppressed. The measuring range thus begins immediately at the field limit of the evaluation field. The measured values in the transition area of the evaluation field are less precise as a result.

Software blanking window

The measurement data is only output at distances beyond a defined software blanking window from the device. The defined evaluation range in front of the device is thereby completely blanked out. Software blanking windows from 20 mm up to 50,000 mm can be configured.

In contrast to the hardware blanking window, the software blanking window can be configured with millimeter precision. If there is an object in the software blanking window that already generates one echo, then only one more echo is available after the software blanking window. This is a difference compared to the hardware blanking window because both echoes would continue to be available in the case of the hardware blanking window.

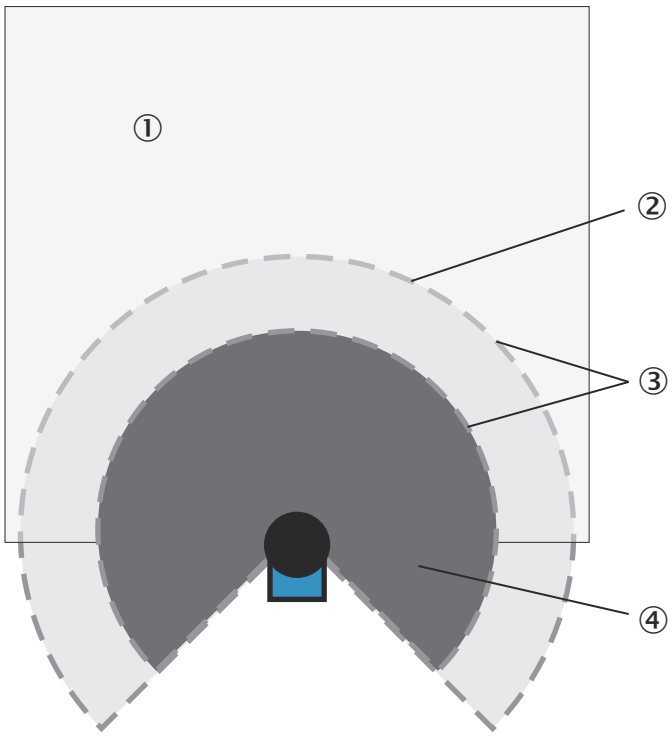


Figure 23: Influence of the hardware blanking window on the detection fields of the field application

① Detection field

② From 2 m: Reliable field evaluation

③ Up to 1 m before blanking window: Field evaluation not reliable

④ Blanking window: No measured values, no field evaluation

A hardware blanking window influences the detection fields of the field application **see "Detection fields", page 39****:**

- No field evaluation is possible in the area of the configured blanking window.
- Field evaluation is not possible reliably up to 1 m in front of the configured blanking window.
- Field evaluation is reliably possible from 2 m in front of the area of the configured blanking window.

3.7.1.4.3 Echo filter

The echo filter screens out unwanted measurement data and signals caused by edge hits, rain, dust, snow and other ambient conditions.

The other pulses triggered by undesirable ambient conditions are not taken into account.

3.7.1.4.4 Particle filter

The particle filter blanks small, irrelevant reflection pulses in dusty environments and in rain or snow which are caused by dust particles, raindrops, snowflakes or the like.

In doing so, successive scans are continuously evaluated in order to detect static objects.

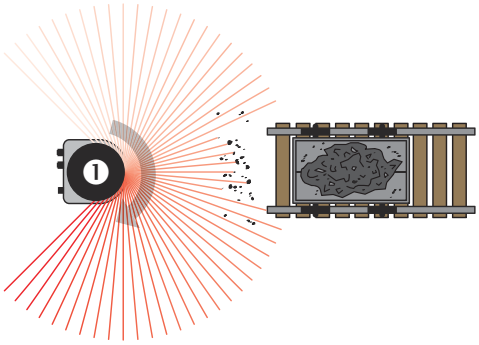


Figure 24: Without the particle filter: Violation of the contour due to dust particles in the vicinity of the object.

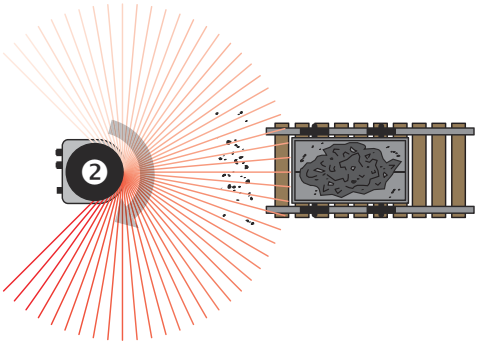


Figure 25: Using the particle filter: The response to dust particles in the detection field is delayed by one scan. Particles can thereby be blanked.

3.7.1.4.5 Average filter

Using the mean filter, a blockwise (not moving) arithmetic average is calculated from the configured number of scans and then outputted. The big advantage when using this filter: Potential noise, i.e. minimal deviation of values, is reduced, which also lowers the quantity of data.

3.7.1.5 Output of measured values

For output of measured values, the device supplies measured values at one of the data interfaces. Prerequisite for data output is that the device is in measurement mode. You have two options for starting the measurement mode:

- Start in SOPAS ET (**Project tree**, **LMS...**, **Parameters**, **Basic settings**, **Measurement** area)
- Start via a telegram see "Telegram listing (EN)", page 88.

NOTE
After starting the measurement mode, the device needs some time to reach status 7 (“measurement”). That is why you should request the status of the device with the sRN STIms telegram.

Then request measured data via telegram at the data interface from which you want to receive measured data. This can be achieved in two ways:

- The request of exactly one measured value with the sRN LMDscandata telegram – the last measured scan is transmitted.
- Continuous request of measured data with the sEN LMDscandata telegram – measured data is transmitted until the output of measured values is stopped with the sEN LMDscandata telegram.

Example of one-time output of measured values

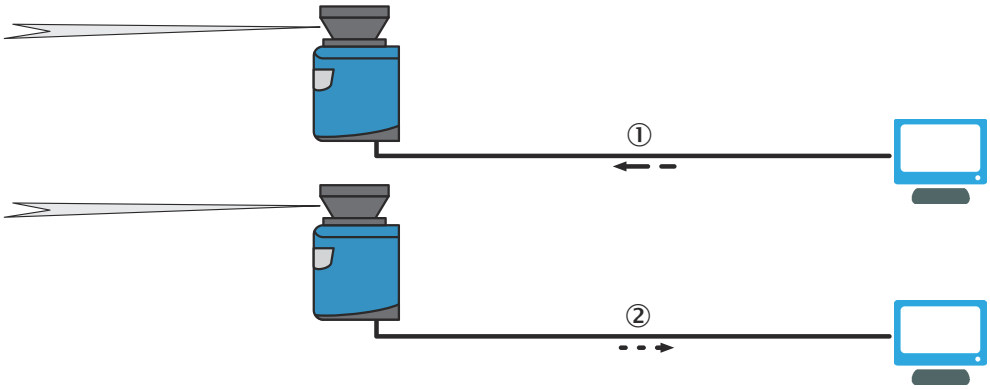


Figure 26: Query of measured value telegram

① Measured value query

② Output of a measured value telegram

1. Log in

Request: <STX>sMN SetAccessMode 03 F4724744<ETX>

Reply: <STX>sAN SetAccessMode <ETX>

2. Start measurement

Request: <STX>sMN LMCstartmeas<ETX>

Reply: <STX>sAN LMCstartmeas 0<ETX>

3. Log out and start device

Request: <STX>sMN Run<ETX>

Reply: <STX>sAN Run 0<ETX>

4. Request measurement status

NOTE
Query the status until status 7 (“measurement”) is achieved in the response.

Request: <STX>sRN STlms<ETX>

Reply: <STX>sRA STlms 7 0 8 00:00:00 8 01.0 1.06 0 0 0<ETX>

NOTE
If the status is less than 7, send request again.

5. Starting output of measured values of a single scan

Request: <STX>sRN LMDscandata<ETX>

Reply: <STX>sRA LMDscandata<ETX>

If an error is present and reported in the response telegram: Depending on the type of error, the device will either stop measurement mode or not.

Example of continuous output of measured values

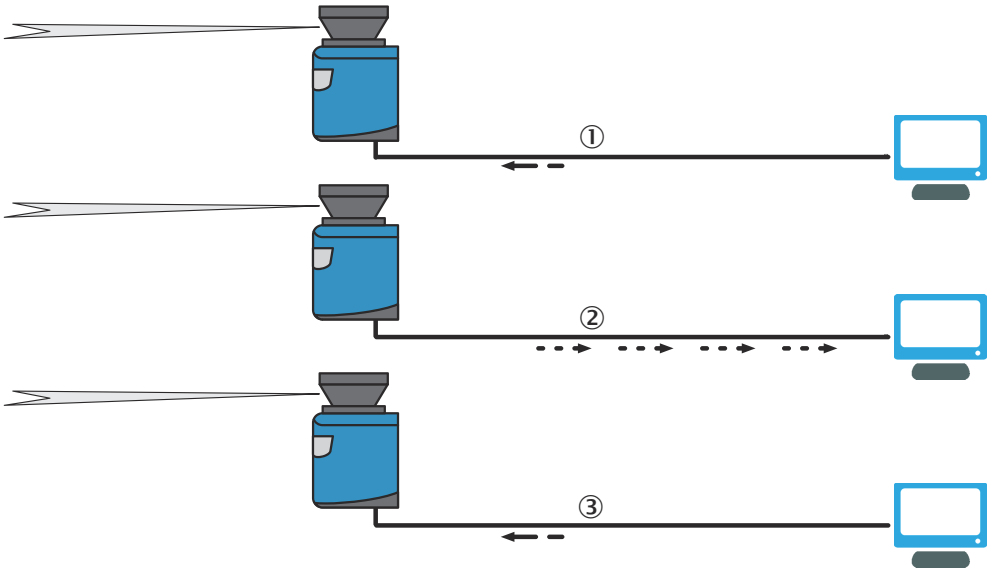


Figure 27: Continuous output of measured values

① Measured value query

② Output measured-value telegrams

③ Command to stop the output

NOTE

If you cannot be sure that the scan data can be processed at the same speed as it is output by the device, you should only query the scan data individually. The scan counter, telegram counter, and time stamp in the output of measured values can serve as an indication that processing is too slow.

1. Log in

Request: <STX>sMN SetAccessMode 03 F4724744<ETX>

Reply: <STX>sAN SetAccessMode <ETX>

2. Start measurement

Request: <STX>sMN LMCstartmeas<ETX>

Reply: <STX>sAN LMCstartmeas 0<ETX>

3. Log out and start device

Request: <STX>sMN Run<ETX>

Reply: <STX>sAN Run 0<ETX>

4. Request measurement status

NOTE

Query the status until status 7 (“measurement”) is reached in the response.

Request: <STX>sRN STlms<ETX>

Reply: <STX>sRA STlms 7 0 8 00:00:00 8 01.0 1.06 0 0 0<ETX>

NOTE
If the status is less than 7, send request again.

5. Starting continuous output of measured values

The scan data is output until output of measured values is over.

Request: <STX>sEN LMDscandata 1<ETX>

Confirmation: <STX>sEA LMDscandata 1<ETX>

Continuous response: <STX>sSN LMDscandata<ETX>

6. Stopping continuous output of measured values

Request: <STX>sEN LMDscandata 0<ETX>

Confirmation: <STX>sEA LMDscandata 0<ETX>

3.7.2 Field evaluation

Using the integrated field application, the device evaluates up to ten detection fields within its scanning range, depending on the type. You can use the field application to implement systems for collision protection, object protection or access monitoring, for example.

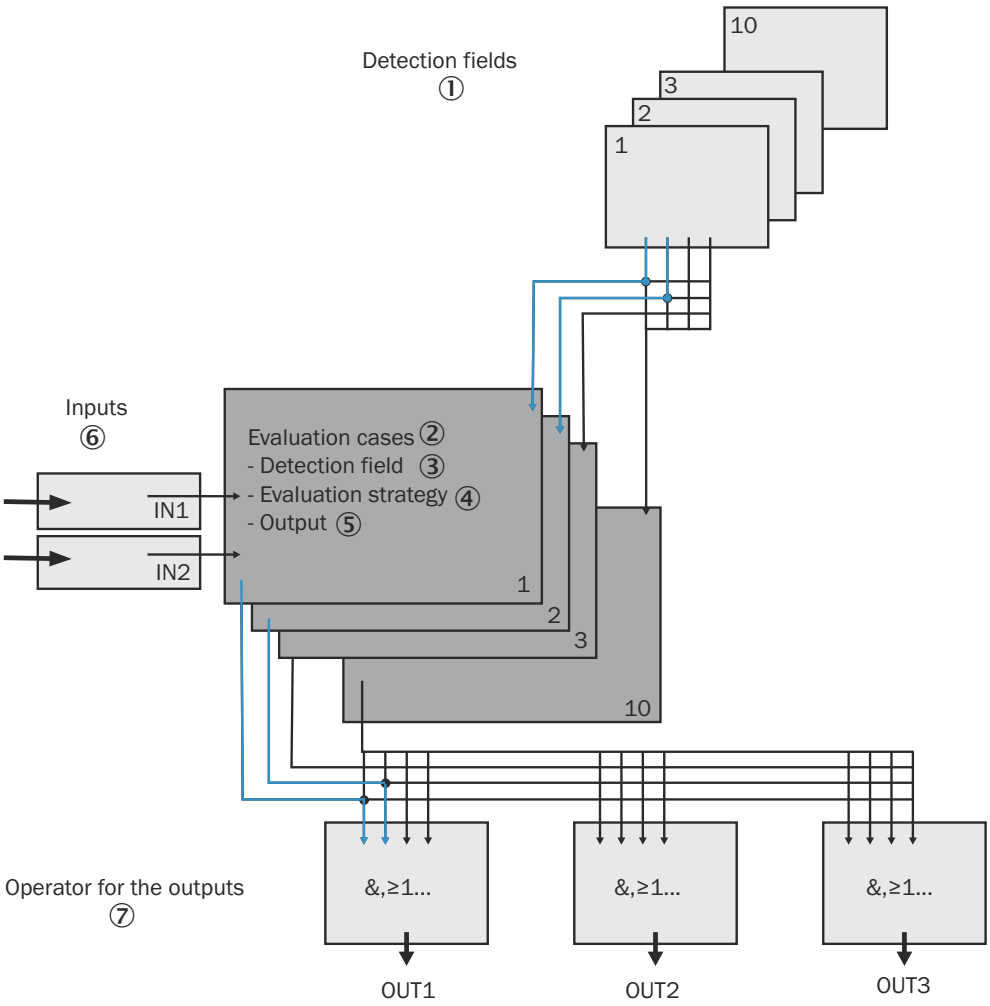


Figure 28: Principle of field evaluation

① Detection fields
② Evaluation cases
③ Detection field
④ Analysis strategy
⑤ Outputs
⑥ Inputs
⑦ Linking of outputs

The device is adapted to the evaluation situation with the help of up to ten evaluation cases. In the evaluation case, one of ten configurable detection fields, an evaluation strategy, an output, and possibly a combination of inputs or time periods that activate the evaluation case are selected. For each output, a link is chosen which determines the result of the output if more than one evaluation case affects the output.

In the example in figure 28, detection field 1 is used in evaluation case 1, evaluation field 2 in evaluation case 2. Both evaluation cases act on output OUT1. If the results of the evaluation cases are AND-linked, the output only switches when both evaluation cases report an event.

3.7.2.1 Detection fields

Using the integrated field application, you can configure up to ten detection fields. The size and shape of these detection fields are (almost) freely configurable.

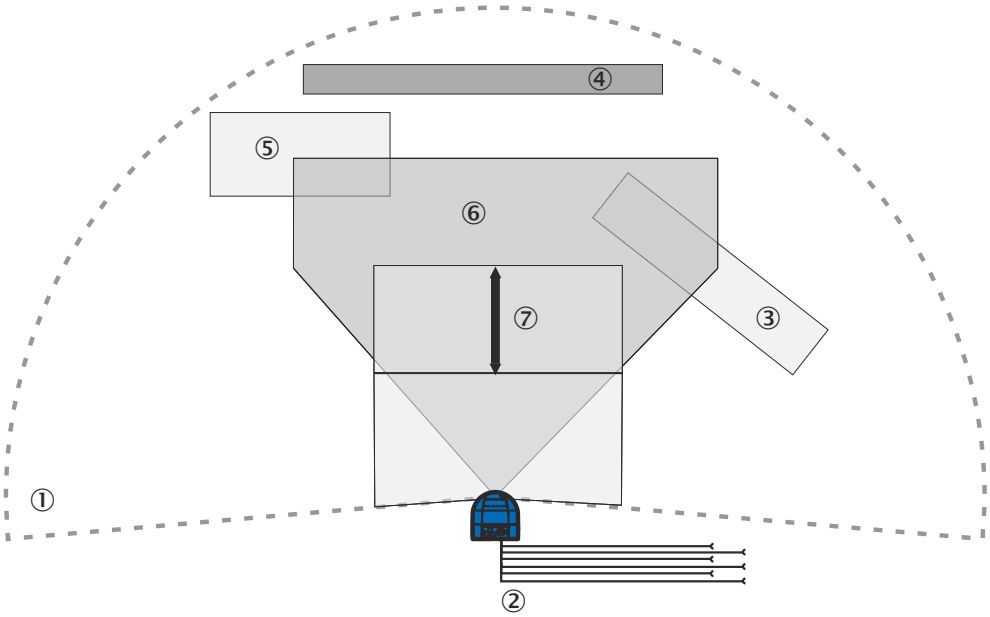


Figure 29: Examples of 4 different detection field shapes

① Measuring range of the device
② Outputs
③ Rotated detection field
④ Detection field for contour monitoring
⑤ Square detection field with distance to the device
⑥ Polygonal detection field
⑦ Dynamic detection field

The detection fields can be drawn using SOPAS ET according to the needs of your application. Detection fields can have the following properties:

- Polygon
- Square
- Beginning on device
- With distance to device (“island field”)
- Dynamic, i.e. the length varies according to the speed detected by the encoder, see "Encoder inputs", page 42 (LMS10x/LMX11x/LM15x only)

You configure the detection fields in SOPAS ET: **Project tree**, **LMS...**, **Parameters**, **Evaluation fields**.

If the area to be monitored changes, the field can be easily reconfigured using the SOPAS ET configuration software.

3.7.2.2 Evaluation cases

An evaluation case determines which detection field is evaluated in which way and on which output it acts.

You can configure up to ten evaluation cases; all configured evaluation cases are active simultaneously.

For each evaluation case, configure in SOPAS ET:

- Inputs or time periods that activate an evaluation case, if applicable. If nothing is specified, the evaluation case is always active.
- The evaluation strategy
- The detection field
- The output affected by the analysis case
- The response time **Project tree**, **LMS...**, **Parameters**, **Evaluation cases**.

Inputs

If the evaluation case is not permanently active, you can configure an input combination that activates the evaluation case.

Table 7: Examples for input combinations with the LMS10x/LMS11x/LMS15x/LMS16x

Input 1	Input 2	Evaluation case
Active high	Active low	1
Active high	Active low	2
Active low	Active high	3
Active low	Active low	4

NOTE
An input combination can also be defined for several evaluation cases, then e.g. two evaluation cases become active simultaneously.

NOTE
LMC12x/LMC13x VdS (object protection) The entry conditions for these devices are provided in the document “Technical Information (VdS Installer Instructions)”, Part no.: 8013748

LMS12x/LMS13x/LMS14x Security The entry conditions for these devices are provided in the document “Technical Information (Security Installer Instructions)”, Part no.: 8014486

Analysis strategy

In SOPAS ET, you choose from the following possible detection strategies:

- **Pixel analysis**: The device evaluates the entire area of the field. Each individual beam is taken into account in evaluation. If an object enters the field, this result is forwarded to the corresponding output.
- **Blanking**: The device evaluates the entire area of the field. However, objects that are smaller than the specified size are ignored. An object is only detected if it is larger than the configured blanking size. However, this only applies to objects that are at the maximum distance at which they still meet the criteria for the minimum object size. At greater distances, the set blanking size can no longer be achieved by the optical resolution of the device. The device behaves exactly as with pixel evaluation.
- **Contour**: The device also evaluates the existence of a contour, which has to be fully situated within the detection field at all times. This allows the device to detect, for example, that a door opens outwards or that the position of the device has changed. It is also possible to detect someone crawling beneath a vertical detection field or a laser beam deflected in a mirror. Blanking can be used to hide the missing part of the contour up to a certain size.
- **I/O linking**: The I/O linking evaluation strategy allows you to link the inputs of the device to its outputs see "Inputs and outputs", page 42.

Response time

Define a response time for the **Pixel analysis**, **Blanking** and **Contour** evaluation strategies. For a device to detect an object using the **Pixel analysis** and **Blanking** evaluation strategies, it must be detected at one point for at least the duration of the response time. For the **Contour** evaluation strategy, the contour breach must be detected at the same location for at least the duration of the response time.

Manipulation prevention

When **Pixel analysis** is configured, dazzling can cause the device to stop monitoring a field. When **Blanking** is configured, small objects in the vicinity of the device can cause large shadows.

If you use detection fields at a distance to the device, the object or the object erroneously measured due to dazzle lies outside the detection field and is not detected.

To prevent this, you can configure the **Manipulation protection** option.

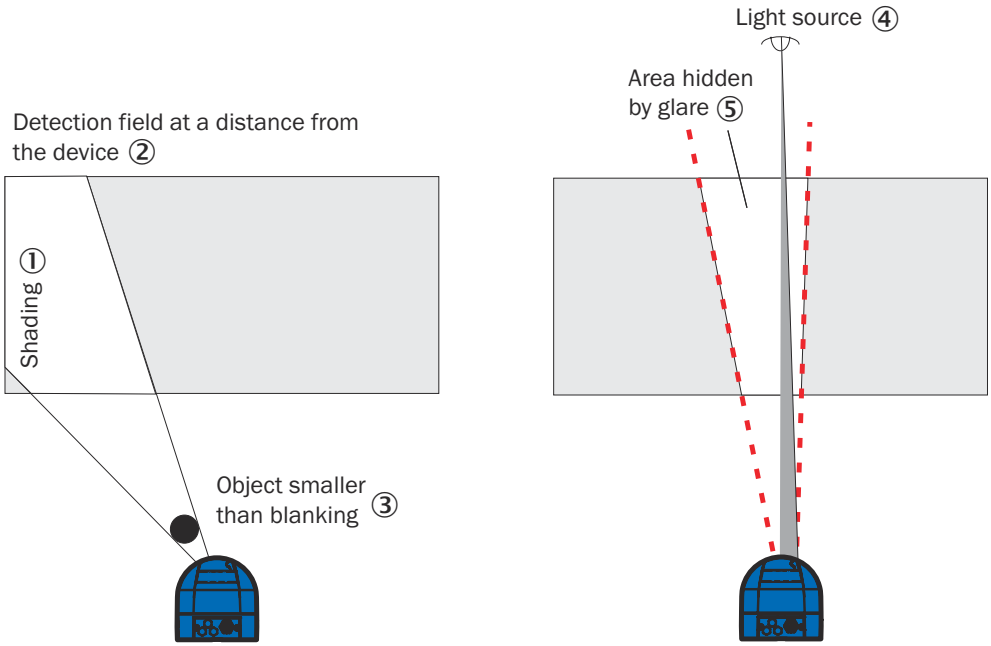


Figure 30: Tamper-proofing against shadowing and dazzle

① Shadowing
② Detection field with distance to the device
③ Object smaller than blanking
④ Light source
⑤ Blanked area

The **Manipulation protection** option switches the detection field:

- If an object less than or equal to the blanked object size is in front of the laser output aperture of the device for the response time configured for manipulation protection.
- If the device is blanked for longer than the response time configured for manipulation protection.

Detection field

For the evaluation case, select a field from the already configured detection fields. Its shape must fit the evaluation strategy see "Detection fields", page 39.

Output

Select one of the outputs for the evaluation case. If several evaluation cases act on one output, you must determine how the results of the evaluation cases are linked see "Linking of evaluation cases at the output", page 42.

NOTE
The respective output must be configured for “Application”.

Negating the result

Negating the result reverses the field evaluation output. The output used is thus switched, for example, if the detection field is free or if the contour is not interrupted.

NOTE
Do not confuse negation of the result with the Active HIGH/Active LOW setting of the outputs see "Linking of evaluation cases at the output", page 42.

3.7.2.3 Linking of evaluation cases at the output

If several evaluation cases act on one output, you must determine how the results of the evaluation cases are linked. The respective results can be linked with AND or OR.

SOPAS ET: **Project tree**, **LMS...**, **Parameters**, **Network/Interfaces/IOs**, **Digital outputs**.

The outputs are configured as **Active Low** by default. You can also configure them as **Active High**.

Resetting an output

In the default setting, the outputs are immediately reset to the deactivated state. You can configure a delay of up to 10 s (e.g. to activate a horn or to transmit the output signal to a PLC).

Alternatively, you can also reset the output via an input. The output is only reset when the assigned input assumes the configured status.

3.8 Inputs and outputs

NOTE
LMC12x/LMC13x VdS (object protection) Further information on the designs of the inputs and outputs of these devices is provided in the document “Technical Information (VdS Installer Instructions)”, Part no.: 8013748

LMS12x/LMS13x/LMS14x/LMS17x Security Further information on the designs of the inputs and outputs of these devices can be found in the document “Technical Information (Security Installer Instructions)”, Part no.: 8014486

Digital inputs

The device has two digital inputs. These inputs could activate different evaluation cases see "Evaluation cases", page 39.

The inputs can also be used to reset the outputs of the device see "Resetting an output", page 42.

Encoder inputs

The LMS10x/LMS11x/LMS15x/LMS16x has two encoder inputs.

The encoder pulses can be used to influence the size of so-called dynamic fields, for example for speed-dependent vehicle monitoring. A dynamic field changes its length with the speed measured e.g. by encoders.

SOPAS: **Project tree**, **LMS...**, **Parameters**, **Increment Configuration**.

When stationary (V = 0 m/s), the detection field has the size of the configured basic field. The size increases continuously with increasing speed until the largest field extension is reached at maximum speed.

Digital outputs

The LMS10x/LMS11x/LMS15x/LMS16x has three digital outputs.

The outputs can be used as digital outputs to ground, as floating outputs, or as resistance-monitored outputs see "Wiring the inputs and outputs", page 62.

SOPAS: **Project tree**, **LMS...**, **Parameters**, **Network/Interfaces/IOS**, **Digital Outputs**.

If the field evaluation application is being used to switch an output, the device may signal object detections or contour interruptions. In SOPAS ET, you can configure which evaluation case affects which output.

External digital outputs

Additional digital outputs can be controlled via a CAN expansion module (part no. 6038825). To use the additional outputs, the expansion module must be activated.

The external digital outputs have the same functional options as the digital outputs of the device.

SOPAS: **Project tree**, **LMS...**, **Parameters**, **Network/Interfaces/IOS, External Outputs**.

Linking of inputs and outputs

With the aid of an evaluation case, you can link the inputs and outputs of several devices together see "Evaluation cases", page 39.

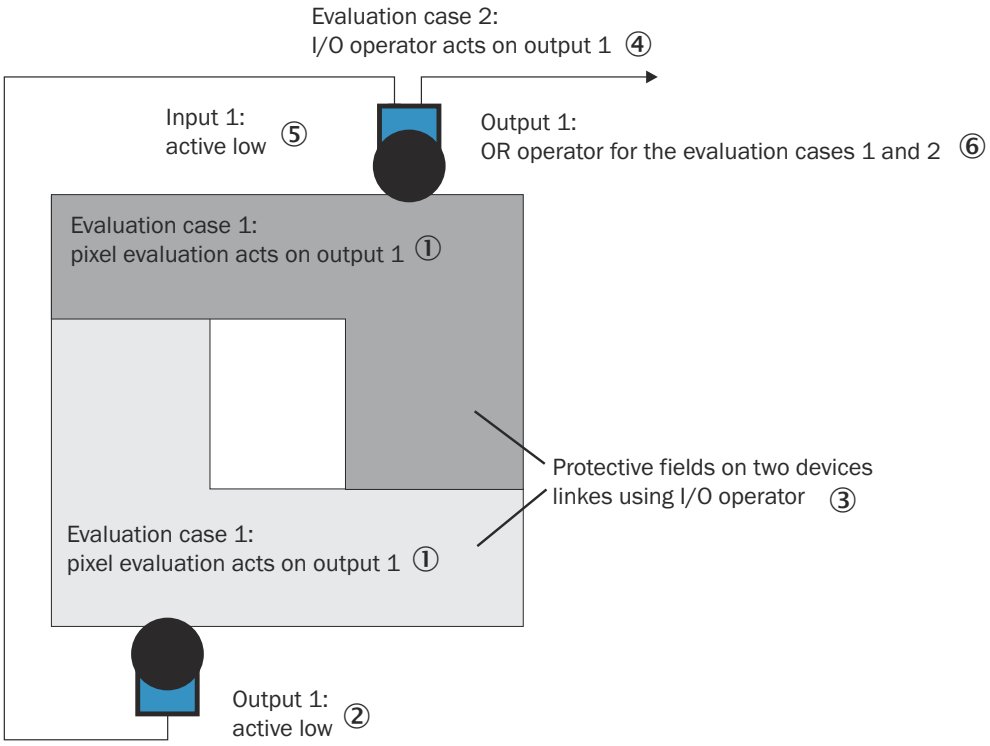


Figure 31: Linking of inputs and outputs

① Evaluation case 1: Pixel evaluation acts on output 1

② Output 1: Active low

③ Protective fields of two devices linked by I/O link

④ Evaluation case 2: I/O link acts on output 1

⑤ Input 1: Active low

⑥ Output 1: OR-link of evaluation cases 1 and 2

In the example above, output 1 of the lower device is connected to input 1 of the upper device. An interruption of the detection field is thereby reported to the input of the upper device. In its evaluation case 2, this device links the input to output 1. At the same time, evaluation case 1 affects the upper device and its output 1. By OR-linking the two results, evaluation field interruptions are reported on both devices at output 1 of the upper device.

3.9 Data interfaces

The device has different data interfaces for configuration and the transmission of measured values.

NOTE

- It is only possible to output all measured values of a scan in real time using the Ethernet interface.
- The data transmission rate of the RS-232 interfaces is limited. Therefore these interfaces are not suitable for transmitting scan data in real time.

For a description of the electrical interfaces see "Connections", page 55.

Ethernet interface

The Ethernet interface has a data transmission rate of 10/100 Mbit/s. The interface is a TCP/IP, UDP/IP interface supporting full duplex and half duplex.

The Ethernet interface allows the configuration of the device as well as the output of measured values.

The factory setting for the Ethernet interface is as follows:

- IP address: 192.168.0.1
- Subnet mask: 255.255.255.0
- TCP port: 2111

You might need to adjust the configuration of the Ethernet interface so that a connected computer (client) can communicate with the device via Ethernet: **Project tree**, **LMS...**, **Parameters, Network/Interfaces/IOs**, **Ethernet**.

NOTE

If you change the parameters of the Ethernet interface via the Ethernet interface, you must first save the data permanently to non-volatile memory in the device and then restart the device. A **Restart** button is provided in SOPAS ET for this purpose.

CAN

The CAN interface supports data transmission rates between 10 bit/s and 1 MBit/s. For data communication via CAN, you need to configure the device accordingly so that it can communicate with the host: SOPAS **Project tree**, **LMS...**, **Network/Interfaces/IOs**, **Serial**, **Serial host interface** area.

The following interface parameters can be configured

- Baud rate of the CAN bus
- ID of the device in the CAN

Serial host interface

NOTE

The LMS12x/LMS13x/LMS14x/LMS17x Security and LMC12x/LMC13x VdS variants do not have a serial host interface.

The serial host interface is an RS-232 interface. The host interface permits the configuration of the device and limited output of measured values.

The interface parameters are freely configurable: SOPAS ET **Project tree**, **LMS...**, **Parameters, Network/Interfaces/IOs**, **Serial**, **Serial host interface** area.

The factory setting for the host interface is as follows:

- 57.6 kBd
- 8 data bits
- 1 stop bit
- No parity

NOTE

If you change the parameters for the host interface via the host interface, the connection to the device will be lost. You must then scan for the device again in SOPAS ET (see "Performing a scan", page 67).

Serial aux interface

The serial host interface is an RS-232 interface. The auxiliary interface can be used to configure the device.

The interface parameters are freely configurable: SOPAS ET **Project tree**, **LMS...**, **Network/Interfaces/IOs**, **Serial**, **Serial auxiliary interface** area.

The factory settings for the auxiliary interface are configured as follows:

- 57.6 kBd
- 8 data bits
- 1 stop bit
- No parity

NOTE

If you change the parameters for the auxiliary interface via the auxiliary interface, the connection to the device will be lost. You must then scan for the device again in SOPAS ET (see "Performing a scan", page 67).

3.10 Communicating data via telegrams

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

- Request measured values via the host and then output them once via the device
- Parameter setting by the host for the configuration of the device
- Parameters and status log querying by the host.

Each telegram comprises a frame and the application data.

A detailed description of the different telegrams can be found in the Telegram Listing, see "Telegram listing (EN)", page 88.

Frame and coding of the telegrams

The payload varies depending on the coding.

Table 8: Frame of the telegrams when using ASCII coding (CoLa-A)

	Frame	Telegram	Frame
Designation	STX	Application data	ETX
Length (Byte)	1	≤ 30 kB	1
Description	Start of text character	ASCII coded	End of text character

You can configure the frame of the serial host interface in SOPAS ET: **Project tree**, **LMS...**, **Parameters**, **Network/Interfaces/IOs**, **Serial**, **Serial host interface** area.

Table 9: Frame of the telegrams when using binary coding

	Frame					Telegram	Frame
Designation	STX	STX	STX	STX	Telegram length	Payload	Checksum
Length (Byte)	1	1	1	1	4	≤ 2,495	1
Description	Start of text character				Length of the payload without checksum	Binary coded. The length depends on the previous send telegram.	The checksum is calculated by XORing all bytes of the payload, i.e., without frame.

4 Transport and storage

4.1 Transport

NOTICE
Damage due to improper transport!

- The product must be packaged with protection against shock and damp.
- Recommendation: Use the original packaging.
- Note the symbols on the packaging.
- Do not remove packaging until immediately before you start mounting.

4.2 Unpacking

- To protect the device against condensation, allow it to equilibrate with the ambient temperature before unpacking if necessary.
- Handle the device with care and protect it from mechanical damage.

4.3 Transport inspection

Immediately upon receipt in Goods-in, check the delivery for completeness and for any damage that may have occurred in transit. In the case of transit damage that is visible externally, proceed as follows:

- Do not accept the delivery or only do so conditionally.
- Note the scope of damage on the transport documents or on the transport company's delivery note.
- File a complaint.

NOTE
Complaints regarding defects should be filed as soon as these are detected. Damage claims are only valid before the applicable complaint deadlines.

4.4 Storage

- Do not store outdoors.
- Store in a place protected from moisture and dust.
- Recommendation: Use the original packaging.
- Do not expose to any aggressive substances.
- Protect from sunlight.
- Avoid mechanical shocks.
- Storage temperature: see "Technical data", page 79.
- Relative humidity: see "Technical data", page 79.
- For storage periods of longer than 3 months, check the general condition of all components and packaging on a regular basis.

5 Mounting

5.1 Mounting instructions

- Observe the technical data.
- Protect the sensor from direct sunlight.
- To prevent condensation, avoid exposing the device to rapid changes in temperature.
- The mounting site has to be designed for the weight of the device.
- It should be mounted so that it is exposed to as little shock and vibration as possible. Optional mounting accessories are available, see "Accessories", page 87.
- Observe maximum torque of the fixing screws: M5 = max. 5.9 Nm
- Regularly check the tightness of the fixing screws.
- For indoor mounting, use a protection hood if necessary, for outdoor installation use a weather protection hood if necessary (both optional accessories), see "Accessories", page 87.
- Do not mount the device on or directly in front of a bright metallic surface or other reflective surface, since reflections can falsify the measurements.
- Avoid having shiny or reflective surfaces in the scanning range, e.g., stainless steel, aluminum, glass, reflectors, or surfaces with these types of coatings.
- Protect the device from moisture, contamination, and damage.
- Make sure that the status indicator is clearly visible.
- Do not subject the device to excessive shock or vibrations. In systems subjected to heavy vibrations, secure the fixing screws with screw-locking devices.

5.2 Mounting the device

NOTE

For direct mounting without the SICK mounting bracket:

- There must be a sufficient distance between the housing and the support structure to prevent heat accumulation.
- It must be ensured that no moisture is permanently present on the sealing surfaces and that the housing can dry in a short time.
- In order to prevent bimetallic corrosion, appropriate mounting materials must be used or there must be sufficient distance between the housing (die cast aluminum) and the support structure.

1. Mount the device directly or in a suitably prepared bracket using the fixing holes provided (see "Dimensional drawings", page 85). Mounting brackets are available as optional accessories, see "Accessories", page 87.
2. Make the electrical connection, see "Electrical installation", page 50.
3. Align the vertical center line of the field of view of the device with the center of the area to be monitored.
4. Switch on the supply voltage.

✔ The green **OK** LED lights up after initialization.

5. The device is ready for use. Perform a fine adjustment using a test target and, if necessary, use the alignment aid, see "Accessories", page 87.

5.3 Mounting multiple devices

The device has been designed to minimize the probability of mutual interference, including between different LiDAR sensors. To rule out even the slightest effects on the measurement accuracy, the devices should be arranged such the laser beams are not received by another device.

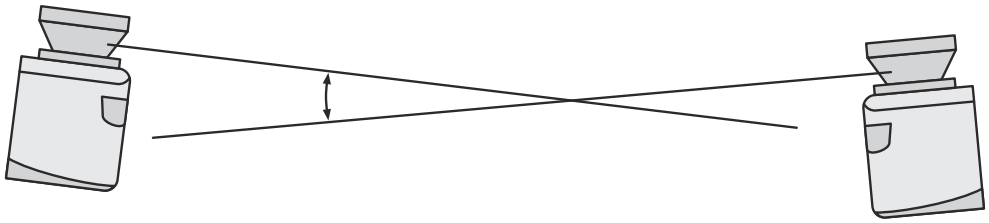


Figure 32: Arrangement for 2 devices

6 Electrical installation

6.1 Wiring instructions

NOTE

Pre-assembled cables can be found on the product page.

The product page can be accessed via the **SICK Product ID**: **pid.sick.com/{P/N}/{S/N}**

{P/N} corresponds to the part number of the product, see type label.

{S/N} corresponds to the serial number of the product, see type label (if indicated).

NOTICE

Faults during operation and defects in the device or the system

Incorrect wiring may result in operational faults and defects.

- Follow the wiring notes precisely.

The enclosure rating stated in the technical data is achieved only with screwed plug connectors or protective caps.

All circuits connected to the device must be configured as SELV or PELV circuits. SELV = safety extra-low voltage, PELV = protective extra-low voltage.

Connect the connecting cables in a de-energized state. Do not switch on the supply voltage until installation is complete and all connecting cables are connected to the device and controller.

Wire cross-sections in the supply cable from the customer's power system must be implemented in accordance with the applicable standards.

6.2 Overview of the installation steps

1. Wire the digital outputs (application-dependent).
2. Connect the computer temporarily (configuration).
3. Wire the data interface for operation.
4. Set up the voltage supply for the device.

6.3 Prerequisites for safe operation of the device



WARNING

Risk of injury and damage caused by electrical current!

As a result of equipotential bonding currents between the device and other grounded devices in the system, faulty grounding of the device can give rise to the following dangers and faults:

- Dangerous voltages are applied to the metal housings.
- Devices will behave incorrectly or be destroyed.
- Cable shielding will be damaged by overheating and cause cable fires.

Remedial measures

- Only skilled electricians should be permitted to carry out work on the electrical system.
- If the cable insulation is damaged, disconnect the voltage supply immediately and have the damage repaired.
- Ensure that the ground potential is the same at all grounding points.
- Where local conditions do not meet the requirements for a safe earthing method, take appropriate measures. For example, ensure low-impedance and current-carrying equipotential bonding.

The device is connected to the peripheral devices (any local trigger sensor(s), system controller) via shielded cables. The cable shield – for the data cable, for example – rests against the metal housing of the device.

The device can be grounded through the cable shield or through a blind tapped hole in the housing, for example.

If the peripheral devices have metal housings and the cable shields are also in contact with their housings, it is assumed that all devices involved in the installation have the **same ground potential**.

This is achieved by complying with the following conditions:

- Mounting the devices on conductive metal surfaces
- Correctly grounding the devices and metal surfaces in the system
- If necessary: low-impedance and current-carrying equipotential bonding between areas with different ground potentials

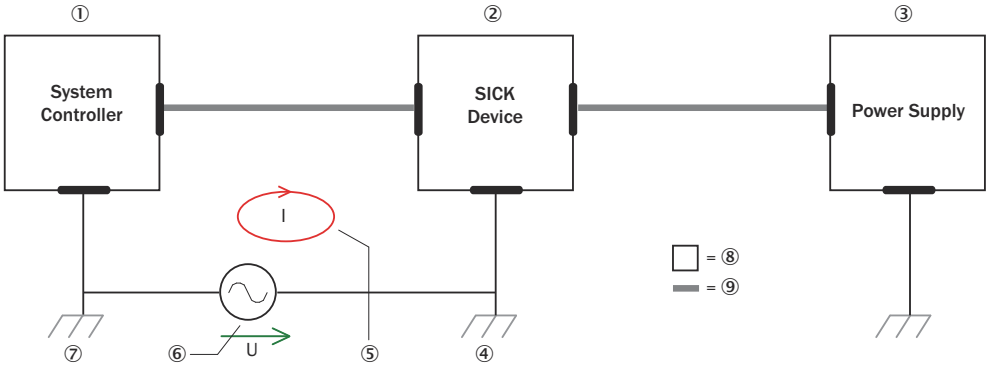


Figure 33: Example: Occurrence of equipotential bonding currents in the system configuration

① System controller
② Device
③ Voltage supply
④ Grounding point 2
⑤ Closed current loop with equalizing currents via cable shield
⑥ Ground potential difference
⑦ Grounding point 1
⑧ Metal housing
⑨ Shielded electrical cable

If these conditions are not fulfilled, equipotential bonding currents can flow along the cable shielding between the devices due to differing ground potentials and cause the hazards specified. This is, for example, possible in cases where there are devices within a widely distributed system covering several buildings.

Remedial measures

The most common solution to prevent equipotential bonding currents on cable shields is to ensure low-impedance and current-carrying equipotential bonding. If this equipotential bonding is not possible, the following solution approaches serve as a suggestion.

NOTICE
We expressly advise against opening up the cable shields. This would mean that the EMC limit values can no longer be complied with and that the safe operation of the device data interfaces can no longer be guaranteed.

Measures for widely distributed system installations

On widely distributed system installations with correspondingly large potential differences, the setting up of local islands and connecting them using commercially available **electro-optical signal isolators** is recommended. This measure achieves a high degree of resistance to electromagnetic interference.

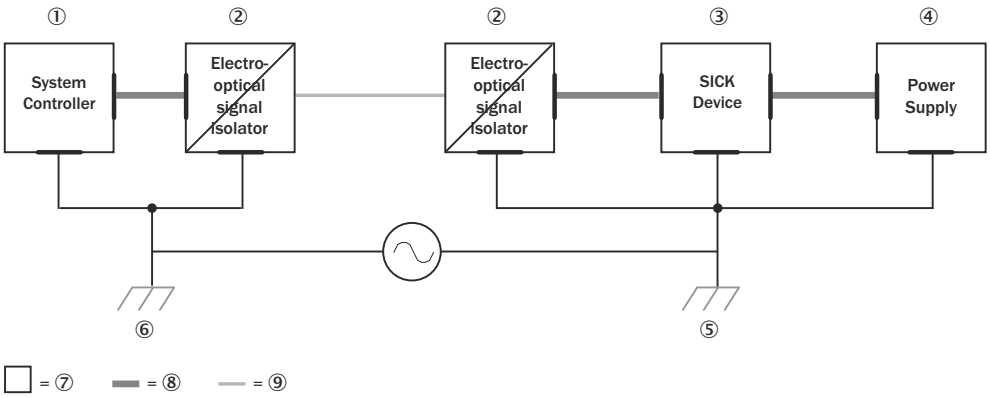


Figure 34: Example: Prevention of equipotential bonding currents in the system configuration by the use of electro-optical signal isolators

① System controller
② Electro-optical signal isolator
③ Device
④ Voltage supply
⑤ Grounding point 2
⑥ Grounding point 1
⑦ Metal housing
⑧ Shielded electrical cable
⑨ Fibers

The use of electro-optical signal isolators between the islands isolates the ground loop. Within the islands, a stable equipotential bonding prevents equalizing currents on the cable shields.

Measures for small system installations

For smaller installations with only slight potential differences, insulated mounting of the device and peripheral devices may be an adequate solution.

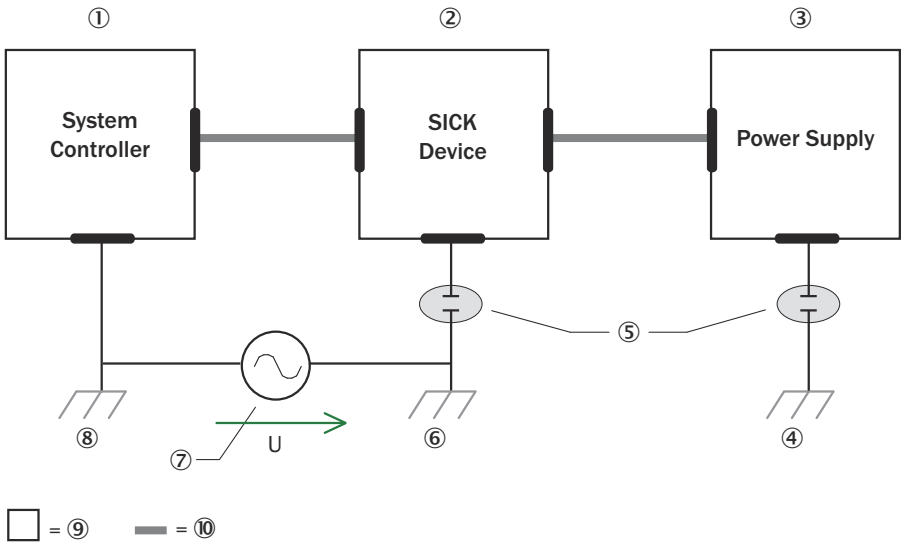


Figure 35: Example: Prevention of equipotential bonding currents in the system configuration by the insulated mounting of the device

① System controller
② Device
③ Voltage supply
④ Grounding point 3
⑤ Insulated mounting
⑥ Grounding point 2
⑦ Ground potential difference
⑧ Grounding point 1
⑨ Metal housing
⑩ Shielded electrical cable

Even in the event of large differences in the ground potential, ground loops are effectively prevented. As a result, equalizing currents can no longer flow via the cable shields and metal housing.

NOTICE

The voltage supply for the device and the connected peripheral devices must also guarantee the required level of insulation.

Under certain circumstances, a tangible potential can develop between the insulated metal housings and the local ground potential.

6.4 Connections

6.4.1 Connections of the LMS10x

Table 10: Pin assignment of the "Ethernet" connection

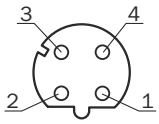
Male/female connector	Contact	Short form	Signal description
 **Figure 36:** M12 female connector, 4-pin, D-coded	1	TX+	Sender+
	2	Rx+	Receiver+
	3	TX–	Sender–
	4	Rx–	Receiver–

Table 11: Pin assignment of the "AUX" connection

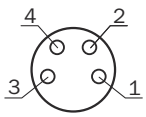
Male/female connector	Contact	Short form	Signal description
 **Figure 37:** M8 female connector, 4-pin	1	RxD AUX	Receiver RS-232 (auxiliary interface)
	2	Reserved	Do not use
	3	GND RS	RS-232 ground
	4	TxD AUX	Sender RS-232 (auxiliary interface)

Table 12: LMS10x: Terminal assignment (2 x terminal strip, 17-pin) in the system plug

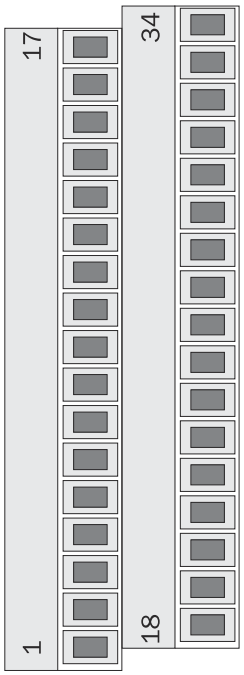
Male/female connector	Contact	Short form	Signal description
	01	Reserved	Do not use
	02	Reserved	Do not use
	03	RxD HOST	Receiver RS-232 (Host interface)
	04	Reserved	Do not use
	05	Reserved	Do not use
	06	IN1	Digital input 1
	07	GND IN1	Digital input 1 ground
	08	IN2	Digital input 2
	09	GND IN2	Digital input 2 ground
	10	INC1 A	Input encoder 1, contact A
	11	INC1 B	Input encoder 1, contact B
	12	GND INC1	Encoder 1ground
	13	OUT1 A	Digital output 1, contact A
	14	OUT1 B	Digital output 1, contact B
	15	OUT1 R	Digital output 1, resistance monitoring
	16	OUT2 A	Digital output 2, contact A
	17	OUT2 B	Digital output 2, contact B
	18	GND	Sensor ground
	19	V_S	Sensor supply voltage
	20	TxD HOST	Sender RS-232 (Host interface)
	21	Reserved	Do not use
	22	GND RS/CAN	RS-232/CAN ground
	23	CAN H	CAN BUS high
	24	CAN L	CAN BUS low
	25	VS CAN (DC 24 V)	DC 24 V supply voltage CAN bus
	26	GND RS/CAN	RS232/CAN ground
	27	CAN H	CAN BUS low
	28	CAN L	CAN BUS high
	29	VS CAN (DC 24 V)	CAN BUS supply voltage
	30	OUT3 R	Digital output 3, resistance monitoring
	31	OUT3 B	Digital output 3, contact B
	32	OUT3 A	Digital output 3, contact A
	33	OUT2 R	Digital output 2, resistance monitoring
	34	Shield	Housing/Screen

Figure 38: 2x terminal strip, 17-pin

NOTE
The OUT1 B, OUT2 B and OUT3 B connections are connected internally.

Yellow bypass switch at the indoor system plug (2043973)

By moving the bypass switch, you can select which pins provide the supply voltage for the outputs.

NOTE
If the bypass switch is set as "bridge", a higher power is consumed via the supply cable of the sensor (supply power for sensor).

Table 13: Supply voltage outputs

	Bottom S1 position (default)	Top S1 position (CAN operation)
		
V_S OUT	Via Vs from Pin18	Via external from Pin29
GND OUT	Via GND from Pin19	Via external from Pin26

6.4.2 Connections of the LMS11x, LMS15x and LMS16x

Table 14: Pin assignment of the "Power" connection

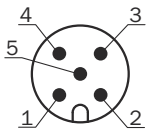
Male/female connector	Contact	Short form	Signal description
 **Figure 39:** M12 male connector, 5-pin, A-coded	1	VS	Device supply voltage
	2	VS heat.	Heating supply voltage
	3	GND	Device ground
	4	Reserved	Do not use!
	5	GND heat.	Heating ground

Table 15: Pin assignment of the "Data" connection

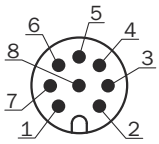
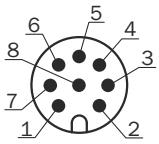
Male/female connector	Contact	Short form	Signal description
 **Figure 40:** M12 male connector, 8-pin, A-coded	1	RxD HOST	Receiver RS-232 (Host interface)
	2	TxD HOST	Sender RS-232 (Host interface)
	3	CAN H	CAN BUS high
	4	CAN L	CAN BUS low
	5	GND RS/CAN	RS-232/CAN ground
	6	IN1	Digital input 1
	7	IN2	Digital input 2
	8	GND IN1/IN2	Digital input 1 and 2 ground

Table 16: Pin assignment of the "I/O" connection

Male/female connector	Contact	Short form	Signal description
 **Figure 41:** M12 female connector, 8-pin, A-coded	1	LMS11x: INC1 A LMS15x/LMS16x: NC1 A or IN41)	LMS11x: Encoder 1 input, contact A LMS15x/LMS16x: Encoder 1 input, contact A or input 4
	2	LMS11x: INC1 B LMS15x/LMS16x: NC1 B or IN3 1)	LMS11x: Encoder 1 input, contact B LMS15x/LMS16x: Encoder 1 input, contact B or input 3
	3	GND INC1	Encoder 1 ground
	4	OUT1 A	Digital output 1, contact A
	5	OUT2 A	Digital output 2, contact A
	6	OUT3 A	Digital output 3, contact A
	7	OUT1...3 B	Digital output 1 .. 3, contacts B
	8	OUT1...3 R	Digital output 1 .. 3, resistance monitoring

NOTE
The OUT1 B, OUT2 B and OUT3 B connections are connected internally.

Table 17: Pin assignment of the "Ethernet" connection

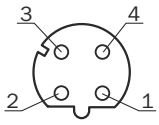
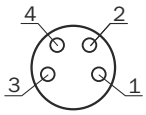
Male/female connector	Contact	Short form	Signal description
 **Figure 42:** M12 female connector, 4-pin, D-coded	1	TX+	Sender+
	2	Rx+	Receiver+
	3	TX–	Sender–
	4	Rx–	Receiver–

Table 18: Pin assignment of the "AUX" connection

Male/female connector	Contact	Short form	Signal description
 **Figure 43:** M8 female connector, 4-pin	1	RxD AUX	Receiver RS-232 (auxiliary interface)
	2	Reserved	Do not use
	3	GND RS	RS-232 ground
	4	TxD AUX	Sender RS-232 (auxiliary interface)

6.5 Preparing electrical installation

6.5.1 SELV supply voltage

The following type-dependent supply voltages in accordance with IEC 60 364-4-41 and power outputs from the user's power supply unit are required for commissioning and operation of the device:

Table 19: Supply voltages and power consumption of the device

Device	Supply voltage	Power output
LMS10x LMS11x LMS15x LMS16x	DC 10.8 V ... 30 V	LMS100: Typ. 10 W LMS111: Typ. 55 W at maximum heat output LMS151: Typ. 55 W at maximum heat output LMS161: Typ. 55 W at maximum heat output

Device	Supply voltage	Power output
LMS12x	DC 9 V ... 30 V	LMS12x: Typ. 10 W
LMC12x		LMC12x: Typ. 10 W
LMS13x		LMS13x: Typ. 55 W at maximum heat output
LMS14x		LMS14x: Typ. 55 W at maximum heat output
LMS173		LMS173: Typ. 10 W
LMC13x		LMC13x: Typ. 55 W at maximum heat output

WARNING
Risk of injury due to electric current!

If the supply voltage is produced by a power supply unit, then insufficient electrical separation between input and output current circuit of the power supply unit may lead to an electric shock.

- Only use a power supply unit whose output circuit has safe electrical separation from the input circuit. For example with a safety transformer in accordance with EN 61558-1.

Protecting the supply line

To ensure protection against short-circuits/overload in the incoming supply lines, the wire cross-sections used must be appropriately selected and protected.

The device must be protected with the following fuse values in the incoming circuit at the beginning of the supply line(s) depending on the supply concept:

- Isolated supply voltages for sensor and heating. For the sensor there is a maximum current consumption of the device at the lowest permissible supply voltage, for the heating circuit it is the other way round. Recommended fuse value: Device: 2 A slow-blow, heating 2 A slow-blow
- Common supply voltage for device and heating. This results in a maximum current consumption of the device at the highest permissible common supply voltage – the recommended fuse value for the heating in this case: Device including heating: 4 A slow-blow

Device isolator – installation of a switch in the supply line

A switch, emergency stop switch or the like must be installed in the supply line to the device in order to be able to safely disconnect the device from the supply voltage in the event of a fault. If the device and the heating are supplied via isolated supply voltages, both supply lines must be routed via the switch.

6.5.2 Conductor cross-sections

Table 20: Required conductor cross-sections

Connection of	Minimum cross-section
Supply voltage	0.25 mm² if local voltage supply (power supply unit) is in the immediate proximity
	1.0 mm² at maximum length of cable of 220 m when connected to existing 24 V DC network
Digital outputs	0.25 mm², maximum length of cable 50 m at 0.5 mm²
Data interfaces	0.25 mm²

- All connections must be wired with copper cables.
- All communication cables must be stranded and shielded.

NOTE
With the LMS10x, the external diameter of the joint cable must not exceed 9 mm due to the cable entry.

NOTE
If you use flexible connecting cables with strands to connect to the terminals of the LMS10x/LMS12x/LMS173, then you must not use ferrules.

6.5.3 General Requirements of the Data Interfaces

The following table shows the maximum recommended lengths of cables depending on the selected data transmission rate.

Interface type	Data transmission rate	Maximum length of cable
RS-232	115,200 Bd	10 m
CAN-Bus [1]	1 MBit/s	40 m

1) With corresponding line termination, termination according to the respective specification.

- Use shielded cables (twisted pairs) with at least 0.25 mm².
- To prevent interference factors, do not lay data cables over a longer route in parallel with voltage supply cables and motor cables, for example in cable ducts.

6.5.4 Cable reserve on system plug

Allow for sufficient cable reserve of the supplied cables at the system plug. You can easily exchange the device with the cable reserve if needed.

Keep the cable reserve only long enough that the system plug cannot be accidentally plugged into an adjacent device when replacing the device! This prevents a device with an incorrect configuration being put into operation. Experience has shown that 200 to 300 mm of cable reserve on the device is ideal.

The reserve cable should be laid as a drip loop so no moisture (e.g., condensation) is directed towards the device but instead drips off the cable beforehand.

6.5.5 General conditions for data interfaces

The following table shows the maximum recommended length of cables depending on the selected data transmission rate.

Table 21: Maximum lengths of cables for the data interfaces

Interface type	Transfer rate	Maximum length of cable
RS-232	115.2 kBd	2 m
	38.4 ... 57.6 kBd	3 m
	Max. 19.2 kBd	10 m
RS-422	Max. 115.2 kBd	500 m
	Max. 38.4 kBd	1,200 m

NOTE
Use shielded cables (twisted pairs) with at least 0.25 mm².

To prevent interference factors, do not lay data cables over a longer route in parallel with voltage supply cables and motor cables, in cable channels, for example.

6.6 Performing the electrical installation

6.6.1 Connection to AUX and to the Ethernet interface of the device

Pre-assembled cables are available for configuring the device using the AUX and the Ethernet interface.

NOTE

The recommended connecting cables and their associated technical data can be found on the online product page.

The product page can be accessed via the **SICK Product ID**: **pid.sick.com/{P/N}/{S/N}**

{P/N} corresponds to the part number of the product, see type label.

{S/N} corresponds to the serial number of the product, see type label (if indicated).

6.6.2 LMS10x/LMS12x/LMS173/LMC12x: Wiring the system plug

NOTE

Reduced enclosure rating! If the system plug is removed, the device no longer complies with enclosure rating IP65.

- To prevent damage from penetrating moisture and dirt, only open the system plug in a clean and dry environment.
- If necessary, pre-wire and mount the system plug in a suitable environment.

Prerequisites on device for enclosure rating IP65

- The system plug is attached to the device, its 4 screws are tightened.
- The cables in the cable entries correspond to the suitable diameter and are locked with the coupling nut.
- A corresponding cable is plugged into the M12 round connector (Ethernet) and locked. If the connection is not used, it must be covered with a tightly-fastened protective plug (as in the delivery condition).
- The black rubber cover of the auxiliary interface (“AUX” connection) on the front of the device is connected and flush with the device.

1. Make sure that the voltage supply for the device is switched off.
2. Remove system plug on the rear side of the device. To do so, loosen the two fixing screws (see figure 56, page 85) and carefully pull the system plug up out of the device vertically.
3. The PG7 cable entry (metal) has a ground connection to the device. If a shielded connecting cable is used, connect the shielding braid of the cable to the cable entry. To do this, shorten the shielding braid before the fitting of the cable entry and push it back over the plastic insert of the cable entry.
4. Unscrew the fitting of the PG7 cable entry.
5. Feed cable for the supply voltage, data interface (RS-232) and the digital outputs with a maximum outer diameter of Ø 5.6 mm through the plastic insert of the PG7 cable entry.
6. Connect the wires to the two terminal blocks without tension and without pulling, see "Connections of the LMS10x", page 55.
7. If necessary, connect the shielding braid of the cables to the cable entry.
8. Attach and tighten the fitting of the PG7 cable entry.
9. Carefully reconnect the system plug to the device.
10. Tighten the fixing screws of the system plug.

6.6.3 LMS11x/LMS13x/LMS15x/LMS16x/LMC13x: Connecting M12 round connectors

NOTE

Prerequisites on device for enclosure rating IP67

- The device is only connected to the M12 round connectors provided.
- Use connecting cables with M12 round connectors that are compliant with enclosure rating IP67.
- The SICK cables attached to the M12 round connectors are locked. Any connections that are not being used must be fitted with protective caps or plugs that are screwed tight (as in the delivery condition).
- The black cover of the USB auxiliary interface (“AUX” connection) on the front of the device must be flush with the device.

Pre-assembled cables can be ordered as accessories for connection to the M12 round connectors. These cables consist of the round connector and a 5, 10, or 20 m cable with an open end.

Torque screwdrivers with a defined torque are available as accessories for tightening and loosening M12 plug connections on the system plug.

NOTE

The recommended connecting cables and their associated technical data can be found on the online product page.

The product page can be accessed via the **SICK Product ID**: **pid.sick.com/{P/N}/{S/N}**

{P/N} corresponds to the part number of the product, see type label.

{S/N} corresponds to the serial number of the product, see type label (if indicated).

6.6.4 Wiring the inputs and outputs

Wire non-floating digital inputs (related to supply voltage)

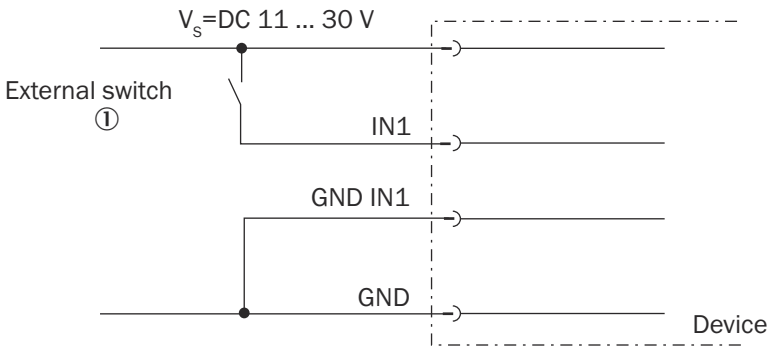


Figure 44: Digital input, e.g. IN 1 wired non-floating

① External Switch

NOTE

The inputs require a switching voltage of at least 11 V. For this reason, the supply voltage must be at least 11 V.

Wire digital floating inputs

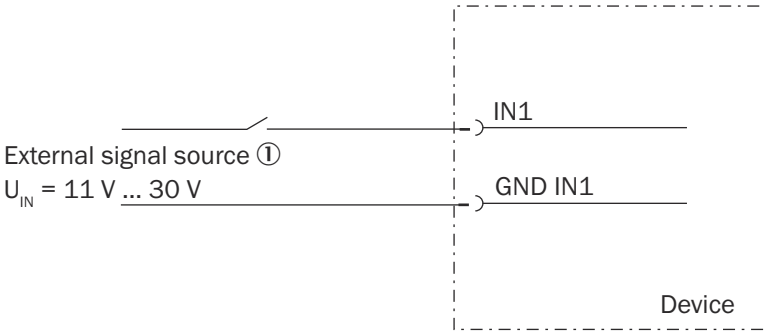


Figure 45: Digital input, e.g. IN 1, wired floating

① External signal source

Wiring encoder inputs

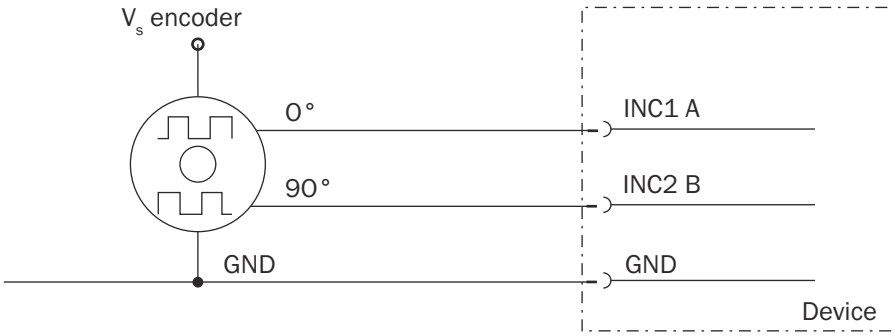


Figure 46: Wiring encoder inputs

Connect the digital outputs of the LMS10x/LMS11x/LMS15x/LMS16x to a PLC with a non-floating connection

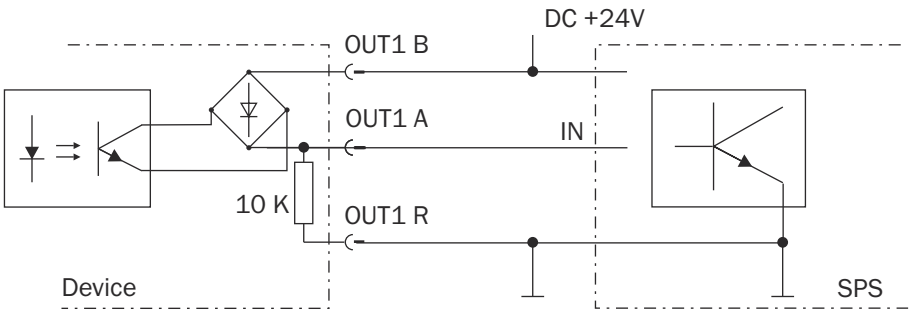


Figure 47: Connect digital outputs, e.g. OUT 1, to a PLC, non-floating (PNP)

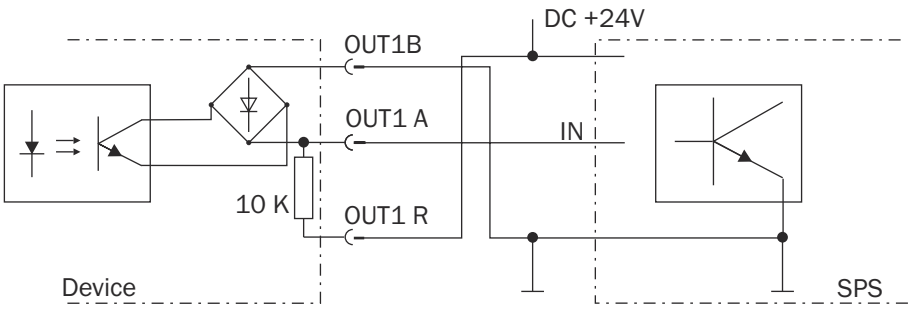


Figure 48: Connect digital outputs, e.g. OUT 1, to a PLC, non-floating (NPN)

Connect the digital outputs of the LMS10x/LMS11x/LMS15x/LMS16x to a PLC with a floating connection

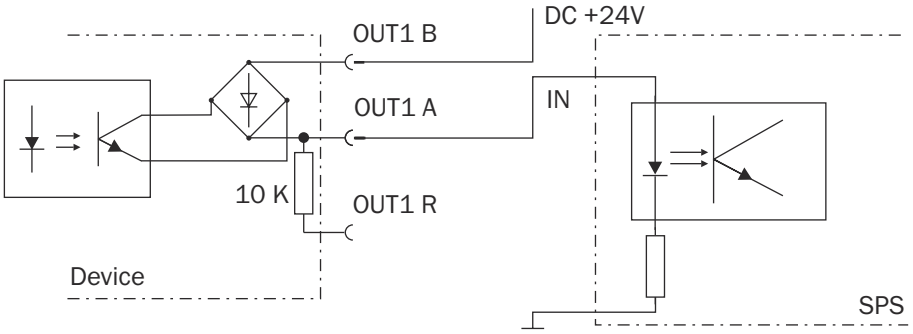


Figure 49: Connect digital outputs, e.g. OUT 1, to a PLC, floating (PNP)

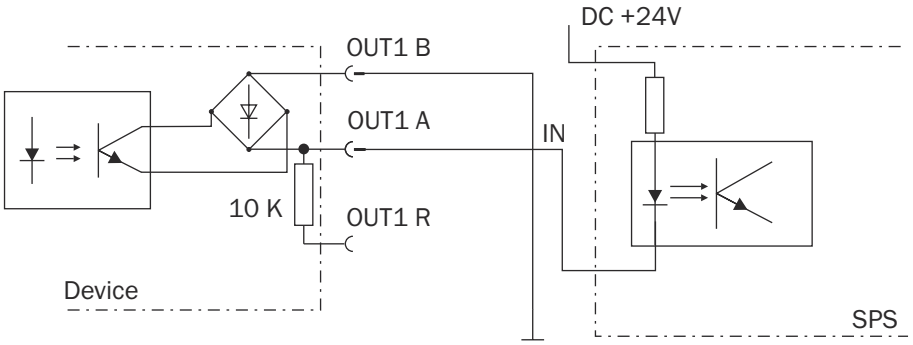


Figure 50: Connect digital outputs, e.g. OUT 1, to a PLC, floating (NPN)

Connecting to an object protection system

NOTE

LMC12x/LMC13x VdS (object protection) Information on connecting the devices to an object protection system can be found in the document “Technical information (VdS Installer Instructions)”, Part no.: 8013748

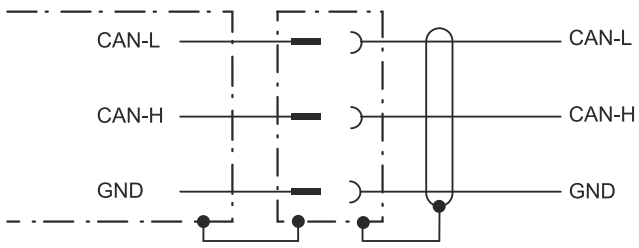
LMS12x/LMS13x/LMS14x Security Information on connecting the devices to an object protection system can be found in the document “Technical information (Security Installer Instructions)”, Part no.: 8014486

Wiring of the CAN interface (type-dependent)

A shielded "twisted pair" cable is required to connect the CAN interface. The 120 Ω terminator must be connected.

NOTE

Observe the maximum length of cable according to see "General conditions for data interfaces", page 60.

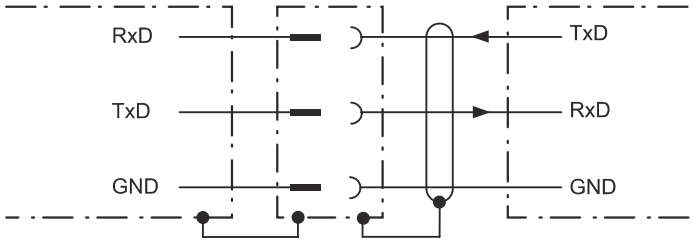


Wiring of the RS-232 interface

A shielded cable is required to connect the RS-232 interface.

NOTE

Observe the maximum length of cable according to see "General conditions for data interfaces", page 60.



7 Commissioning

7.1 Overview of the commissioning steps

1. Install the SOPAS ET configuration software.
2. Establish communication with the device.
3. Create an application-specific parameter set with SOPAS ET and save it permanently in the non-volatile device memory and on the computer.
4. Test the device for correct functioning.

7.2 SOPAS ET configuration software

The interactive configuration is performed using the SOPAS Engineering Tool (SOPAS ET) software. You can use this configuration software to parameterize and test the measuring properties, evaluation behavior, and output properties of the system as required. The configuration data are stored and archived as a parameter set (project file) on the computer.

NOTE
The most up-to-date version of the SOPAS ET software can be downloaded from **www.sick.com/SOPAS_ET**. The respective system requirements for installing SOPAS ET are also specified there.

Downloading and installing SOPAS ET

1. Start computer.
2. Download and install the latest version of the SOPAS ET configuration software, as well as current device description files (*.sdd) for the device variant from the online product page for the software by following the instructions provided there.
3. In this case, select the “Complete” option as suggested by the installation wizard. Administrator rights may be required on the computer to install the software.
4. When the installation is finished, start the **SOPAS ET** program option. Path: Start > Programs > SICK > SOPAS ET Engineering Tool > SOPAS.
5. Establish a connection between SOPAS ET and the device. In order to do this, select the desired communication interface for searching in the connection wizard. (Default Ethernet address: IP address: 192.168.0.1, subnet mask: 255.255.255.0).

✔ SOPAS ET establishes communication with the device and loads the associated device description file.

SOPAS ET default settings

Table 22: SOPAS ET default settings

Parameters	Value
Program interface language	English (you must restart the software after making any changes)
Units of length	Metric
User group (operating level)	Machine operator
Download the parameters to the device	Immediately after a change, temporarily in the RAM of the device
Upload the parameters from the device	Automated after online switching
Window arrangement	3 (project tree, help, working area)
Serial communication	COM1: 9,600 Bd / 19,200 Bd, 8 data bits, no parity, 1 stop bit

7.3 Establishing communication with the device

NOTE
To communicate via Ethernet TCP/IP, the TCP/IP protocol on the computer must be active.

When connecting a PC/host, please adhere to the following order:

1. Connect the computer to the device with a data cable.
2. Switch on computer.
3. Switch on supply voltage of the device.

✔ The device performs a self-test and initializes itself.

Connecting data interfaces

Connect the device to the computer via one of the following data interfaces:

- Ethernet
- RS-232

Starting SOPAS ET and calling up scan assistant

1. Start SOPAS ET.
2. In the main window under **Scan Assistant**, click on the **Configuration** button.

✔ The **Scan Assistant** dialog window appears.

Configuring the Ethernet connection

NOTE
Deactivate all programs on your computer to access Ethernet or TCP/IP.

1. In the **Scan Assistant** dialog box under **Internet Protocol, IP Protocol**, select the **Use Auto-IP** checkbox.
2. Confirm the settings in the **Scan Assistant** dialog window by clicking **OK**.

✔ The **Scan Assistant** dialog window closes.

Configuring serial connection

1. In the **Scan Assistant** dialog box under **Serial Connection**, **Standard Protocol**, select the **Activate Serial Communication** checkbox.
2. Click the **Advanced...** button.
3. Select the following **PORT Settings**: 8 data bits, no parity, 1 stop bit.
4. Confirm the settings with **OK**.

✔ The **Advanced Scan Settings** dialog window is closed.

5. Confirm the settings in the **Scan Assistant** dialog window by clicking **OK**.

✔ The **Scan Assistant** dialog window closes.

Performing a scan

1. In the **Scan Assistant** dialog window, click the **Start Scan** button.
2. Select the listed devices and confirm by clicking **Add Device**.

✔ The connection is used to search for connected devices. SOPAS ET adds the devices found to the project tree and loads the current parameter set to the device via upload.

7.4 Initial commissioning

The device is adjusted to the situation on site using the SOPAS ET configuration software. To do this, SOPAS ET creates an application-specific parameter set of parameter values. For this purpose, the parameter set can first be loaded from the device (upload, initial commissioning: factory default setting). Or it is created independently, either based on the factory default settings or as a modification of an already existing parameter set of the device of the same type and firmware version.

The parameter set is then loaded into the device (download). This either happens immediately (**Download immediately** option in SOPAS ET) or manually (**Download all parameters to the device** command in SOPAS ET).

NOTE

After completing configuration, the changed parameter set must be permanently stored in the non-volatile memory of the device. As part of a backup concept for the created parameter values, we recommend saving the parameter set as a project file (*.sopas file with configuration data) on the computer for archival purposes.

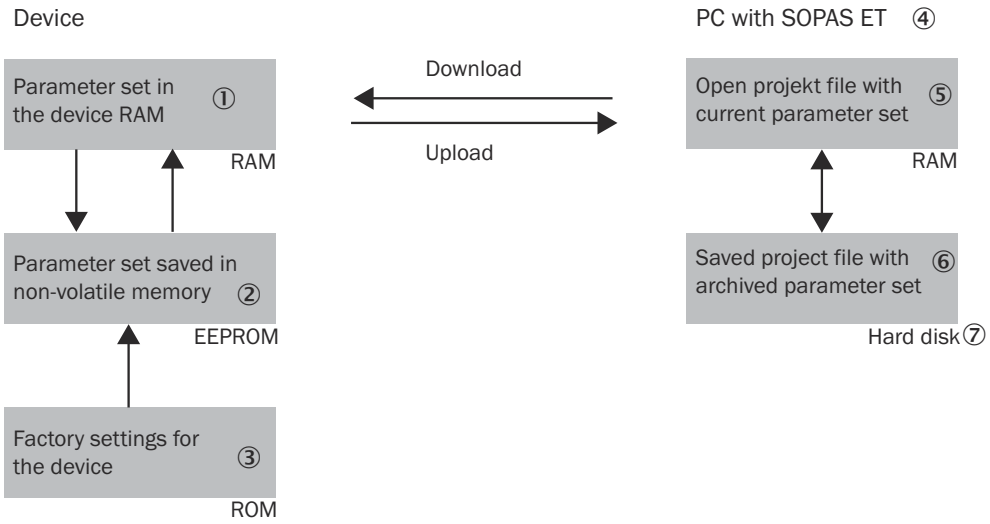


Figure 51: Data storage principle

① Parameter set in the working memory of the device
② Permanently saved parameter set
③ Factory-set pre-settings of the device
④ Computer with SOPAS ET
⑤ Opened project file with current parameter set
⑥ Saved project file with archived parameter set (*.spr)
⑦ Hard drive

Device configuration

You can configure the device in two ways:

- Interactively using SOPAS ET. This section describes the interactive configuration.
- Using configuration telegrams, see "Communicating data via telegrams", page 46.

Interactive configuration with SOPAS ET

All configurable parameters of the device are collected together in a corresponding device description (.sdd file) for SOPAS ET. You can access these parameters via the project tree for the device description.

A content-sensitive explanation of each of the parameters is available directly next to the parameter via the **?** button or in the online help (**F1** key). The **Parameter Info** display window lists the valid range of values and the default setting (right-click when the pointer is located over the parameter).

NOTE
Software access to the device is protected by user levels and passwords. After successfully configuring the device, you should change the passwords so they can fulfill their protective function.

Table 23: Passwords

User levels	Password
Maintenance	`main`
Authorized Client	`client`
Service	`servicelevel`
Operator	-

NOTE
Change the passwords during initial commissioning to protect your device.

A higher user level can change the password of a lower user level.

NOTE
If the password for the Service user level has been lost: see "Resetting the password for the Service user level", page 77.

NOTE
LMC12x/LMC13x VdS (object protection) Access to these devices using the SOPAS ET configuration software is provided in the document “Technical information (VdS Installer Instructions)”, Part no.: 8013748

LMS12x/LMS13x/LMS14x/LMS17x Security Access to these devices using the SOPAS ET configuration software is provided in the document “Technical Information (Security Installer Instructions)”, Part no.: 8014486

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

NOTICE
Loss of configuration data in the connected device

1. Do not switch off the voltage supply during the configuration of the device. Otherwise all parameters not yet saved permanently will be lost.

1. In the **Options** menu, select the **Log into device** command and log in as an **Authorized Client** using the password `client`.
2. Configure the device for the required application using the parameters in SOPAS ET. Help on using the program user interface as well as for the different options can be found in SOPAS ET.

Export mode or quickstart

The LMS12x provides a quickstart configuration mode that can be used to quickly and easily configure the parameters of the LMS12x for the specific object protection requirements.

The LMS10x/LMS11x/LMS15x/LMS16x only provides the expert mode. The expert mode can also be used with the LMS12x.

Resetting the configuration

NOTE

To reset the device to the delivery condition, use the **Factory defaults** option in SOPAS ET.

7.5 Connecting and performing a test measurement

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

1. In the project tree, select **LMS**..., **Monitor**, **Scan view pro**.
2. Compare the measurement line with the required result.

NOTE

- The **Pro scan display** in the **Monitor** depends on the available processing power of the computer and is not a real time display. For this reason not all measured values are displayed. The same restriction also applies to saving the displayed measured values in a file.
- The monitor displays the measured values unfiltered, i.e. the effect of filters cannot be checked using the monitor.

3. After successful completion of the test measurement, save the configuration permanently in the device: Menu **LMS...**, **Parameters**, **Save permanently**.

8 Maintenance

8.1 Cleaning

NOTICE
Equipment damage due to improper cleaning.

Improper cleaning may result in equipment damage.

- Only use recommended cleaning agents and tools.
- Never use sharp objects for cleaning.

→ Clean the optics cover at regular intervals and in the event of contamination with a lint-free lens cloth and plastic cleaning agent. Rinse off coarse dirt first with water. The cleaning interval essentially depends on the ambient conditions.

8.2 Maintenance plan

During operation, the device works maintenance-free.

NOTE
No maintenance is required to ensure compliance with the laser class.

Depending on the assignment location, the following preventive maintenance tasks may be required for the device at regular intervals:

Table 24: Maintenance plan

Maintenance work	Interval	To be carried out by
Check device and connecting cables for damage at regular intervals.	Depends on ambient conditions and climate.	Specialist
Clean housing and optics cover.	Depends on ambient conditions and climate.	Specialist
Check the screw connections and plug connectors.	Depends on the place of use, ambient conditions or operating requirements. Recommended: At least every 6 months.	Specialist
Check the mounting accessories and vibration dampers used.	Depends on the place of use, ambient conditions or operating requirements. Recommended: At least every 6 months.	Specialist
Check that all unused connections are sealed with protective caps.	Depends on ambient conditions and climate. Recommended: At least every 6 months.	Specialist

8.3 Device replacement

Faulty or damaged devices must be dismantled and replaced with new or repaired devices of the **same type**.

As all external cable connections terminate in the system plug or the round connectors, there is no need to repeat the electrical installation when the device is replaced. The replacement device can simply be connected.

NOTE

Disconnect the power to the system!

While the device is connected, the system may start unintentionally.

Make sure that the entire system is disconnected from the voltage supply during all exchange processes.

8.3.1 Parameter cloning (LMS10x/LMS12x/LMS17x)

NOTE

The parameter cloning function can only be used between devices of the same type.

Functionality

When the supply voltage is switched on, the device starts up and organizes its current parameter values as follows:

- Initial commissioning of a brand-new device: The device starts up with the default setting. The cloning parameter memory of the system connector contains the default setting.
- For each adjustment to the function configuration of the device: The device saves the parameter values permanently in its internal, non-volatile parameter memory. Additionally, it always saves a permanent copy of the parameter set in the cloning parameter memory of the system plug. (This is initiated using SOPAS ET and the **Save permanently** function.) The parameter set remains saved in the system plug even if the voltage supply is interrupted.
- The next time it is started up, the device transfers the parameter set from the cloning parameter memory to its RAM see "Initial commissioning", page 68.

If a device is replaced, after start-up the replacement device checks whether there is a plausible parameter set in the system plug. If so, the device loads this parameter set to its RAM ("cloning plug gains" strategy).

As the existing system plug continues to be in use, the LMS10x/LMS12x/LMS17x therefore make it possible for the last parameter values used in the system plug to be passed on to a replacement device of the same type automatically (cloning).

Cloning involves the replacement device accepting all the adjustable parameters, including field evaluation settings and the IP address. Accordingly, the process overwrites all the existing parameter values in the device. The exceptions in this case are the following parameters, which are always device-specific: serial number, MAC address, operating hours counter, and error memory.

NOTE

Testing the device for application suitability by means of acceptance after device replacement

If the replacement device is going on to be used unchanged, with the existing system plug and the parameter values saved in it for the application, acceptance by qualified safety personnel is not required. However, a test based on the regulations for daily testing, or a functional test, must be carried out.

If the replacement device will continue to be used with a system plug that is different from the previous one, the existing configuration must be transferred to the sensor using SOPAS ET. In this case, acceptance by qualified safety personnel may be required (depending on the system; e.g., Security).

8.3.2 Replacing the device

LMS10x / LMS12x / LMS17x

NOTICE

Danger of damage due to environmental influences!

While the system plug is removed from the device, the device and contacts are unprotected and the enclosure rating no longer meets IP65.

- Replacement of devices with removable system plug (its removal and installation) must only be performed in a clean, dry and dust-free environment. For this purpose, move the device from the outdoor area to a corresponding indoor area, if necessary.
- To prevent damage to the devices, firmly attach the existing system plug to the replacement device immediately after replacement.
- Also, immediately attach the remaining, non-wired system plug for the replacement device firmly to the removed (and potentially defective) device.
- Make sure that the inlay sealing, the supporting surface opposite, and all the electrical contacts of the system connector are free from dirt and moisture.

1. Switch off the voltage supply.
2. Dismantle the device from the bracket. If necessary, mark the alignment appropriately.
3. If the device is provided with a protection hood/weather protection hood, remove it from the device.
4. Recommendation: Turn the device upside down while disconnecting the system plug to avoid dirt falling into the electrical connections.
5. Loosen and remove the old system plug including wired cables/fixed cables at the sensor.
6. Make sure that the inlay sealing, the opposite supporting surface, and all the electrical contacts are undamaged and free from dirt and moisture. If necessary, clean the corresponding elements.
7. Remove the unwired system plug from the corresponding replacement device.
8. Replace the previous system plug including cables at the rear of the replacement device and mount it.
9. If necessary, remount the protection hood/weather protection hood to the replacement device.
10. Re-mount the replacement device to the bracket see "Mounting", page 48 and check alignment.
11. Automatically configure replacement device see "Parameter cloning (LMS10x/LMS12x/LMS17x)", page 72.

LMS11x/LMS13x/LMS14x/LMS15x/LMS16x

1. Switch off the voltage supply.
2. Dismantle the device from the bracket. If necessary, mark the alignment appropriately.
3. If the device is provided with a protection hood/weather protection hood, remove it from the device.
4. Disconnect and pull out all connecting cables from the device.
5. Mount the replacement device see "Mounting", page 48.
6. Reconnect and tighten all connecting cables on the replacement device.
7. If necessary, remount the protection hood/weather protection hood to the replacement device.

8. Re-mount the replacement device to the bracket and check alignment.
9. Configure replacement device manually. This is done using the parameter set of the previous device stored on your computer according to the backup concept see "Commissioning", page 66.

9 Troubleshooting

9.1 Repairs

Repair work on the device may only be performed by qualified and authorized personnel from SICK AG. Interruptions or modifications to the device by the customer will invalidate any warranty claims against SICK AG.

9.2 Returns

→ Only send in devices after consulting with SICK Service.
→ The device must be sent in the original packaging or an equivalent padded packaging.

NOTE
To enable efficient processing and allow us to determine the cause quickly, please include the following when making a return:

- Details of the contact person
- Description of the application
- Description of the fault that occurred

9.3 General faults, warnings, and errors

The following sections describe possible faults and troubleshooting measures to correct them. For faults that cannot be rectified using the information below, please contact SICK Service. To find your agency, see the final page of this document.

NOTE
Before calling, make a note of all type label data such as type designation, serial number, etc., to ensure faster assistance.

9.3.1 LED fault indicators

NOTE
In the case of the LMS12x, the status indicators are switched off so that the device, as far as possible, will not be recognized as part of an object protection system. They are turned on in Walk-through test mode.



Table 25: LED fault indicators

Display	Possible causes	Troubleshooting
③ and ① off	Supply voltage missing or too low	• Check the voltage supply and activate, if necessary.
② lights up	Optics cover is contaminated, operation is still guaranteed	• Clean the optics cover.
② flashes at 1 Hz	Optics cover is contaminated, operation is not guaranteed	• Clean the optics cover.
② flashes at 4 Hz	System error	• Note the error indicated on the 7-segment display or run diagnostics using SOPAS ET. • Switch off the device and turn it on again, if necessary.

9.3.2 Displays in the 7-segment display

Display	Possible causes	Troubleshooting
	Not an error	• Device is in measuring mode
E.	Device error	• Contact SICK Service.
– flashes	Motor starts	• Not an error
. flashes	IDLE mode, outputs are in OFF state, laser is switched off.	• Not an error. If the criteria for IDLE mode are withdrawn, operational readiness is restored.
Each segment flashes once in sequence	Device powers up.	• Wait until the device has started.
Other displays	Display is controlled by system controller	• See the system controller documentation or contact the supplier of the system controller.

9.3.3 Detailed fault analysis

The device reports any errors that occur in a number of ways. Fault output is staggered, allowing for an increasingly detailed level of analysis:

Telegram

Communication errors can occur when transmitting telegrams to the device. The device then returns a fault code. A description of the communication errors can be found in the Telegram Listing, see "Telegram listing (EN)", page 88.

System log

Information about the status of the device is displayed in the system log.

SOPAS ET: **Project tree**, **LMS...**, **Service**, **System Status**

Field evaluation monitor

With the field evaluation monitor, you can analyze if and how detection fields are interrupted and how the outputs of the device behave.

SOPAS ET: **Project tree**, **LMS...**, **Monitor**, **Field evaluation monitor**.

Field evaluation logging

With field evaluation logging, you can record any object detection that occurs. The field evaluation logging helps with diagnosing longer periods of time, analyzing faults, or optimizing processes.

SOPAS ET: **Project tree**, **LMS...**, **Monitor**, **Recording of object detections**.

9.3.4 Resetting the password for the Service user level

If you have forgotten the password of the **Service** user level, you can reset it with the assistance of SICK.

NOTE

The responsible SICK sales company or the responsible SICK service partner carefully checks each code request to reset the password. A risk of deception by third parties nevertheless exists. The operating entity should therefore take suitable security measures.

The operating entity should also take suitable measures to limit, as best as possible, access to the product. This includes, in particular, physical access as well as access to the software interfaces of the product.

Requesting an unlock code

1. Open SOPAS ET.
2. Open the device window.
3. Open the device name > **Password** > **Reset Service password**.

✔ The **Reset password** window appears.

4. Enter the relevant data.

   NOTE | Do not press **Generate** if an unlock code has already been requested from SICK. Only press this button if a new device code is required when inquiring again.

5. Click **Generate e-mail with data**.

✔ Your SICK subsidiary will create the unlock code based on the information provided and send it to you.
The code is only valid once for the reset process. You can close the window by clicking on the **x** without interrupting the reset process. If you select **Cancel** or enter an incorrect code several times, the current reset process is terminated. The requested code is no longer valid. The process must be restarted.

6. Wait for the unlock code: The dialog box can be closed and the device switched off.

Entering the unlock code

Prerequisite

- SICK has sent an unlock code.

1. Open SOPAS ET.
2. Open the device window.
3. Open the device name > **Password** > **Reset Service password**.

✔ The **Reset password** window appears.

4. Click **Next**.
5. Enter the code sent by SICK.
6. Click **Ok**.

✔ Password has been reset to the default password `servicelevel`. Parameters are not changed.

Assigning a new password for the Service user level

1. Open SOPAS ET.
2. Log on to the device with the **Service** user level and the default password `servicelevel`.
3. Open the device name > **Password** > **Change password**.
4. Assign the new password for the **Service** user level.

9.4 Disposal

If a device can no longer be used, dispose of it in an environmentally friendly manner in accordance with the applicable country-specific waste disposal regulations. Do not dispose of the product along with household waste.

NOTICE

Danger to the environment due to improper disposal of the device.

Disposing of devices improperly may cause damage to the environment.

Therefore, observe the following information:

- Always observe the national regulations on environmental protection.
- Separate the recyclable materials by type and place them in recycling containers.

10 Technical data

NOTE

The relevant online product page for your product, including technical data, dimensional drawing, and connection diagrams, can be downloaded, saved, and printed from the Internet.

The product page can be accessed via the **SICK Product ID**: **pid.sick.com/{P/N}/{S/N}**

{P/N} corresponds to the part number of the product, see type label.

{S/N} corresponds to the serial number of the product, see type label (if indicated).

Please note: This documentation may contain further technical data.

10.1 Features

Application	Indoor Outdoor Security
Light source	Infrared (905 nm)
Laser class	Laser class 1, EN 60825-1:2014+A11:2021; IEC 60825-1:2014 Complies with 21 CFR 1040.10 and 1040.11 except for conformance with IEC 60825-1 Ed. 3 as described in Laser Notice No. 56, dated May 8, 2019.
Aperture angle	270°
Scan rate	25 Hz 50 Hz
Angular resolution	At 25 Hz: 0.25° At 50 Hz: 0.5°
Scan field flatness	Typical taper error: ± 0.3° Typical skew: ± 1°
Heating	no (Indoor, LMS10x, LMS12x, LMC12x) yes (Outdoor, LMS11x, LMS13x, LMS14x, LMS15x, LMS16x, LMC13x)
Working range	0.5 m ... 20 m
Scanning range for 10 % remission	18 m ... 30 m
Scanning range for 90 % remission	20 m ... 50 m
Number of echoes evaluated	2

Working range diagrams

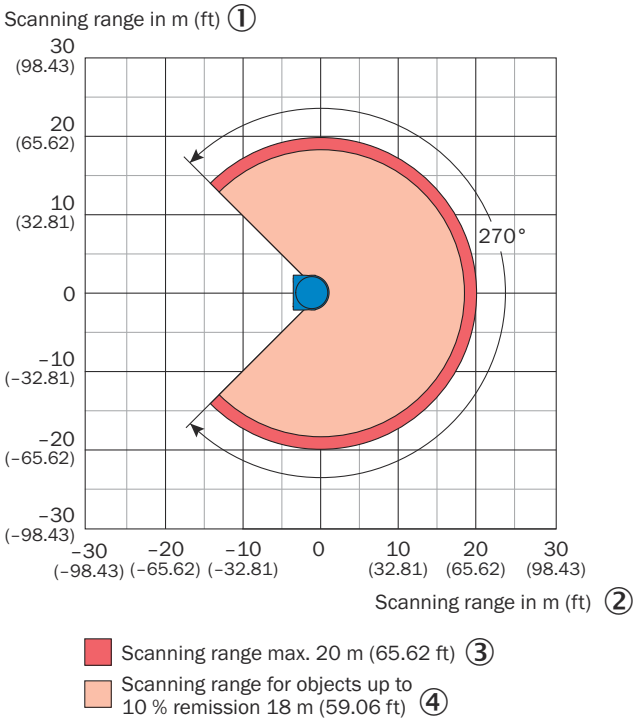


Figure 52: LMS10x/LMS11x/LMS12x/LMS13x/LMS17x working range, unit: m (ft), decimal separator: period

① Scanning range in m (ft)

② Scanning range in m (ft)

③ Maximum scanning range 20 m

④ Scanning range for objects with up to 10 % remission 18 m

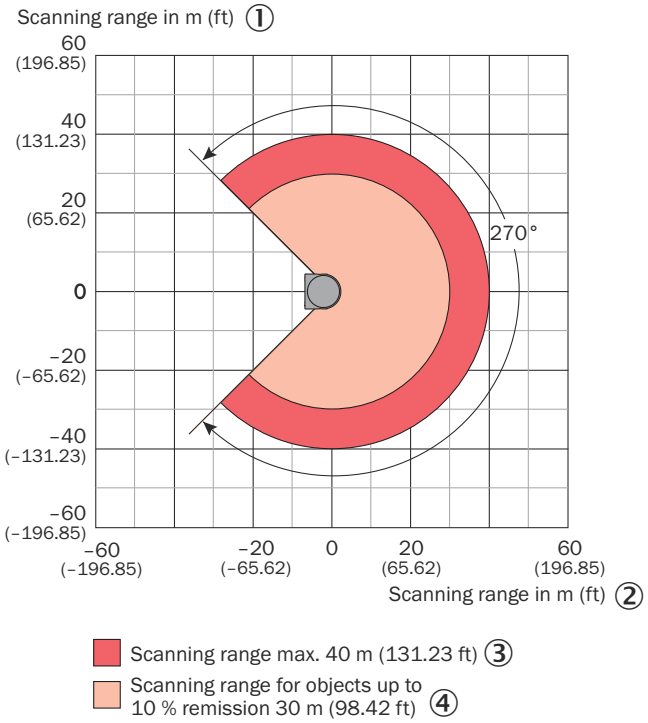


Figure 53: LMS14x working range, unit m (ft), decimal separator: period

① Scanning range in m (ft)

② Scanning range in m (ft)

③ Scanning range max. 40 m

④ Scanning range for objects with up to 10 % remission 30 m

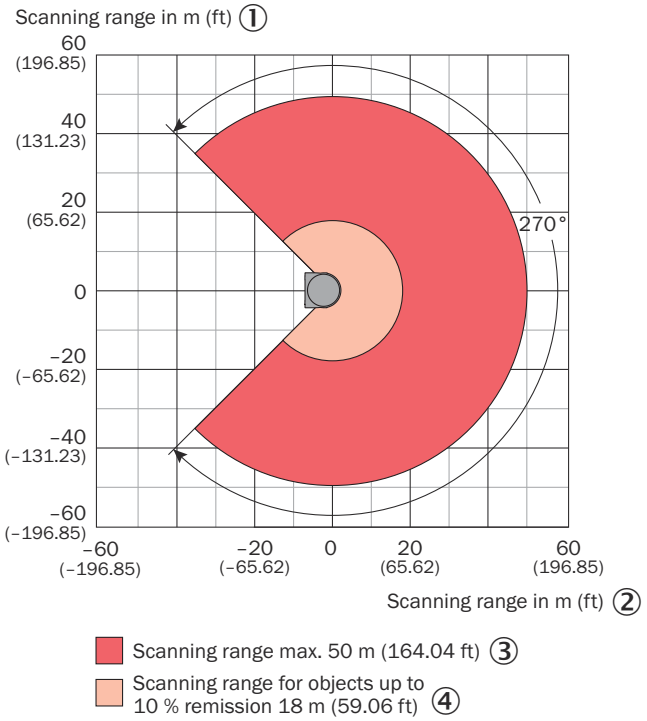


Figure 54: LMS15x working range, unit m (ft), decimal separator: period

① Scanning range in m (ft)

② Scanning range in m (ft)

③ Scanning range max. 50 m

④ Scanning range for objects with up to 10 % remission 18 m

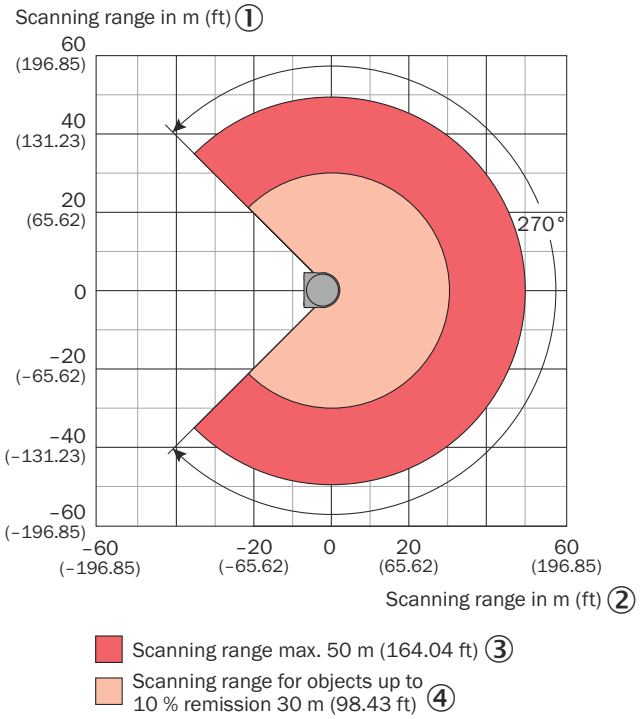


Figure 55: LMS16x working range, unit m (ft), decimal separator: period

① Scanning range in m (ft)

② Scanning range in m (ft)

③ Max. scanning range 50 m

④ Scanning range for objects with up to 10% remission: 30 m

10.2 Mechanics/Electronics

Electrical connection	1 x M12 round connector LMS161: 4 x M12 round connectors		
Electrical safety	According to EN 61010-1 (2010-10)		
Supply voltage	10.8 V DC ... 30 V DC		
Permissible residual ripple	± 5 %		
Power consumption	P_{max} combined: 35 W P_{max} sensor: 22 W P_{max} heating: - P_{max} output: 13 W (total output of all switching outputs)	P_{max} combined: 95 W P_{max} sensor: 22 W P_{max} heating: 60 W P_{max} output: 13 W (total output of all switching outputs)	P_{max} combined: 83 W P_{max} sensor: 22 W P_{max} heating: 60 W P_{max} output: 1 W (total output of all switching outputs)
Start-up current	The start-up current depends on the supply voltage and the length of cable. This will be a multiple of the continuous load current. Example: For 24 V on the device and a 5 m long cable, the peak start-up current is approx. 14 A. The rise/fall time ratio is approx. 12 µs / 240 µs.		

Screw terminal	Rigid wire cross-circuit: 0.14 ... 1.5 mm² Flexible wire cross-circuit: 0.14 ... 1.0 mm² Stripping length of the wires: 5 mm Screw tightening torque 0.22 ... 0.3 Nm	–
Housing material	GD ALSI12 3.2582.05	
Housing color	LMS1x0: RAL 5012 (blue) LMS1x1: RAL 7032 (pebble gray) LMS1x2: RAL 9005 (deep black) LMS1x3: RAL 9003 (signal white)	
Optics cover	Material: Polycarbonate, scratch-resistant coating on the exterior	
System plug (LMS10x and LMS12x/LMS173 Security)	GD ALSI12 3.2582.05, RAL 9005 (deep black)	
Cable entry (LMS10x and LMS12x/LMS173 Security)	Stainless steel/Plastic	
Enclosure rating	According to EN 60 529 (1991-10); A1 (2002-02) LMS10x LMS12x/LMS173 Security/LMC12x VdS: IP65 LMS11x/LMS13x/LMS14x/LMS15x/LMS16x/LMC13x VdS: IP67 [1]	
Protection class	III according to EN 61010-1 (2010-10)	
Weight	1.1 kg	
Dimensions	152 mm x 102 mm x 105 mm	

1) Under the following conditions: The SICK cables attached to the M12 connections must be screwed tight. Any electrical connections that are not being used must be fitted with protective caps or plugs that are screwed tight (as in the delivery condition). The M8 connection for the AUX interface on the front of the device is fastened.

10.3 Performance

Response time [1]	Typ. ≥ 20 ms
Power-up time	15 s – 60 s
Systematic error 1 Echo	± 30 mm ± 50 mm
Statistical error 1 Echo, all types except LMS14x Security	12 mm (1 m ... 10 m) 20 mm (10 m ... 20 m)
Statistical error 1 Echo, LMS14x Security	12 mm (0.5 m ... 10 m) 20 mm (10 m ... 20 m) 35 mm (20 m ... 40 m)
Integrated application	Field evaluation with flexible fields, perpendicular distance measurement
Filter	Echo filter, fog filter, particle filter, fixed average filter, hardware blanking window, software blanking window

1) Depending on the selected filter settings, scanning frequency, angular resolution and the object size.

10.4 Interfaces

Ethernet	TCP/IP, UDP/IP Function: Host Data transmission rate: 10/100 Mbit/s Standard IP address: 192.168.0.1

Serial (RS-232)	Function: Host/AUX Data transmission rate: 9.6 kBaud ... 115.2 kBaud
CAN bus	Function: Output extension 20 kbit/s, 500 kbit/s, 1 Mbit/s
Digital inputs	2 digital and 2 encoder inputs
Digital outputs	3
Optical displays	5 LEDs (additional 7-segment display)

10.5 Ambient data

Remission factor	3 % ... > 1,000 % (reflectors) [1]
Electromagnetic compatibility (EMC)	According to EN 61000-6-2 (2005-08), EN 61000-6-3 (2007-01)/A1 (2011-03)
Vibration resistance	According to IEC 60068-2-6 (1995-04) 10 Hz ... 150 Hz: amplitude 0.35 mm to 5 g, 20 cycles
Shock resistance	Single shock according to IEC 60068-2-27 (1993-03) 15 g, 11 ms, 6 shocks per axis Continuous shock according to IEC 60068-2-27 (1993-03) 10 g, 16 ms, 1000 shocks per axis
Ambient operating temperature	LMS10x LMS12x/LMS173 Security/LMC12x VdS: 0 °C ... +50 °C LMS11x/LMS15x LMS13x/LMC13x VdS: -30 °C ... +50 °C LMS16x/LMS14x Security: –40 °C ... +60 °C LMC12x VdS with mounting kit 1: -30 °C ... +45 °C LMC12x VdS with mounting kit 2: –30 °C ... +50 °C
Storage temperature	–30 °C ... +70 °C (max. 24 h) LMS16x/LMS14x Security only: –40 °C ... +70 °C (max. 24 h)
Relative humidity	Max. 95 %, non-condensing according to DIN EN 60068-2-61, method 1
Ambient light immunity	80 klx

[1] Corresponds to Diamond Grade 3000X™ (approx. 1,250 cd/lx × m²)

10.6 Dimensional drawings

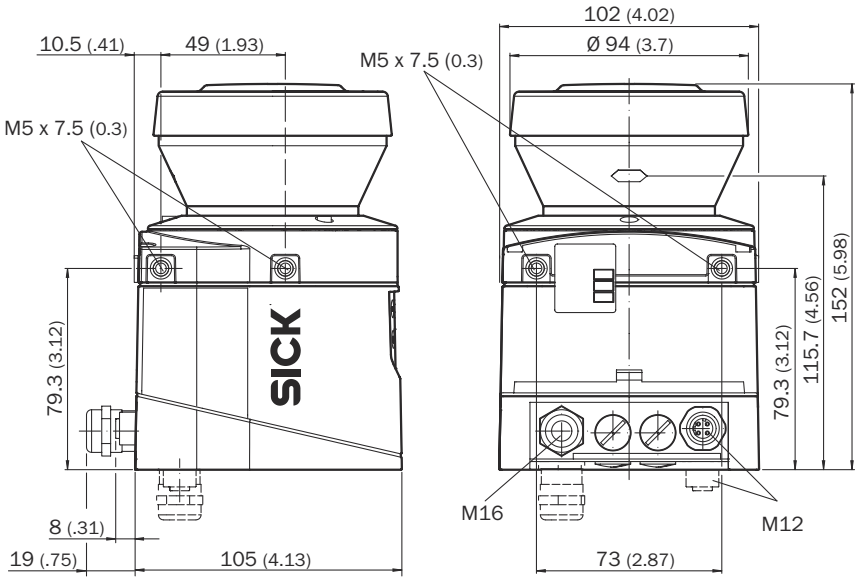


Figure 56: Device dimensions for LMS10x/LMS12x/LMS173/LMC12x; unit: mm (inch), decimal separator: period

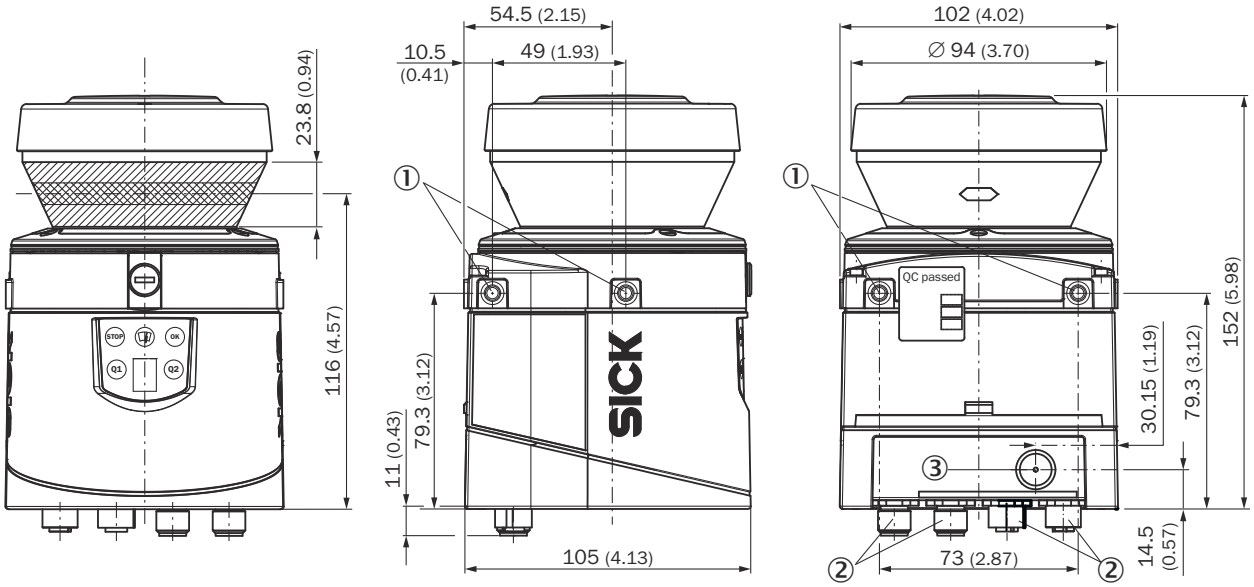


Figure 57: Device dimensions for LMS11x/LMS13x/LMS14x/LMS15x/LMS16x/LMS17x/LMC13x; unit: mm (inch), decimal separator: period

① M5 x 7.5 fixing hole

② M12 plug connector

③ Ventilation element (outdoor variants only)

11 Accessories

NOTE

On the product page you will find accessories and, if applicable, related installation information for your product.

The product page can be accessed via the **SICK Product ID**: **pid.sick.com/{P/N}/{S/N}**

{P/N} corresponds to the part number of the product, see type label.

{S/N} corresponds to the serial number of the product, see type label (if indicated).

Support Portal

NOTE

In the SICK Support Portal (**support.sick.com**) you will find, besides useful service and support information for your product, further detailed information on the available accessories and their use.

12 Annex

12.1 Declarations of conformity and certificates

You can download declarations of conformity and certificates via the product page.

The product page can be accessed via the **SICK Product ID**: **pid.sick.com/{P/N}/{S/N}**

{P/N} corresponds to the part number of the product, see type label.

{S/N} corresponds to the serial number of the product, see type label (if indicated).

12.2 Telegram listing (EN)

Contents

12.2.1.1 About this document ... 90
12.2.1.1.1 Information on the telegram listing ... 90
12.2.1.1.2 Explanation of symbols ... 91
12.2.1.2 Communication format ... 92
12.2.1.2.1 Binary telegram (CoLa B) ... 92
12.2.1.2.2 ASCII telegram (CoLa A) ... 92
12.2.1.2.3 Variable types ... 93
12.2.1.2.4 Command basics ... 94
12.2.1.2.5 Log in: Required user level ... 94
12.2.1.3 Workflows ... 95
12.2.1.3.1 Parameterize the scan ... 95
12.2.1.3.2 Set timestamp/data angle ... 95
12.2.1.3.3 Common telegrams ... 95
12.2.1.4 Telegrams ... 96
12.2.1.4.1 Log in [sMN SetAccessMode] ... 97
12.2.1.4.2 Basic Settings ... 98
12.2.1.4.2.1 Set frequency and angular resolution/measurement sectors [sMN mLMPsetscancfg] ... 98
12.2.1.4.2.2 Read for frequency and angular resolution [sRN LMPscancfg] ... 100
12.2.1.4.2.3 Activate standby mode [sMN LMCstandby] ... 101
12.2.1.4.2.4 Start measurement [sMN LMCstartmeas] ... 102
12.2.1.4.2.5 Stop measurement [sMN LMCstopmeas] ... 103
12.2.1.4.2.6 Autostart measurement [sWN LMPautostartmeas] ... 103
12.2.1.4.2.7 Load factory defaults [sMN mSCloadfacdef] ... 104
12.2.1.4.2.8 Load application defaults [sMN mSCloadappdef] ... 105
12.2.1.4.2.9 Change password [sMN SetPassword] ... 106
12.2.1.4.2.10 Check password [sMN CheckPassword] ... 107
12.2.1.4.2.11 Set contamination measurement settings [sWN LCMcfg] ... 108
12.2.1.4.2.12 Read contamination measurement settings [sRN LCMcfg] ... 109
12.2.1.4.2.13 Read contamination measurement detailed values [sRN CMContLvlM] ... 110
12.2.1.4.2.14 Read contamination status of the device [sRN LCMstate] ... 111
12.2.1.4.2.15 Save parameters permanently [sMN mEEwriteall] ... 112

12.2.1.4.2.16 Set to run [sMN Run] 112
12.2.1.4.2.17 Reboot device [sMN mSCreboot] 113
12.2.1.4.3 Measurement output telegram 114
12.2.1.4.3.1 Configure the data content for the scan [sWN LMDscandatacfg] 114
12.2.1.4.3.2 Configure aperture angle of the scandata for output [sWN LMPoutputRange] 116
12.2.1.4.3.3 Read for actual output range [sRN LMPoutputRange] 117
12.2.1.4.3.4 Poll one telegram [sRN LMDscandata] 118
12.2.1.4.3.5 Send data permanently [sEN LMDscandata] 119
12.2.1.4.4 Time stamp 127
12.2.1.4.4.1 Set time synchronization [sWN TSCRole] 127
12.2.1.4.4.2 Set time stamp [sMN LSPsetdatetime] 127
12.2.1.4.4.3 Read time stamp and status of the measurement function [sRN STIms] 129
12.2.1.4.4.4 Set NTP (Network Time Protocol) parameters 130
12.2.1.4.5 Filters 136
12.2.1.4.5.1 Set particle filter [sWN LFPparticle] 136
12.2.1.4.5.2 Set average filter [sWN LFPmeanfilter] 137
12.2.1.4.5.3 Set n-pulse to 1-pulse filter (Echo filter) [sWN LFPnto1filter] 138
12.2.1.4.5.4 Set fog filter [sWN MSsuppmode] 139
12.2.1.4.6 Encoder 139
12.2.1.4.6.1 Set increment source [sWN LICsrc] 139
12.2.1.4.6.2 Set encoder settings [sWN LICencset] 140
12.2.1.4.6.3 Set encoder resolution [sWN LICencres] 141
12.2.1.4.6.4 Set fixed speed [sWN LICFixVel] 142
12.2.1.4.6.5 Read speed threshold [sRN LICSpTh] 142
12.2.1.4.6.6 Read encoder speed [sRN LICencsp] 143
12.2.1.4.7 Inputs and Outputs 144
12.2.1.4.7.1 Read state of the ports [sRN LIDportstate] 144
12.2.1.4.7.2 Read state of the inputs [sRN LIDinputstate] 146
12.2.1.4.7.3 Read state of the outputs [sRN LIDoutputstate] 147
12.2.1.4.7.4 Receive outputstate by event [sEN LIDoutputstate] 148
12.2.1.4.7.5 Set output state [sMN mDOSetOutput] 149
12.2.1.4.7.6 Set debouncing time for input x [sWN DI3DebTim] 150
12.2.1.4.7.7 Reset output counter [sMN LIDrstoutpcnt] 151
12.2.1.4.8 Status 152
12.2.1.4.8.1 Read firmware version [sRN DeviceIdent] 152
12.2.1.4.8.2 Read the device state [sRN SCdevicestate] 153
12.2.1.4.8.3 Read device order number [sRN DIornr] 154
12.2.1.4.8.4 Read device type [sRN DItype] 154
12.2.1.4.8.5 Read operating hours [sRN ODoprh] 155
12.2.1.4.8.6 Read power on counter [sRN ODpwrc] 156
12.2.1.4.8.7 Read temperature [sRN OPcurtmpdev] 157
12.2.1.4.8.8 Set device name [sWN LocationName] 158

12.2.1.4.8.9 Read device name [sRN LocationName] 159
12.2.1.4.8.10 Read heating state [sRN OPheatstateext] 159
12.2.1.4.9 Interfaces 160
12.2.1.4.9.1 Set IP address [sWN EIIpAddr] 160
12.2.1.4.9.2 Read IP address [sRN EIIpAddr] 161
12.2.1.4.9.3 Set Ethernet gateway [sWN Elgate] 162
12.2.1.4.9.4 Read Ethernet gateway [sRN Elgate] 163
12.2.1.4.9.5 Set IP mask [sWN EImask] 164
12.2.1.4.9.6 Read IP mask [sRN EImask] 165
12.2.1.4.9.7 Read MAC address [sRN EIMacAdr] 166
12.2.1.4.9.8 Set baud rate for host interface [sWN SIHstBaud] 167
12.2.1.4.9.9 Read baud rate of host interface [sRN SIHstBaud] 167
12.2.1.4.9.10 Set Host port number [sWN EIHstPort] 168
12.2.1.4.9.11 Set Host port Command Language (CoLa dialect) [sWN EIHstCola] 169
12.2.1.4.9.12 Enable/Disable Front Panel [sWN LMLfpen] 170
12.2.1.4.9.13 Set function front panel [sWN LMLfpFcn] 172
12.2.1.4.9.14 Set front LEDs [sMN mLMLSetLed] 173
12.2.1.4.9.15 Set 7-segment display to specific symbol or number [sMN mLMLSetDisp] 174
12.2.1.4.10 Application 176
12.2.1.4.10.1 Request status change of monitoring fields on event [sEN ECRChangeArr] 176
12.2.1.4.10.2 Individual request of monitoring fields to their status changes – ECR xy [sRN ECRxy] 178
12.2.1.4.10.3 Request SOPAS field data structure [sMN mLFEgetField] 182
12.2.1.4.10.4 Request minimal and maximal perpendicular distance once [sRN LFEperpdistresult] 185
12.2.1.4.10.5 Request minimal and maximal perpendicular distance continiously on event [sEN LFEperpdistresult] 188
12.2.1.4.10.6 Request latest field infringement info [sRN LFEinfringe-mentinfo] 192
12.2.1.4.10.7 Request field infringement info continiously on event [sEN LFEinfringementinfo] 194
12.2.1.5 Diagnostics 197
12.2.1.5.1 SOPAS error codes 197

12.2.1.1 About this document

12.2.1.1.1 Information on the telegram listing

NOTE

In case you prefer to use complete drivers instead of single telegrams, the following options are availabe:

C++ drivers: **https://github.com/SICKAG/sick_scan_xd**

ROS drivers: **https://github.com/SICKAG/sick_scan_xd**

ROS2 drivers: **https://github.com/SICKAG/sick_scan_xd**

NOTE
Telegrams that are not described in this document for the device should not be implemented as they may either be incompetible or cause undesired effects.

NOTE
CoLa 2 is a SICK specific communication protocol which is used for communication between SICK devices and SICK specific tools and services only.

Please read this chapter carefully before beginning to use the telegram listing.

The telegram listing shows how to send telegrams via a terminal program using the SICK protocol CoLa A (ASCII and hexadecimal values, with TCP port 2111 or 2112) or CoLa B (binary/hexadecimal values, with TCP port 2112 only) to the device . This comprises the query of the current device state or certain parameter values, how to modify parameter values and the way in which the device confirms or responds to commands/telegrams.

The devices generally support automatic IP address discovery.

Default IP address is:
- 192.168.0.1

Subnet mask is 255.255.255.0.

IP ports:
- 2111: CoLa A (fixed)
- 2112: CoLa A (can be switched to CoLa B)
- 2213: UDP

The telegram listing does not or only in a few exceptional cases differentiate between individual device versions or sub product families such as LMS11x and LMS14x. Most parameter changes also require certain user levels. Additionally, commands may change during the product lifecycle and development process with a new firmware.

This document is based on the following firmware version (or newer):
- V2.00.3 (for LMS11x/15x/16x)
- V2.16 (for LMS12x/13x/14x)

If commands do not seem to work, please verify that your device version supports this functionality, that the minimum required user level has been selected and check on updates of this documentation.

12.2.1.1.2 Explanation of symbols

NOTE
... highlights useful tips and recommendations as well as information for efficient and trouble-free operation.



Telegram to device



Telegram from device



Unformatted example to copy and paste

12.2.1.2 Communication format

12.2.1.2.1 Binary telegram (CoLa B)

The binary telegram is a basic protocol of the scanner (CoLa B). All values are in hexadecimal code and grouped into pairs of two digits (= 1 byte). The string consists of four parts: header, data length, data and checksum (CS). It is highly recommended to use this protocol especially for measurement data, as the transmitted data amount is only about half as much as with CoLa A).

The header indicates with 4 × STX (02 02 02 02) the start of the telegram.

The data length defines the size of the data part (command part) by indicating the number of digit pairs in the third part. The size of the data length itself is 4 bytes, which means that the data part might have a maximum of 16^8 = 4,294,967,295 digit pairs (bytes).

The data part comprises the actual command with letters and characters converted to Hex (according to the ASCII chart) and the parameters of either decimal numbers converted to Hex or fixed Hex values with a specific, intrinsic meaning (no conversion). There is always a space (20) between the command and the parameters, but not between the different parameter values.

The checksum finally serves to verify that the telegram has been transferred correctly. The length of the checksum is 1 byte, CRC8. It is calculated with XOR.

Table 26: Example: Binary telegram

02 02 02 02	00 00 00 17	73 4D 4E 20 53 65 74 41 63 63 65 73 73 4D 6F 64 65 20 03 F4 72 47 44	B3
Header	Length	Data	CS

This is an example telegram for setting the user level “Authorized Client”:

- Header = 02 02 02 02
- Length = 23 bytes (17h)
- Data:
  - 73 4D 4E 20 = sMN = start of Sopas command (and space)
  - 53 65 74 41 63 63 65 73 73 4D 6F 64 65 20 = Set Access Mode = the actual command for setting the user level (and space)
  - 03 = fixed Hex value meaning user level “Authorized Client”
  - F4 72 47 44 = fixed Hex value, serving as password for the selected user level “Authorized Client”
- Checksum = B3 from XOR calculation

12.2.1.2.2 ASCII telegram (CoLa A)

The ASCII telegram is an alternative to the binary telegram, suitable especially to parametrize the sensor. However, due to the variable string length of ASCII telegrams, the Binary telegram is still recommended when using scanners with a PLC.

The ASCII telegram has the advantage that commands can be written in plaintext. The string consists only of two parts: the framing and the data part.

The framing indicates with <STX> and <ETX> the start and stop of each telegram.

The data part comprises the actual command with letters and characters (plaintext), parameter values either in decimal (special indicator required) or in hexadecimal (example: a frequency of 25 Hz = +2500 (decimal) = 09C4 (Hex)) and fixed hexadecimal values with a specific, intrinsic meaning.

NOTE
Leading zeros are deleted in ASCII. Therefore a space is always required between all command parts and parameter parts.

As further alternative within CoLa A, depending on the preferences of the user, all values can be written directly in Hex. This means however a 1:1 conversion of all letters and characters including numbers and fixed hexadecimal values via the ASCII chart.

NOTE
The device will confirm parameter values always in hexadecimal code, regardless of the code sent.

Table 27: Example: ASCII telegram

ASCII	<STX>	sMN{SPC}SetAccessMode{SPC}03{SPC}F4724744	<ETX>
Hex	02	73 4D 4E 20 53 65 74 41 63 63 65 73 73 4D 6F 64 65 20 30 33 20 46 34 37 32 34 37 34 34	03
	Start	Data	Stop

This is again an example telegram for setting the user level “Authorized Client”. As only fixed hexadecimal parameter values are needed, the option to use parameter values in decimal code with special indicator cannot be applied here:

- Framing = <STX> = telegram start = 02 (Hex)
- Data:
  - sMN = start of Sopas command (and blank) = 73 4D 4E 20 (Hex)
  - SetAccessMode = the actual command for setting the user level (and blank) = 53 65 74 41 63 63 65 73 73 4D 6F 64 65 20 (Hex)
  - 03 = fixed Hex value meaning user level “Authorized Client” (and blank) = 30 33 20 (Hex)
  - F4 72 47 44 = fixed Hex value, serving as password for the selected user level “Authorized Client” = 46 34 37 32 34 37 34 34 (Hex)
- Framing = <ETX> = telegram stop = 03 (Hex)

12.2.1.2.3 Variable types

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 ... 65,535	No
Int_16	2	-32,768 ... +32,767	Yes
Uint_32	4	0 ... 4,294,967,295	No
Int_32	4	-2,147,483,648 ... +2,147,483,647	Yes
ULInt_64	8	0 ... 18446744073709551616	No
Enum_8	1	Certain values defined in a list of Choices (0 ... 255)	No
Enum_16	2	Certain values defined in a list of Choices (0 ... 65535)	No
String	Context-dependent	Strings are not terminated in zeroes	

Variable type	Length (byte)	Value range	Sign
FlexString	array of visible characters with preceeding current length (UInt lenght) (array of 8 bit)	See description of String and FlexArray	
Real	4	Float according to IEEE754 (see www.h-schmidt.net/FloatConverter/IEEE754de.html)	Yes

Data length is always given in Bytes!

Struct	A structure is a sequence of further types. These types can be of a BasicType, Structs again or an Array.
Array	An Array is a repetition of a type. The length of the array is defined with each Array. The types can be of a BasicType, a Struct or an Array again (n-dimensional).
Flex Array	A FlexArray is a repetition of a type with a variable length. The maximum length of the array is defined with each FlexArray. The current length of the FlexArray is transferred as a UInt preceeding the Array itself. The types can be of a BasicType, a Struct or an Array again (n- dimensional).

12.2.1.2.4 Command basics

SOPAS communication is a index based communication and can be identified with telegram beginning of: sRI, sWI, sMI, sAI, sEI, sSI. Since the parallel usage of one port might be confusing, the usage of separate ports is adviced.

Every response telegram starts with a separat framed string:

<STX>sSI 2 1<ETX><STX>“Answer”<ETX>

Description	Value ASCII	Value Hex	Value Binary
Start of text	<STX>	02	02 02 02 02 + given length
End of text	<ETX>	03	Calculated checksum
Read	sRN	73 52 4E	
Write	sWN	73 57 4E	
Method	sMN	73 4D 4E	
Event	sEN	73 45 4E	
Answer	sRA sWA sAN sEA sSN	73 52 41 73 57 41 73 41 4E 73 45 41 73 53 4E	
Space	{SPC}	20	20

If values are divided into two parts (e.g. measurement data), they are documented according to LSB 0 (e.g. 00 07), output however is according to MSB (e.g. 07 00).

NOTE

Every write command (sWN) has a read (sRN) counter part even if it may not be described in the Telegram Listing. You can verify the current parameters this way, e.g. read the current IP address via `sRN EIIpAddr` see "Read IP address [sRN EIIpAddr]", page 161.

12.2.1.2.5 Log in: Required user level

Task	Required user level
Change sensor parameters	Authorized Client

Task	Required user level
Requests or queries (e.g. for measurement data or device state)	None
Manage passwords	Service

In general, every sWN command for changing paramters requires to log in to the device first see "Log in [sMN SetAccessMode] ", page 97. When being logged in, any desired parameter valid for this user level can be changed. All changes become active only after having logged off again from the device via the sMN Run command see "Set to run [sMN Run]", page 112.

In this document, a required, specific user level is indicated in the telegram structure head line.

12.2.1.3 Workflows

12.2.1.3.1 Parameterize the scan

Log in: `sMN SetAccessMode` see "Log in [sMN SetAccessMode] ", page 97

Set frequency and resolution: `sMN mLMPsetscancfg` see "Set frequency and angular resolution/measurement sectors [sMN mLMPsetscancfg]", page 98

Configure scandata content: `sWN LMDscandatacfg` see "Configure the data content for the scan [sWN LMDscandatacfg]", page 114

Configure scandata output: `sWN LMPoutputRange` see "Configure aperture angle of the scandata for output [sWN LMPoutputRange]", page 116

Store parameters: `sMN mEEwriteall` see "Save parameters permanently [sMN mEEwriteall]", page 112

Log out: `sMN Run` see "Set to run [sMN Run]", page 112

Request scan:

`sRN LMDscandata` see "Poll one telegram [sRN LMDscandata]", page 118

`sEN LMDscandata` see "Send data permanently [sEN LMDscandata]", page 119

(Device output ...)

12.2.1.3.2 Set timestamp/data angle

Log in: `sMN SetAccessMode` see "Log in [sMN SetAccessMode] ", page 97

SOPAS command: `sMN LSPsetdatetime` see "Set time stamp [sMN LSPsetdatetime]", page 127

Log out: `sMN Run` see "Set to run [sMN Run]", page 112

12.2.1.3.3 Common telegrams

The following telegrams are valid for a wide range of non-safe LiDAR sensors from SICK. Please refer to the telegram listing of the respective device for a detailed description of all valid telegrams.

"Log in [sMN SetAccessMode] ", page 97
"Set frequency and angular resolution/measurement sectors [sMN mLMPsetscancfg]", page 98
"Start measurement [sMN LMCstartmeas]", page 102
"Stop measurement [sMN LMCstopmeas]", page 103
"Load factory defaults [sMN mSCloadfacdef]", page 104
"Load application defaults [sMN mSCloadappdef]", page 105

"Change password [sMN SetPassword]", page 106
"Check password [sMN CheckPassword]", page 107
"Reboot device [sMN mSCreboot]", page 113
"Save parameters permanently [sMN mEEwriteall]", page 112
"Set to run [sMN Run]", page 112
"Configure the data content for the scan [sWN LMDscandatacfg]", page 114
"Configure aperture angle of the scandata for output [sWN LMPoutputRange]", page 116
"Read for actual output range [sRN LMPoutputRange]", page 117
"Poll one telegram [sRN LMDscandata]", page 118
"Send data permanently [sEN LMDscandata]", page 119
"Set particle filter [sWN LFPparticle]", page 136
"Read state of the inputs [sRN LIDinputstate]", page 146
"Read state of the outputs [sRN LIDoutputstate]", page 147
"Receive outputstate by event [sEN LIDoutputstate]", page 148
"Set output state [sMN mDOSetOutput]", page 149
"Read firmware version [sRN DeviceIdent]", page 152
"Read the device state [sRN SCdevicestate]", page 153
"Read device order number [sRN DIornr]", page 154
"Read device type [sRN DItype]", page 154
"Read operating hours [sRN ODoprh]", page 155
"Read power on counter [sRN ODpwrc]", page 156
"Set device name [sWN LocationName]", page 158
"Read device name [sRN LocationName]", page 159
"Reset output counter [sMN LIDrstoutpcnt]", page 151
"Set IP address [sWN EIIpAddr]", page 160
"Read IP address [sRN EIIpAddr]", page 161
"Set Ethernet gateway [sWN EIgate]", page 162
"Read Ethernet gateway [sRN EIgate]", page 163
"Set IP mask [sWN EImask]", page 164
"Read IP mask [sRN EImask]", page 165
"Set Host port number [sWN EIHstPort]", page 168
"Set Host port Command Language (CoLa dialect) [sWN EIHstCola]", page 169

12.2.1.4 Telegrams

Telegrams listed in this document are described in the following basic structure:

Table 28: Telegram structure: "Command type" "Command"

Telegram structure: "Command type" "Command" (Minimum required user level. If nothing is stated, no user level required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Lists the different parts of the telegram.	Describes the corresponding telegram parts.	Defines the type of the variable.	Defines the length in byte.	Gives further information regarding the values in CoLa A/ CoLa B if neccessary.	Defines the value of the telegram part in CoLa A (ASCII).	Defines the value of the telegram part in CoLa B (Binary).

NOTE
Commands are colored blue, parameters orange for further differentiation.

Table 29: Example: "Command type" "Command"

CoLa A	<"Start of text">"Command type value (ASCII)""space""Command value (ASCII)""space""Parameter value (ASCII)""space""Parameter value (ASCII)"<"End of text">	
	Copy example with framing (ASCII)	
	Copy example without framing (ASCII)	
	Copy example with framing (Hex)	
CoLa B	"Start of text and given length" "Command type value (Binary)""space""Command value (Binary)""space""Parameter value (Binary)""Parameter value (Binary)""Calculated checksum"	
	Copy example without framing (Binary)	

Table 30: Telegram structure: "Command type" "Command" (Answer)

Telegram structure: "Command type" "Command"						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Lists the different parts of the telegram.	Describes the corresponding telegram parts.	Defines the type of the variable.	Defines the length in byte.	Gives further information regarding the values in CoLa A/ CoLa B if neccessary.	Defines the value of the telegram part in CoLa A (ASCII).	Defines the value of the telegram part in CoLa B (Binary).

Table 31: Example: "Command type" "Command" (Answer)

CoLa A	<"Start of text">"Command type value (ASCII)""space""Command value (ASCII)""space""Parameter value (ASCII)""space""Parameter value (ASCII)"<"End of text">
	<"Start of text">"Command type value (Hex)""space""Command value (Hex)""space""Parameter value (Hex)""space""Parameter value (Hex)"<"End of text">
CoLa B	"Start of text and given length""Command type value (Binary)""space""Command value (Binary)""space""Parameter value (Binary)""Parameter value (Binary)""Calculated checksum"

12.2.1.4.1 Log in [sMN SetAccessMode]

NOTE

- An automated hash-value calculator can be found in SOPAS ET under menu “password”. Required userlevel 'Service' (see "Change password [sMN SetPassword]", page 106).

A log in to the device is necessary to change parameters. In most cases, the user level 'Authorized client' is needed. Changed parameters will be reset to the previous state via a reboot unless the are saved. To save parameter changes the command "sMN mEEwriteall" (see "Save parameters permanently [sMN mEEwriteall]", page 112) must be send before log out.

Table 32: Telegram structure: sMN SetAccessMode

Telegram structure: sMN SetAccessMode						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3		sMN	73 4D 4E
Command	User level	String	13		SetAccess-Mode	53 65 74 41 63 63 65 73 73 4D 6F 64 65
User level	Select user level	Int_8	1	Maintenance: Authorized client: Service:	2 3 4	02 03 04
Password	Hash value for the selected user level	Uint_32	4	Maintenance: Authorized client: Service:	B21ACE26 F4724744 81BE23AA	B2 1A CE 26 F4 72 47 44 81 BE 23 AA

Table 33: Example: sMN SetAccessMode - Log in as “Authorized client” with password “F4724744”

CoLa A	<STX>sMN{SPC}SetAccessMode{SPC}3{SPC}F4724744<ETX>
	<STX>sMN SetAccessMode 3 F4724744<ETX>
	sMN SetAccessMode 3 F4724744
	02 73 4D 4E 20 53 65 74 41 63 63 65 73 73 4D 6F 64 65 20 30 33 20 46 34 37 32 34 37 34 34 03
CoLa B	02 02 02 02 00 00 00 17 73 4D 4E 20 53 65 74 41 63 63 65 73 73 4D 6F 64 65 20 03 F4 72 47 44 B3
	73 4D 4E 20 53 65 74 41 63 63 65 73 73 4D 6F 64 65 20 03 F4 72 47 44

Table 34: Telegram structure: sAN SetAccessMode

Telegram structure: sAN SetAccessMode						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sAN	73 41 4E
Command	User level	String	13		SetAccess-Mode	53 65 74 41 63 63 65 73 73 4D 6F 64 65
Change user level	Changed level	Bool_1	1	Error: Success:	0 1	00 01

Table 35: Example: sAN SetAccessMode

CoLa A	<STX>sAN{SPC}SetAccessMode{SPC}1<ETX>
	02 73 41 4E 20 53 65 74 41 63 63 65 73 73 4D 6F 64 65 20 31 03
CoLa B	02 02 02 02 00 00 00 13 73 41 4E 20 53 65 74 41 63 63 65 73 73 4D 6F 64 65 20 01 38

12.2.1.4.2 Basic Settings

12.2.1.4.2.1 Set frequency and angular resolution/measurement sectors [sMN mLMPsetscancfg]

NOTE
Please note that the new values will be activated only after log out (from the user level), when re-entering the Run mode (see "Set to run [sMN Run]", page 112).

NOTE
Start angle and stop angle are fixed values and not changeable only in the data output!

Coordination system:

(-45° to 225°), front = 90°

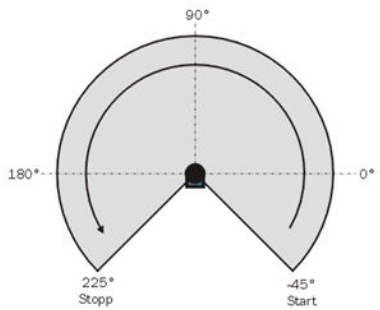


Table 36: Telegram structure: sMN mLMPsetscancfg

Telegram structure: sMN mLMPsetscancfg (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3		sMN	73 4D 4E
Command	Configuration of scan frequency and angular resolution	String	14		mLMPsets-cancfg	6D 4C 4D 50 73 65 74 73 63 61 6E 63 66 67
Scan frequency	[1/100 Hz]	Uint_32	4	25 Hz: 50 Hz:	+2500d (9C4h) +5000d (1388h)	00 00 09 C4 00 00 13 88
Reserved	-	Int_16	2	Always:	+1 (0001h)	00 01
Angular resolution	[1/10000°]	Uint_32	4	0.25°: 0.5°:	+2500d (9C4h) +5000d (1388h)	00 00 09 C4 00 00 13 88
Start angle	[1/10000°] (Values are fixed)	Int_32	4	Fixed: -45°	-450000d (FFF92230h)	FF F9 22 30
Stop angle	[1/10000°] (Values are fixed)	Int_32	4	Fixed: +225°	+2250000d (225510h)	00 22 55 10

Example

NOTE
Scan angle can not be changed here, only in the data output!

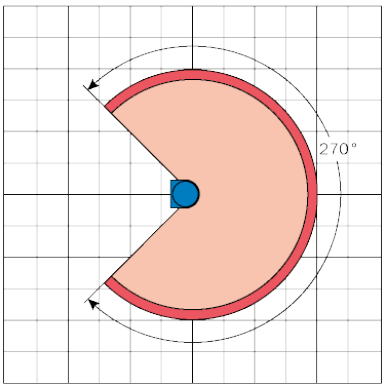


Figure 58: Example - 1 measurement sector of 270°

Table 37: Example: sMN mLMPsetscancfg - 50 Hz scan frequency, 0.5° angular resolution, -45° start angle (fixed), +225° stop angle (fixed)

CoLa A	<STX>sMN{SPC}mLMPsetscancfg{SPC}+5000{SPC}+1{SPC}+5000{SPC}-450000{SPC}+2250000<ETX> Alternatively: <STX>sMN{SPC}mLMPsetscancfg{SPC}1388{SPC}1{SPC}1388{SPC}FFF92230{SPC}225510<ETX>	
	<STX>sMN mLMPsetscancfg +5000 +1 +5000 -450000 +2250000<ETX> Alternatively: <STX>sMN mLMPsetscancfg 1388 1 1388 FFF92230 225510<ETX>	
	sMN mLMPsetscancfg +5000 +1 +5000 -450000 +2250000 Alternatively: sMN mLMPsetscancfg 1388 1 1388 FFF92230 225510	
	02 73 4D 4E 20 6D 4C 4D 50 73 65 74 73 63 61 6E 63 66 67 20 2B 35 30 30 30 20 2B 31 20 2B 35 30 30 30 20 2D 34 35 30 30 30 30 20 2B 32 32 35 30 30 30 30 03 Alternatively: 02 73 4D 4E 20 6D 4C 4D 50 73 65 74 73 63 61 6E 63 66 67 20 31 33 38 38 20 31 20 31 33 38 38 20 46 46 46 39 32 32 33 30 20 32 32 35 35 31 30 03	

CoLa B	02 02 02 02 00 00 00 25 73 4D 4E 20 6D 4C 4D 50 73 65 74 73 63 61 6E 63 66 67 20 00 00 13 88 00 01 00 00 13 88 FF F9 22 30 00 22 55 10 21 73 4D 4E 20 6D 4C 4D 50 73 65 74 73 63 61 6E 63 66 67 20 00 00 13 88 00 01 00 00 13 88 FF F9 22 30 00 22 55 10

Table 38: Telegram structure: sAN mLMPsetscancfg

Telegram structure: sAN mLMPsetscancfg						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sAN	73 41 4E
Command	Info of scan frequency and angular resolution	String	14		mLMPsets-cancfg	6D 4C 4D 50 73 65 74 73 63 61 6E 63 66 67
Status code	Accepted when value is 0	Enum_8	1	No error: Frequency error: Resolution error: Resolution and scan area/ frequency error: Scan area error: Other errors:	0 1 2 3 4 5	00 01 02 03 04 05
Scan frequency	[1/100 Hz]	Uint_32	4	25 Hz: 50 Hz:	9C4h 1388h	00 00 09 C4 00 00 13 88
Reserved	-	Int_16	2	Always:	1h	00 01
Angular resolution	[1/10000°]	Uint_32	4	0.25°: 0.5°:	9C4h 1388h	00 00 09 C4 00 00 13 88
Start angle	[1/10000°]	Int_32	4	Fixed: -45°	FFF92230h	FF F9 22 30
Stop angle	[1/10000°]	Int_32	4	Fixed: +225°	225510h	00 22 55 10

Table 39: Example: sAN mLMPsetscancfg

CoLa A	<STX>sAN{SPC}mLMPsetscancfg{SPC}0{SPC}1388{SPC}1{SPC}1388{SPC}FFF92230{SPC}225510<ETX>
	02 73 41 4E 20 6D 4C 4D 50 73 65 74 73 63 61 6E 63 66 67 20 30 20 31 33 38 38 20 31 20 31 33 38 38 20 46 46 46 39 32 32 33 30 20 32 32 35 35 31 30 03
CoLa B	02 02 02 02 00 00 00 26 73 41 4E 20 6D 4C 4D 50 73 65 74 73 63 61 6E 63 66 67 20 00 00 00 13 88 00 01 00 00 13 88 FF F9 22 30 00 22 55 10 2D

12.2.1.4.2.2 Read for frequency and angular resolution [sRN LMPscancfg]

Read the scanning frequency, angular resolution and aperture angle of the device. Values show the information of the values before any filters are applied. Therefore aperature angle is always showing the maximum, scanning frequency and angular resolution may also be static depending on device family.

Table 40: Telegram structure: sRN LMPscancfg

Telegram structure: sRN LMPscancfg						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E
Command	Info of scan frequency and angular resolution	String	10		LMPscancfg	4C 4D 50 73 63 61 6E 63 66 67

Table 41: Example: sRN LMPscancfg

CoLa A	<STX>sRN{SPC}LMPscancfg<ETX>	
	<STX>sRN LMPscancfg<ETX>	
	sRN LMPscancfg	
	02 73 52 4E 20 4C 4D 50 73 63 61 6E 63 66 67 03	
CoLa B	02 02 02 02 00 00 00 0E 73 52 4E 20 4C 4D 50 73 63 61 6E 63 66 67 63	
	73 52 4E 20 4C 4D 50 73 63 61 6E 63 66 67	

Table 42: Telegram structure: sRA LMPscancfg

Telegram structure: sRA LMPscancfg						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sRA	73 52 41
Command	Info of scan frequency and angular resolution	String	10		LMPscancfg	4C 4D 50 73 63 61 6E 63 66 67
Scan frequency	[1/100 Hz]	Uint_32	4	25 Hz: 50 Hz:	9C4h 1388h	00 00 09 C4 00 00 13 88
Reserved	-	Int_16	2	Always:	1	00 01
Angular resolution	[1/10000°]	Uint_32	4	0.25°: 0.5°:	9C4h 1388h	00 00 09 C4 00 00 13 88
Start angle	[1/10000°]	Int_32	4	Fixed: -45°	FFF92230h	FF F9 22 30
Stop angle	[1/10000°]	Int_32	4	Fixed: +225°	225510h	00 22 55 10

Table 43: Example: sRA LMPscancfg

CoLa A	<STX>sRA{SPC}LMPscancfg{SPC}1388{SPC}1{SPC}1388{SPC}FFF92230{SPC}225510<ETX>
	02 73 52 41 20 4C 4D 50 73 63 61 6E 63 66 67 20 31 33 38 38 20 31 20 31 33 38 38 20 46 46 46 39 32 32 33 30 20 32 32 35 35 31 30 03
CoLa B	02 02 02 02 00 00 00 21 73 52 41 20 4C 4D 50 73 63 61 6E 63 66 67 20 00 00 13 88 00 01 00 00 13 88 FF F9 22 30 00 22 55 10 3E

12.2.1.4.2.3 Activate standby mode [sMN LMCstandby]

Shut off the laser in order to extend the lifetime of laser diode. The motor keeps on turning.

Table 44: Telegram structure: sMN LMCstandby

Telegram structure: sMN LMCstandby (User level 'Authorized Client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3		sMN	73 4D 4E
Command	Set device to standby	String	10		LMCstandby	4C 4D 43 73 74 61 6E 64 62 79

Table 45: Example: sMN LMCstandby

CoLa A	<STX>sMN{SPC}LMCstandby<ETX>	
	<STX>sMN LMCstandby<ETX>	
	sMN LMCstandby	
	02 73 4D 4E 20 4C 4D 43 73 74 61 6E 64 62 79 03	
CoLa B	02 02 02 02 00 00 00 0E 73 4D 4E 20 4C 4D 43 73 74 61 6E 64 62 79 65	
	73 4D 4E 20 4C 4D 43 73 74 61 6E 64 62 79	

Table 46: Telegram structure: sAN LMCstandby

Telegram structure: sAN LMCstandby						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sAN	73 41 4E
Command	Set device to standby	String	10		LMCstandby	4C 4D 43 73 74 61 6E 64 62 79
Status code	Accepted when value is 0	Enum_8	1	Success: Not allowed:	0 1	00 01

Table 47: Example: sAN LMCstandby

CoLa A	<STX>sAN{SPC}LMCstandby{SPC}0<ETX>
	02 73 41 4E 20 4C 4D 43 73 74 61 6E 64 62 79 20 30 03
CoLa B	02 02 02 02 00 00 00 10 73 41 4E 20 4C 4D 43 73 74 61 6E 64 62 79 20 00 49

12.2.1.4.2.4 Start measurement [sMN LMCstartmeas]

Start the laser and (unless in Standby mode) the motor of the device

Table 48: Telegram structure: sMN LMCstartmeas

Telegram structure: sMN LMCstartmeas (User level 'Authorized Client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3		sMN	73 4D 4E
Command	Start measurement	String	12		LMCstartmeas	4C 4D 43 73 74 61 72 74 6D 65 61 73

Table 49: Example: sMN LMCstartmeas

CoLa A	<STX>sMN{SPC}LMCstartmeas<ETX>	
	<STX>sMN LMCstartmeas<ETX>	
	sMN LMCstartmeas	
	02 73 4D 4E 20 4C 4D 43 73 74 61 72 74 6D 65 61 73 03	
CoLa B	02 02 02 02 00 00 00 10 73 4D 4E 20 4C 4D 43 73 74 61 72 74 6D 65 61 73 68	
	73 4D 4E 20 4C 4D 43 73 74 61 72 74 6D 65 61 73	

Table 50: Telegram structure: sAN LMCstartmeas

Telegram structure: sAN LMCstartmeas						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sAN	73 41 4E
Command	Start measurement	String	12		LMCstartmeas	4C 4D 43 73 74 61 72 74 6D 65 61 73
Status code	Accepted when value is 0	Enum_8	1	Success: Not allowed:	0 1	00 01

Table 51: Example: sAN LMCstartmeas

CoLa A	<STX>sAN{SPC}LMCstartmeas{SPC}0<ETX>
	02 73 41 4E 20 4C 4D 43 73 74 61 72 74 6D 65 61 73 20 30 03
CoLa B	02 02 02 02 00 00 00 12 73 41 4E 20 4C 4D 43 73 74 61 72 74 6D 65 61 73 20 00 44

12.2.1.4.2.5 Stop measurement [sMN LMCstopmeas]

Shut off the laser and stop the motor.

Table 52: Telegram structure: sMN LMCstopmeas

Telegram structure: sMN LMCstopmeas (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3		sMN	73 4D 4E
Command	Stop measurement	String	11		LMCstopmeas	4C 4D 43 73 74 6F 70 6D 65 61 73

Table 53: Example: sMN LMCstopmeas

CoLa A	<STX>sMN{SPC}LMCstopmeas<ETX>	
	<STX>sMN LMCstopmeas<ETX>	
	sMN LMCstopmeas	
	02 73 4D 4E 20 4C 4D 43 73 74 6F 70 6D 65 61 73 03	
CoLa B	02 02 02 02 00 00 00 0F 73 4D 4E 20 4C 4D 43 73 74 6F 70 6D 65 61 73 10	
	73 4D 4E 20 4C 4D 43 73 74 6F 70 6D 65 61 73	

Table 54: Telegram structure: sAN LMCstopmeas

Telegram structure: sAN LMCstopmeas						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sAN	73 41 4E
Command	Stop measurement	String	11		LMCstopmeas	4C 4D 43 73 74 6F 70 6D 65 61 73
Status code	Accepted when value is 0	Enum_8	1	No error: Not allowed:	0 1	00 01

Table 55: Example: sAN LMCstopmeas

CoLa A	<STX>sAN{SPC}LMCstopmeas{SPC}0<ETX>
	02 73 41 4E 20 4C 4D 43 73 74 6F 70 6D 65 61 73 20 30 03
CoLa B	02 02 02 02 00 00 00 11 73 41 4E 20 4C 4D 43 73 74 6F 70 6D 65 61 73 20 00 3C

12.2.1.4.2.6 Autostart measurement [sWN LMPautostartmeas]

This parameter defines whether the scanner will start to rotate directly and measure when powering up or remain in idle mode. The changed setting (saved with the command `sMN mEEWriteall`, see "Save parameters permanently [sMN mEEwriteall]", page 112) will be then be active with the next power-up cycle.

In case autostart is deactivated, the sensor will remain in idle mode with the laser and motor switched off.

Table 56: Telegram structure: sWN LMPautostartmeas

Telegram structure: sWN LMPautostartmeas (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3		sWN	73 57 4E
Command	Autostart measurement	String	16		LMPautostart-meas	4C 4D 50 61 75 74 6F 73 74 61 72 74 6D 65 61 73
Status code	Activate / Deactivate Autostart	Bool_1	1	Autostart off: Autostart on:	0 1	00 01

Table 57: Example: sWN LMPautostartmeas 1

CoLa A	<STX>sWN{SPC}LMPautostartmeas{SPC}1<ETX>	
	<STX>sWN LMPautostartmeas 1<ETX>	
	sWN LMPautostartmeas 1	
	02 73 57 4E 20 4C 4D 50 61 75 74 6F 73 74 61 72 74 6D 65 61 73 20 31 03	
CoLa B	02 02 02 02 00 00 00 16 73 57 4E 20 4C 4D 50 61 75 74 6F 73 74 61 72 74 6D 65 61 73 20 01 4F	
	73 57 4E 20 4C 4D 50 61 75 74 6F 73 74 61 72 74 6D 65 61 73 20 01	

Table 58: Telegram structure: sWA LMDautostartmeas

Telegram structure: sWA LMPautostartmeas						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sWA	73 57 41
Command	Autostart measurement	String	16		LMPautostart-meas	4C 4D 50 61 75 74 6F 73 74 61 72 74 6D 65 61 73

Table 59: Example: sWA LMPautostartmeas

CoLa A	<STX>sWA{SPC}LMPautostartmeas<ETX>
	02 73 57 41 20 4C 4D 43 73 74 61 72 74 6D 65 61 73 03
CoLa B	02 02 02 02 00 00 00 15 73 57 41 20 4C 4D 50 61 75 74 6F 73 74 61 72 74 6D 65 61 73 20 41

12.2.1.4.2.7 Load factory defaults [sMN mSCloadfacdef]

NOTE

The Factory-Reset (Load factory defaults) deletes the entire parametrization of the device. All parameters, settings and system applications will be set to default. The Host port command language (CoLa dialect) will be set to CoLa A by default.

Table 60: Telegram structure: sMN mSCloadfacdef

Telegram structure: sMN mSCloadfacdef (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3		sMN	73 4D 4E

Telegram structure: sMN mSCloadfacdef (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command	Load factory defaults	String	13		mSCloadfacdef	6D 53 43 6C 6F 61 64 66 61 63 64 65 66

Table 61: Example: sMN mSCloadfacdef

CoLa A	<STX>sMN{SPC}mSCloadfacdef<ETX>
	<STX>sMN mSCloadfacdef<ETX>
	sMN mSCloadfacdef
	02 73 4D 4E 20 6D 53 43 6C 6F 61 64 66 61 63 64 65 66 03
CoLa B	02 02 02 02 00 00 00 11 73 4D 4E 20 6D 53 43 6C 6F 61 64 66 61 63 64 65 66 28
	73 4D 4E 20 6D 53 43 6C 6F 61 64 66 61 63 64 65 66

Table 62: Telegram structure: sAN mSCloadfacdef

Telegram structure: sAN mSCloadfacdef						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sAN	73 41 4E
Command	Load factory defaults	String	13		mSCloadfacdef	6D 53 43 6C 6F 61 64 66 61 63 64 65 66

Table 63: Example: sAN mSCloadfacdef

CoLa A	<STX>sAN{SPC}mSCloadfacdef<ETX>
	02 73 4D 4E 20 6D 53 43 6C 6F 61 64 66 61 63 64 65 66 03
CoLa B	02 02 02 02 00 00 00 12 73 41 4E 20 6D 53 43 6C 6F 61 64 66 61 63 64 65 66 20 04

12.2.1.4.2.8 Load application defaults [sMN mSCloadappdef]

NOTE

The Application-Reset (Load application defaults) deletes only the user parametrization of the Fields, Evaluation cases (EVC) and parameters under the header “Application”. Other parameters like Interface settings, Echo Filter, etc. remain unaffected.

Table 64: Telegram structure: sMN mSCloadappdef

Telegram structure: sMN mSCloadappdef (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3		sMN	73 4D 4E
Command	Load application defaults	String	13		mSCloadappdef	6D 53 43 6C 6F 61 64 61 70 70 64 65 66

Table 65: Example: sMN mSCloadappdef

CoLa A	<STX>sMN{SPC}mSCloadappdef<ETX>
	<STX>sMN mSCloadappdef<ETX>
	sMN mSCloadappdef
	02 73 4D 4E 20 6D 53 43 6C 6F 61 64 61 70 70 64 65 66 03
CoLa B	02 02 02 02 00 00 00 11 73 4D 4E 20 6D 53 43 6C 6F 61 64 61 70 70 64 65 66 2D
	73 4D 4E 20 6D 53 43 6C 6F 61 64 61 70 70 64 65 66

Table 66: Telegram structure: sAN mSCloadappdef

Telegram structure: sAN mSCloadappdef						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sAN	73 41 4E
Command	Load application defaults	String	13		mSCloadappdef	6D 53 43 6C 6F 61 64 61 70 70 64 65 66

Table 67: Example: sAN mSCloadappdef

CoLa A	<STX>sAN{SPC}mSCloadappdef<ETX>
	02 73 41 4E 20 6D 53 43 6C 6F 61 64 61 70 70 64 65 66 03
CoLa B	02 02 02 02 00 00 00 12 73 41 4E 20 6D 53 43 6C 6F 61 64 61 70 70 64 65 66 20 01

12.2.1.4.2.9 Change password [sMN SetPassword]

Changing the log in password for a specific user level.

NOTE
If logged in with a higher user level you may set the password for lower user levels as well.

Table 68: Telegram structure: sMN SetPassword

Telegram structure: sMN SetPassword (Same user level or higher required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3		sMN	73 4D 4E
Command	Set password request	String	13		SetPassword	53 65 74 50 61 73 73 77 6F 72 64
User level	User level that the password will be applied to	Int_8	1	Maintenance: Authorized client: Service:	2 3 4	02 03 04
Password	Hash value of the new password	Uint_32	4		<Hash value>	

Set password for 'Authorized client' to “testtest” (hash value = 1920E4C9).

Table 69: Example: sMN SetPassword

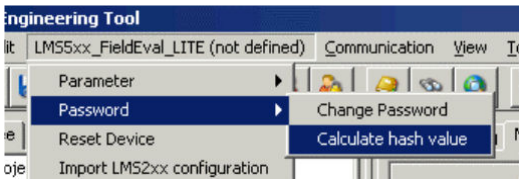
CoLa A	<STX>sMN{SPC}SetPassword{SPC}3{SPC}1920E4C9<ETX>
	<STX>sMN SetPassword 3 1920E4C9<ETX>
	sMN SetPassword 3 1920E4C9
	02 73 4D 4E 20 53 65 74 50 61 73 73 77 6F 72 64 20 33 20 31 39 32 30 45 34 43 39 03

	02 02 02 02 00 00 00 15 73 4D 4E 20 53 65 74 50 61 73 73 77 6F 72 64 20 03 19 20 E4 C9 1A
CoLa B	73 4D 4E 20 53 65 74 50 61 73 73 77 6F 72 64 20 03 19 20 E4 C9

Calculating the hash value of the password

→ Login in SOPAS ET with user level “Service” to the device.

→ Select **[Device]** > **Password** > **Calculate Hash value**.



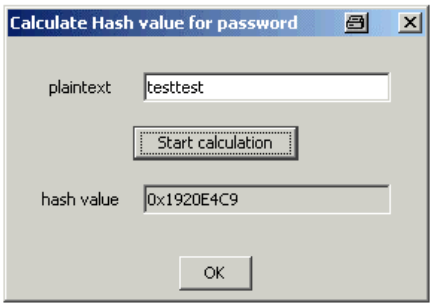


Table 70: Telegram structure: sAN SetPassword

Telegram structure: sAN SetPassword						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sAN	73 41 4E
Command	Set password requested	String	13		SetPassword	53 65 74 50 61 73 73 77 6F 72 64
Success	Confirmation	Int_8	1	Failed: Success:	0 1	00 01

Table 71: Example: sAN SetPassword

CoLa A	<STX>sAN{SPC}SetPassword{SPC}1<ETX>
	02 73 4D 4E 20 53 65 74 50 61 73 73 77 6F 72 64 20 31 03
CoLa B	02 02 02 02 00 00 00 11 73 41 4E 20 53 65 74 50 61 73 73 77 6F 72 64 20 01 00

12.2.1.4.2.10 Check password [sMN CheckPassword]

Check the password for a specific user level, e.g. to verify if it has been changed correctly.

Table 72: Telegram structure: sMN CheckPassword

Telegram structure: sMN CheckPassword (Same User level or higher required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3		sMN	73 4D 4E
Command	Check password request	String	13		CheckPassword	43 68 65 63 6B 50 61 73 73 77 6F 72 64
User level	User level to check the password for	Int_8	1	Maintenance: Authorized client: Service:	2 3 4	02 03 04
Password	Hash value of the password to be checked	Uint_32	4		<Hash value>	

Check password “testtest” for 'Authorized client'.

Table 73: Example: sMN CheckPassword

CoLa A	<STX>sMN{SPC}CheckPassword{SPC}3{SPC}1920E4C9<ETX>
	<STX>sMN CheckPassword 3 1920E4C9<ETX>
	sMN CheckPassword 3 1920E4C9
	02 73 4D 4E 20 43 68 65 63 6B 50 61 73 73 77 6F 72 64 20 33 20 31 39 32 30 45 34 43 39 03
CoLa B	02 02 02 02 00 00 00 17 73 4D 4E 20 43 68 65 63 6B 50 61 73 73 77 6F 72 64 20 03 19 20 E4 C9 1E
	73 4D 4E 20 43 68 65 63 6B 50 61 73 73 77 6F 72 64 20 03 19 20 E4 C9

Table 74: Telegram structure: sAN CheckPassword

Telegram structure: sAN CheckPassword						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sAN	73 41 4E
Command	Check password request	String	13		CheckPassword	43 68 65 63 6B 50 61 73 73 77 6F 72 64
Success	Confirmation	Int_8	1	Failed: Success:	0 1	00 01

Table 75: Example: sAN CheckPassword

CoLa A	<STX>sAN{SPC}CheckPassword{SPC}1<ETX>
	02 73 41 4E 20 43 68 65 63 6B 50 61 73 73 77 6F 72 64 20 31 03
CoLa B	02 02 02 02 00 00 00 13 73 41 4E 20 43 68 65 63 6B 50 61 73 73 77 6F 72 64 20 01 04

12.2.1.4.2.11 Set contamination measurement settings [sWN LCMcfg]

Define if and how early the device shall signal a potential contamination of the optics cover so that it may be cleaned preventively see "Contamination measurement", page 26.

Table 76: Telegram structure: sWN LCMcfg

Telegram structure: sWN LCMcfg (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3		sWN	73 57 4E
Command		String	6		LCMcfg	4C 43 4D 63 66 67
Strategy	Strategy code	Enum_8	1	Inactive: High available Available: Sensitive: Semi-sensitive:	0 1 2 3 4	00 01 02 03 04
Response time	Time lapse	Uint_16	2		+1d ... +60d (01h ... 3Ch)	00 01 ... 00 3C
Threshold warning	Threshold value	Uint_16	2		0d ... +100d (00h ... 64h)	00 00 ... 00 64
Threshold error	Threshold value	Uint_16	2		0d ... +100d (00h ... 64h)	00 00 ... 00 64

Table 77: Example: sWN LCMcfg

CoLa A	<STX>sWN{SPC}LCMcfg{SPC}1{SPC}+30{SPC}+65{SPC}+45<ETX>
	<STX>sWN LCMcfg 1 +30 +65 +45<ETX>
	sWN LCMcfg 1 +30 +65 +45
	02 73 57 4E 20 4C 43 4D 63 66 67 20 31 20 2B 33 30 20 2B 36 35 20 2B 34 35 03
CoLa B	02 02 02 02 00 00 00 18 73 57 4E 20 4C 43 4D 63 66 67 20 01 00 1E 00 41 00 2D 39
	73 57 4E 20 4C 43 4D 63 66 67 20 01 00 1E 00 41 00 2D

Table 78: Telegram structure: sWA LCMcfg

Telegram structure: sWA LCMcfg						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sWA	73 57 41
Command		String	6		LCMcfg	4C 43 4D 63 66 67

Table 79: Example: sWA LCMcfg

CoLa A	<STX>sWA{SPC}LCMcfg<ETX>
	02 73 57 41 20 4C 43 4D 63 66 67 03
CoLa B	02 02 02 02 00 00 00 0B 73 57 41 20 4C 43 4D 63 66 67 20 45

12.2.1.4.2.12 Read contamination measurement settings [sRN LCMcfg]

Read which contamination measurement strategy is currently active see "Contamination measurement", page 26.

Table 80: Telegram structure: sRN LCMcfg

Telegram structure: sRN LCMcfg						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E
Command		String	6		LCMcfg	4C 43 4D 63 66 67

Table 81: Example: sRN LCMcfg

CoLa A	<STX>sRN{SPC}LCMcfg<ETX>
	<STX>sRN LCMcfg<ETX>
	sRN LCMcfg
	02 73 52 4E 20 4C 43 4D 63 66 67 03
CoLa B	02 02 02 02 00 00 00 0A 73 52 4E 20 4C 43 4D 63 66 67 6F
	73 52 4E 20 4C 43 4D 63 66 67

Table 82: Telegram structure: sRA LCMcfg

Telegram structure: sRA LCMcfg						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sRA	73 52 41

Telegram structure: sRA LCMcfg						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command		String	6		LCMcfg	4C 43 4D 63 66 67
Strategy	Strategy code	Enum_8	1	Inactive: High available: Available: Sensitive: Semi-sensitive:	0 1 2 3 4	00 01 02 03 04
Response time	Time lapse	Uint_16	2		+1d ... +60d (00h ... 3Ch)	00 00 ... 00 3C
Threshold warning	Threshold value	Uint_16	2		0d ... +100d (00h ... 64h)	00 00 ... 00 64
Threshold error	Threshold value	Uint_16	2		0d ... +100d (00h ... 64h)	00 00 ... 00 64

Table 83: Example: sRA LCMcfg

CoLa A	<STX>sRA{SPC}LCMcfg{SPC}1{SPC}1{SPC}1E{SPC}46<ETX>
	02 73 57 41 20 4C 43 4D 63 66 67 20 31 20 31 20 31 45 20 34 36 03
CoLa B	02 02 02 02 00 00 00 12 73 52 41 20 4C 43 4D 63 66 67 20 01 00 01 00 1E 00 46 18

12.2.1.4.2.13 Read contamination measurement detailed values [sRN CMContLvlM]

Get the individual visibility levels in percentage values of all 7 measurement sensors along the optics cover.

Table 84: Telegram structure: sRN CMContLvlM

Telegram structure: sRN CMContLvlM						
Telegram part	**Description**	**Variable**	**Length**	**Additional data**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E
Command		String	10		CMContLvlM	43 4D 43 6F 6E 74 4C 76 6C 4D

Table 85: Example: sRN CMContLvlM

CoLa A	<STX>sRN{SPC}CMContLvlM<ETX>
	<STX>sRN CMContLvlM<ETX>
	sRN CMContLvlM
	02 73 52 4E 20 43 4D 43 6F 6E 74 4C 76 6C 4D 03
CoLa B	02 02 02 02 00 00 00 0E 73 52 4E 20 43 4D 43 6F 6E 74 4C 76 6C 4D 6C
	73 52 4E 20 43 4D 43 6F 6E 74 4C 76 6C 4D

Table 86: Telegram structure: sRA CMContLvlM

Telegram structure: sRA CMContLvlM						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sRA	73 52 41
Command		String	10		CMContLvlM	43 4D 43 6F 6E 74 4C 76 6C 4D

Telegram structure: sRA CMContLvlM						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Contamination data for different channels	[% of transparency] in order of the different channels	Uint_8	1	Order of 7 channels: -25.8°/12.8°/51.4°/90°/ 128.6°/ 167.2°/205.8°	0d ... +100d (00h ... 64h)	00 ... 64

-25.8°- to 90°-channel: 100 %, 128.6°- and 167.2°-channel: 84%, 205.8°-channel 83 % availability:

Table 87: Example for LMS5xx: sRA CMContLvlM

CoLa A	<STX>sRA{SPC}CMContLvlM{SPC}64{SPC}64{SPC}64{SPC}64{SPC}54{SPC}54{SPC}53<ETX>
	02 73 52 41 20 43 4D 43 6F 6E 74 4C 76 6C 4D 20 64 64 64 64 54 54 53 03
CoLa B	02 02 02 02 00 00 00 16 73 52 41 20 43 4D 43 6F 6E 74 4C 76 6C 4D 20 64 64 64 64 54 54 53 10

12.2.1.4.2.14 Read contamination status of the device [sRN LCMstate]

Read the current result of the contamination measurement.

Table 88: Telegram structure: sRN LCMstate

Telegram structure: sRN LCMstate						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E
Command	Status of LMS	String	8		LCMstate	4C 43 4D 73 74 61 74 65

Table 89: Example: sRN LCMstate

CoLa A	<STX>sRN{SPC}LCMstate<ETX>	
	<STX>sRN LCMstate<ETX>	
	sRN LCMstate	
	02 73 52 4E 20 4C 43 4D 73 74 61 74 65 03	
CoLa B	02 02 02 02 00 00 00 0C 73 52 4E 20 4C 43 4D 73 74 61 74 65 7A	
	73 52 4E 20 4C 43 4D 73 74 61 74 65	

Table 90: Telegram structure: sRA LCMstate

Telegram structure: sRA LCMstate						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sRA	73 52 41
Command	Status of LMS	String	8		LCMstate	4C 43 4D 73 74 61 74 65
Status code		Enum_8	1	No contamination: Contamination warning: Contamination error: Contamination measurement functionality defective:	0 1 2 3	00 01 02 03

Table 91: Example: sRA LCMstate

CoLa A	<STX>sRA{SPC}LCMstate{SPC}0<ETX>
	02 73 52 41 20 4C 43 4D 73 74 61 74 65 20 30 03
CoLa B	02 02 02 02 00 00 00 0E 73 52 41 20 4C 43 4D 73 74 61 74 65 20 00 55

12.2.1.4.2.15 Save parameters permanently [sMN mEEwriteall]

Save all parameter changes of the device. Must be sent before loging off and/ or hardware rebooting of the device. Else all changes will be lost.

Table 92: Telegram structure: sMN mEEwriteall

Telegram structure: sMN mEEwriteall (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3		sMN	73 4D 4E
Command	Store parameters permanently	String	11		mEEwriteall	6D 45 45 77 72 69 74 65 61 6C 6C

Table 93: Example: sMN mEEwriteall

CoLa A	<STX>sMN{SPC}mEEwriteall<ETX>
	<STX>sMN mEEwriteall<ETX>
	sMN mEEwriteall
	02 73 4D 4E 20 6D 45 45 77 72 69 74 65 61 6C 6C 03
CoLa B	02 02 02 02 00 00 00 0F 73 4D 4E 20 6D 45 45 77 72 69 74 65 61 6C 6C 21
	73 4D 4E 20 6D 45 45 77 72 69 74 65 61 6C 6C

Table 94: Telegram structure: sAN mEEwriteall

Telegram structure: sAN mEEwriteall						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sAN	73 41 4E
Command	Store parameters permanently	String	11		mEEwriteall	6D 45 45 77 72 69 74 65 61 6C 6C
Status code	Accepted when value is 1	Bool_1	1	Error: Success:	0 1	00 01

Table 95: Example: sAN mEEwriteall

CoLa A	<STX>sAN{SPC}mEEwriteall{SPC}1<ETX>
	02 73 41 4E 20 6D 45 45 77 72 69 74 65 61 6C 6C 20 31 03
CoLa B	02 02 02 02 00 00 00 11 73 41 4E 20 6D 45 45 77 72 69 74 65 61 6C 6C 20 01 0C

12.2.1.4.2.16 Set to run [sMN Run]

Log out from device and activate all parameter changes.

Table 96: Telegram structure: sMN Run

Telegram structure: sMN Run						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3		sMN	73 4D 4E
Command	Start the device	String	3		Run	52 75 6E

Table 97: Example: sMN Run

CoLa A	<STX>sMN{SPC}Run<ETX>	
	<STX>sMN Run<ETX>	
	sMN Run	
	02 73 4D 4E 20 52 75 6E 03	
CoLa B	02 02 02 02 00 00 00 07 73 4D 4E 20 52 75 6E 19	
	73 4D 4E 20 52 75 6E	

Table 98: Telegram structure: sAN Run

Telegram structure: sAN Run						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sAN	73 41 4E
Command	Start the device	String	3		Run	52 75 6E
Status code	Accepted when value is 1	Bool_1	1	Error: Success:	0 1	00 01

Table 99: Example: sAN Run

CoLa A	<STX>sAN{SPC}Run{SPC}1<ETX>
	02 73 41 4E 20 52 75 6E 20 31 03
CoLa B	02 02 02 02 00 00 00 09 73 41 4E 20 52 75 6E 20 01 34

12.2.1.4.2.17 Reboot device [sMN mSCreboot]

This command includes saving all parameters.

Table 100: Telegram structure: sMN mSCreboot

Telegram structure: sMN mSCreboot (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3		sMN	73 4D 4E
Command	Reboot device	String	9		mSCreboot	6D 53 43 72 65 62 6F 6F 74

Table 101: Example: sMN mSCreboot

CoLa A	<STX>sMN{SPC}mSCreboot<ETX>	
	<STX>sMN mSCreboot<ETX>	
	sMN mSCreboot	
	02 73 4D 4E 20 6D 53 43 72 65 62 6F 6F 74 03	

CoLa B	02 02 02 02 00 00 00 0D 73 4D 4E 20 6D 53 43 72 65 62 6F 6F 74 2C 73 4D 4E 20 6D 53 43 72 65 62 6F 6F 74

Table 102: Telegram structure: sAN mSCreboot

Telegram structure: sAN mSCreboot						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sAN	73 41 4E
Command	Reboot device	String	9		mSCreboot	6D 53 43 72 65 62 6F 6F 74

Table 103: Example: sAN mSCreboot

CoLa A	<STX>sAN{SPC}mSCreboot<ETX>
	02 73 41 4E 20 6D 53 43 72 65 62 6F 6F 74 03
CoLa B	02 02 02 02 00 00 00 0E 73 41 4E 20 6D 53 43 72 65 62 6F 6F 74 20 00

12.2.1.4.3 Measurement output telegram

12.2.1.4.3.1 Configure the data content for the scan [sWN LMDscandatacfg]

Configuration of the channels (e.g. distance) and information that will be send by the device as measurement data (output data format).

NOTE
For actually transmitting encoder values, the type of encoder has to be selected beforehand (see "Encoder", page 139).

Table 104: Telegram structure: sWN LMDscandatacfg

Telegram structure: sWN LMDscandatacfg (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3		sWN	73 57 4E
Command	Configure scandata	String	14		LMDscanda-tacfg	4C 4D 44 73 63 61 6E 64 61 74 61 63 66 67
Data channel	Defines the telegram content (DIST)	Uint_8	2	Output channel 1: Output channel 2: Output channel 1+2:	1 0 2 0 3 0	01 00 02 00 03 00
RSSI	Enable or disable the output of RSSI values	Uint_8	1	RSSI disabled: RSSI enabled:	0 1	00 01
Resolution	Resolution of RSSI	Enum_8	1	8 Bit: 16 Bit:	0 1	00 01
Unit	Unit of RSSI	Enum_8	1	Digits:	0	00
Encoder	Encoder data	Uint_8	2	No encoder: Channel 1:	0 0 1 0	00 00 01 00
Position	Position values	Bool_1	1	No: Yes:	0 1	00 01
Device name	Sends the device name	Bool_1	1	No: Yes:	0 1	00 01

Telegram structure: sWN LMDscandatacfg (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Comment	Saved comment	Bool_1	1	No: Yes:	0 1	00 01
Time	Sends time stamp information	Bool_1	1	No: Yes:	0 1	00 01
Output rate	Sends the output rate	Uint_16	2	All scans: Each 2nd scan: Max. Each 50000th scan:	+1d (1h) +2d (2h) +50000d (C350h)	00 01 00 02 C3 50

Example 1: output channel 1, remission, resolution 16Bit, no encoder and output rate all scans

Table 105: Example 1: sWN LMDscandatacfg

CoLa A	<STX>sWN{SPC}LMDscandatacfg{SPC}01{SPC}00{SPC}1{SPC}1{SPC}0{SPC}00{SPC}00{SPC}0{SPC}0{SPC}0{SPC}0{SPC}+1<ETX>	
	<STX>sWN LMDscandatacfg 01 0 1 1 0 00 00 0 0 0 0 +1<ETX>	
	sWN LMDscandatacfg 01 00 1 1 0 00 00 0 0 0 0 +1	
	02 73 57 4E 20 4C 4D 44 73 63 61 6E 64 61 74 61 63 66 67 20 30 31 20 30 30 20 31 20 31 20 30 20 30 30 20 30 30 20 30 20 30 20 30 20 30 20 2B 31 03	
CoLa B	02 02 02 02 00 00 00 20 73 57 4E 20 4C 4D 44 73 63 61 6E 64 61 74 61 63 66 67 20 01 00 01 01 00 00 00 00 00 00 00 00 01 42	
	73 57 4E 20 4C 4D 44 73 63 61 6E 64 61 74 61 63 66 67 20 01 00 01 01 00 00 00 00 00 00 00 00 01	

Example 2: output channel 1, remission, resolution 16Bit, no encoder, each 10th scan

Table 106: Example 2: sWN LMDscandatacfg

CoLa A	<STX>sWN{SPC}LMDscandatacfg{SPC}01{SPC}00{SPC}1{SPC}1{SPC}0{SPC}00{SPC}00{SPC}0{SPC}0{SPC}0{SPC}0{SPC}+10<ETX>	
	<STX>sWN LMDscandatacfg 01 00 1 1 0 00 00 0 0 0 0 +10<ETX>	
	sWN LMDscandatacfg 01 00 1 1 0 00 00 0 0 0 0 +10	
	02 73 57 4E 20 4C 4D 44 73 63 61 6E 64 61 74 61 63 66 67 20 30 31 20 30 30 20 30 20 31 20 30 20 30 30 20 30 30 20 30 20 30 20 30 20 30 20 2B 31 30 03	
CoLa B	02 02 02 02 00 00 00 20 73 57 4E 20 4C 4D 44 73 63 61 6E 64 61 74 61 63 66 67 20 01 00 00 01 00 00 00 00 00 00 00 00 10 52	
	73 57 4E 20 4C 4D 44 73 63 61 6E 64 61 74 61 63 66 67 20 01 00 00 01 00 00 00 00 00 00 00 00 10	

Example 3: output channel 2, resolution 16Bit, encoder active and output rate each 10th scan

Table 107: Example3: sWN LMDscandatacfg

CoLa A	<STX>sWN{SPC}LMDscandatacfg{SPC}02{SPC}00{SPC}0{SPC}1{SPC}0{SPC}01{SPC}00{SPC}0{SPC}0{SPC}0{SPC}0{SPC}+10<ETX>	
	<STX>sWN LMDscandatacfg 02 00 0 1 0 01 00 0 0 0 0 +10<ETX>	
	sWN LMDscandatacfg 02 00 0 1 0 01 00 0 0 0 0 +10	
	02 73 57 4E 20 4C 4D 44 73 63 61 6E 64 61 74 61 63 66 67 20 30 32 20 30 20 30 20 31 20 30 20 30 31 20 30 20 30 20 30 20 30 20 30 20 2B 31 30 03	
CoLa B	02 02 02 02 00 00 00 20 73 57 4E 20 4C 4D 44 73 63 61 6E 64 61 74 61 63 66 67 20 02 00 00 01 00 01 00 00 00 00 00 00 4A 0A	
	73 57 4E 20 4C 4D 44 73 63 61 6E 64 61 74 61 63 66 67 20 02 00 00 01 00 01 00 00 00 00 00 00 4A	

Table 108: Telegram structure: sWA LMDscandatacfg

Telegram structure: sWA LMDscandatacfg						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sWA	73 57 41
Command	Configure scandata	String	14		LMDscanda-tacfg	4C 4D 44 73 63 61 6E 64 61 74 61 63 66 67

Table 109: Example: sWA LMDscandatacfg

CoLa A	<STX>sWA{SPC}LMDscandatacfg<ETX>
	02 73 57 41 20 4C 4D 44 73 63 61 6E 64 61 74 61 63 66 67 03
CoLa B	02 02 02 02 00 00 00 13 73 57 41 20 4C 4D 44 73 63 61 6E 64 61 74 61 63 66 67 20 4D

12.2.1.4.3.2 Configure aperture angle of the scandata for output [sWN LMPoutputRange]

Select start and stop angle of the measurement data output. In general only one output range can be be configured.

NOTE
Verify the definition of the angle positions for your product.

Table 110: Telegram structure: sWN LMPoutputRange

Telegram structure: sWN LMPoutputRange (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional Details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3		sWN	73 57 4E
Command	Change output angle range	String	14		LMPoutputRange	4C 4D 50 6F 75 74 70 75 74 52 61 6E 67 65
Reserved	-	Int_16	2	Always:	1	00 01
Angular resolu-tion [1)]	[1/10000°]	Uint_32	4	0.25°: 0.5°:	+2500d (9C4h) +5000d (1388h)	00 00 09 C4 00 00 13 88
Start angle	[1/10000°]	Int_32	4	Fixed: -45°	-450000d (FFF92230h)	FF F9 22 30
Stop angle	[1/10000°]	Int_32	4	Fixed: +225°	+2250000d (225510h)	00 22 55 10

1) Angular resolution can not be changed here, it is taken automatically from the basic scan settings!

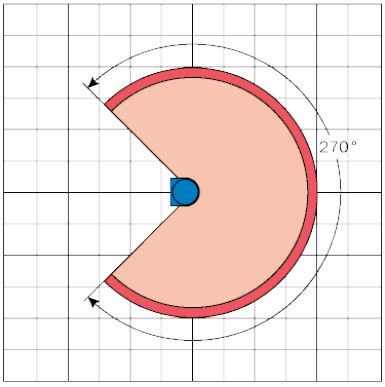


Figure 59: Example - 1 measurement sector of 270°

Table 111: Example: sWN LMPoutputRange - set output data for angular resolution at 0.50° and range from -45° to 225°

CoLa A	<STX>sWN{SPC}LMPoutputRange{SPC}1{SPC}+5000{SPC}-450000{SPC}+2250000<ETX>
	<STX>sWN LMPoutputRange 1 +5000 -450000 +2250000<ETX>
	sWN LMPoutputRange 1 +5000 -450000 +2250000
	02 73 57 4E 20 4C 4D 50 6F 75 74 70 75 74 52 61 6E 67 65 20 31 20 2B 35 30 30 30 20 2D 34 35 30 30 30 30 20 2B 32 32 35 30 30 30 30 03
CoLa B	02 02 02 02 00 00 00 21 73 57 4E 20 4C 4D 50 6F 75 74 70 75 74 52 61 6E 67 65 20 00 01 00 00 13 88 FF F9 22 30 00 22 55 10 92
	73 57 4E 20 4C 4D 50 6F 75 74 70 75 74 52 61 6E 67 65 20 00 01 00 00 13 88 FF F9 22 30 00 22 55 10

Table 112: Telegram structure: sWA LMPoutputRange

Telegram structure: sWA LMPoutputRange						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sWA	73 57 41
Command	Change output angle range	String	14		LMPoutpu-tRange	4C 4D 50 6F 75 74 70 75 74 52 61 6E 67 65

Table 113: Example: sWA LMPoutputRange

CoLa A	<STX>sWA{SPC}LMPoutputRange<ETX>
	02 73 57 41 20 4C 4D 50 6F 75 74 70 75 74 52 61 6E 67 65 03
CoLa B	02 02 02 02 00 00 00 13 73 57 41 20 4C 4D 50 6F 75 74 70 75 74 52 61 6E 67 65 20 74

12.2.1.4.3.3 Read for actual output range [sRN LMPoutputRange]

Read the defined angular resolution and current aperture angle for data output.

Table 114: Telegram structure: sRN LMPoutputRange

Telegram structure: sRN LMPoutputRange						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E
Command	Output range	String	14		LMPoutpu-tRange	4C 4D 50 6F 75 74 70 75 74 52 61 6E 67 65

Table 115: Example: sRN LMPoutputRange

CoLa A	<STX>sRN{SPC}LMPoutputRange<ETX>	
	<STX>sRN LMPoutputRange<ETX>	
	sRN LMPoutputRange	
	02 73 52 4E 20 4C 4D 50 6F 75 74 70 75 74 52 61 6E 67 65 03	
CoLa B	02 02 02 02 00 00 00 12 73 52 4E 20 4C 4D 50 6F 75 74 70 75 74 52 61 6E 67 65 5E	
	73 52 4E 20 4C 4D 50 6F 75 74 70 75 74 52 61 6E 67 65	

Table 116: Telegram structure: sRA LMPoutputRange

Telegram structure: sRA LMPoutputRange						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sRA	73 52 41
Command	Output range	String	14		LMPoutpu-tRange	4C 4D 50 6F 75 74 70 75 74 52 61 6E 67 65
Reserved		Int_16	2	Always:	1h	00 01
Angular resolution	[1/10000°]	Uint_32	4	0.25°: 0.5°:	9C4h 1388h	00 00 09 C4 00 00 13 88
Start angle	[1/10000°]	Int_32	4	Fixed: -45°	FFF92230h	FF F9 22 30
Stop angle	[1/10000°]	Int_32	4	Fixed: +225°	225510h	00 22 55 10

Table 117: Example: sRA LMPoutputRange – device output set at 0.5° angular resolution and range from -45° to +225°

CoLa A	<STX>sRA{SPC}LMPoutputRange{SPC}1{SPC}1388{SPC}FFF92230{SPC}225510<ETX>
	02 73 52 41 20 4C 4D 50 6F 75 74 70 75 74 52 61 6E 67 65 20 31 20 31 33 38 38 20 46 46 46 39 32 32 33 30 20 32 32 35 35 31 30 03
CoLa B	02 02 02 02 00 00 00 21 73 52 41 20 4C 4D 50 6F 75 74 70 75 74 52 61 6E 67 65 20 00 01 00 00 13 88 FF F9 22 30 00 22 55 10 98

12.2.1.4.3.4 Poll one telegram [sRN LMDscandata]

Asking the device for the measurement values of the last valid scan. The device will respond, even if currently no measurement data is created (e.g. due to standby or log in).

NOTE
After changing the scanning frequency, there will be no data telegram or answer from the device for up to 30 seconds.The same applies when the device is powering up or rebooting.

Table 118: Telegram structure: sRN LMDscandata

Telegram structure: sRN LMDscandata						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E
Command	Only one telegram	String	11		LMDscandata	4C 4D 44 73 63 61 6E 64 61 74 61

Table 119: Example: sRN LMDscandata

CoLa A	<STX>sRN{SPC}LMDscandata<ETX>
	<STX>sRN LMDscandata<ETX>
	sRN LMDscandata
	02 73 52 4E 20 4C 4D 44 73 63 61 6E 64 61 74 61 03
CoLa B	02 02 02 02 00 00 00 0F 73 52 4E 20 4C 4D 44 73 63 61 6E 64 61 74 61 05
	73 52 4E 20 4C 4D 44 73 63 61 6E 64 61 74 61

Table 120: Telegram structure: sRA LMDscandata

Telegram structure: sRA LMDscandata						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Find complete telegram structure of the answer in "Send data permanently [sEN LMDscandata]", page 119.						

12.2.1.4.3.5 Send data permanently [sEN LMDscandata]

Start/ stop continuous retrievement of measurement data from device. Data will be transmitted as configured in "Configure the data content for the scan [sWN LMDscandatacfg]", page 114 as soon as measurement data is generated by the device. No data is generated when there is an acitve log in, the laser is shut off (e.g. in Standby mode), the motor is stopped or in case of certain error modes (e.g. Device not ready).

NOTE
After changing the scanning frequency, there will be no data telegram or answer from the device for up to 30 seconds.The same applies when the device is powering up or rebooting.

Table 121: Telegram structure: sEN LMDscandata

Telegram structure: sEN LMDscandata						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Event	String	3		sEN	73 45 4E
Command	Data telegram	String	11		LMDscandata	4C 4D 44 73 63 61 6E 64 61 74 61
Measure-ment	Start/stop	Enum_8	1	Stop: Start:	0 1	00 01

Table 122: Example: sEN LMDscandata

CoLa A	<STX>sEN{SPC}LMDscandata{SPC}1<ETX>
	<STX>sEN LMDscandata 1<ETX>
	sEN LMDscandata 1
	02 73 45 4E 20 4C 4D 44 73 63 61 6E 64 61 74 61 20 31 03
CoLa B	02 02 02 02 00 00 00 11 73 45 4E 20 4C 4D 44 73 63 61 6E 64 61 74 61 20 01 33
	73 45 4E 20 4C 4D 44 73 63 61 6E 64 61 74 61 20 01

Table 123: Telegram structure: sEA LMDscandata

Telegram structure: sEA LMDscandata						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sEA	73 45 41
Command	Data telegram	String	11		LMDscandata	4C 4D 44 73 63 61 6E 64 61 74 61
Measurement	Start/stop	Enum_8	1	Stop: Start:	0 1	00 01

Table 124: Example: Confirmation of sEA LMDscandata

CoLa A	<STX>sEA{SPC}LMDscandata{SPC}1<ETX>
	02 73 45 41 20 4C 4D 44 73 63 61 6E 64 61 74 61 20 31 03
CoLa B	02 02 02 02 00 00 00 11 73 45 41 20 4C 4D 44 73 63 61 6E 64 61 74 61 20 01 3C

Telegram stream

The answer to the telegram will be followed by the scandata:

Table 125: Telegram structure: Datastream of sRA LMDscandata/sSN LMDscandata

Telegram structure: sRA LMDscandata / sSN LMDscandata						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	Answer to sRN LMDscandata: Answer to sEN LMDscandata:	sRA sSN	73 52 41 73 53 4E
Command	Data telegram	String	11		LMDscandata	4C 4D 44 73 63 61 6E 64 61 74 61
Version number	For detecting format changes by the version.	Uint_16	2		1h ... FFFFh	00 01 ... FF FF
				Device		
Device number	Defined with SOPAS	Uint_16	2		0h ... FFFFh	00 00 ... FF FF
Serial number	Production period (year, calendar week, number): YYWWxxxx	Uint_32	4		0h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
Device status	(See values column)	Uint_8	2 x 1	Ok: Error: Pollution warning: Pollution warning with device error: Pollution error with no device error: Pollution error with device error:	0 0 1 0 2 0 3 0 4 0 5 0	00 00 01 00 02 00 (00 10 00 00) 03 00 (00 11 00 00) 04 00 (01 00 00 00) 05 00 (01 01 00 00)
				Status info		

Telegram structure: sRA LMDscandata / sSN LMDscandata						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Telegram counter	Number of measurement telegrams finished in the scanner and given to the interface. Does not count how many telegrams were really given out; is relevant if not all scans are delivered from the scan core.	Uint_16	2		0h ... FFFFh	00 00 ... FF FF
Scan counter	Number of scans which were created in the device; counts how many scans were really done.	Uint_16	2		0h ... FFFFh	00 00 ... FF FF
Time since start up in µs	Counting the time since power up the device; starting with 0. In the output telegram this is the time at the zero index before the measurement itself starts.	Uint_32	4		0h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
Time of transmission in µs	Time in µs when the complete scan is transmitted to the buffer for data output; starting with 0 at scanner bootup.	Uint_32	4		0h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
Status of digital inputs	Low byte represents input 1.	Uint_8	2 x 1	All inputs low: All inputs high:	0 0 3 0	00 00 03 00
Status of digital outputs	Low byte represents output 1.	Uint_8	2 x 1	All outputs low: All internal outputs high: All outputs high (incl. Ext. Out):	0 0 7 0 07 FF	00 00 07 00 07 FF
Reserved	-	Uint_16	2	Always:	0	00 00 00 00
Frequencies						
Scan frequency	[1/100 Hz]	Uint_32	4	25 Hz: 50 Hz:	9C4h 1388h	09 C4 13 88
Measurement frequency	Inverse of the time between two measurement shots (in 100 Hz).	Uint_32	4	Example: 50 Hz, 0.5° (360/0.5)/(1/50) = 36 kHz	00000000h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
Encoder						
Amount of encoder	If 0, then next two telegram parts are missing.	Enum_16	2		0 ... 3	00 00 ... 00 03
Encoder position	Info in ticks	Uint_32	4	Only filled if parameter is activated	0h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
Encoder speed	mm/sec or milligrad/sec	Uint_16	2	Only filled if parameter is activated	0h ... FFFFh	00 00 ... FF FF
16 bit output channel (Distance and RSSI)						
Amount of 16 bit channels	Number of 16 bit channels that provide measured data	Enum_16	2	Output channels	1, 2 or 4	01, 02 or 04

Telegram structure: sRA LMDscandata / sSN LMDscandata						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Content	Defines the content of the output channel Radial distance values (DIST) in mm	String	5	Distance values of first pulse: Distance values of second pulse: Energy values of first pulse: Energy values of second pulse:	DIST1 DIST2 RSSI1 RSSI2	44 49 53 54 31 44 49 53 54 32 52 53 53 49 31 52 53 53 49 32
Scale factor	Scale factor or factor of the measurement values	Real as float according to IEEE754	4	Factor x 1: Factor x 2:	3F800000h 40000000h	3F 80 00 00 40 00 00 00
Scale factor offset	Sets starting point of measurement	Real as float according to IEEE754	4		00000000	00 00 00 00
Start angle	[1/10000°]	Int_32	4	Fixed: -45°	FFF92230h	FF F9 22 30
Size of single angular step	Output format in degree: 1/10000°	Uint_16	2	(depends on the angular resolution see "Configure aperture angle of the scandata for output [sWN LMPoutputRange]", page 116)	9C4h ... 1388h	09 C4 ... 13 88
Amount of data	Defines the number of items on measured output	Uint_16	2		0h ... FFFFh	00 00 ... FF FF
8 bit output channel						
Amount of 8 bit channels	Number of 8 bit channels that provide measured data	Enum_16	2	Output channels:	1 or 2	01 or 02
Content	Defines the content of the output channel	String	5		RSSI1 RSSI2	52 53 53 49 31 52 53 53 49 32
Scale factor	Scale factor or factor of the measurement values	Real as float according to IEEE754	4	Factor x 1: Factor x 2 (values have to be scaled by factor two):	3F800000h 40000000h	3F 80 00 00 40 00 00 00
Reserved	-	Real as float according to IEEE754	4	Always:	00000000h	00 00 00 00
Start angle	Output format: [1/10000°]	Uint_32	4	Fixed: -45°	FFF92230h	FF F9 22 30
Size of single angular step	Output format: 1/10000°	Uint_16	2	(depends on the angular resolution see "Configure aperture angle of the scandata for output [sWN LMPoutputRange]", page 116)	3E8h ... 2710h	03 E8 ... 27 10
Amount of data	Defines the number of items on measured output	Uint_16	2		0h ... FFFFh	00 00 ... FF FF
Data_1 Data_n	Data stream starting Data_1 to Data_n	Uint_8	1		00h ... FFh	00 ... FF
Reserved						
Reserved	-	Enum_16	2	Always:	0	00 00
Name						

Telegram structure: sRA LMDscandata / sSN LMDscandata						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Name	Device name	Enum_16	2	No device name defined: Device name defined:	0 1	00 00 00 01
Length	Length of name	Uint_16	2	Only filled if parameter is activated	0h ... Fh	00 ... 0F
Name	Device name in characters	String	16		20h ... 7Ah	20 ... 7A
				Reserved		
Reserved	-	Enum_16	2	Always:	0	00 00
				Time		
Time	Transmits a time stamp	Enum_16	2	No time: Time:	0 1	00 00 00 01
Year	1970 to 2037	Uint_16	2	Only filled if parameter is activated	7B2h ... 07F5h	07 B2 ... 07 F5
Month	1 to 12	Uint_8	1		0h ... Ch	00 ... 0C
Day	Day of month 1 to 31	Uint_8	1		0h ... 1Fh	00 ... 1F
Hour	0 to 23	Uint_8	1		0h ... 17h	00 ... 17
Minute	0 to 59	Uint_8	1		0h ... 3Bh	00 ... 3B
Second	0 to 59	Uint_8	1		0h ... 3Bh	00 ... 3B
Micro-second	0 to 999999	Uint_32	4		0h ... F423Fh	00 00 00 00 ... 00 0F 42 3F
				Reserved		
Reserved	-	Uint_16	2	Always:	0	00 00

LMDscandata - reserved values

Valid distance measurement values are values starting from 16d upwards; everything below has the following meaning:

DIST	RSSI	Description
0d	0d	Invalid measurement value; caused by very low remission (extremely dark object), object distance not within measurement range (too close or too far away) or selected filter settings at device
1d	FFFFh (16Bit output) FFh (8Bit output)	Invalid measurement value, device was dazzled or blinded, e.g. by measuring into the sun
2d	0h	Implausible measurement values
3d	0h	Value was set to invalid by a filter (Echo Filter, Particle Filter)
4d – 15d	0d	Reserved, currently not in use
≥16d	>0h	Valid measurement values

Max. measurement value (LMS10x, LMS11x, LMS12x, LMS13x): Dez: 20 000 mm --> Hex: 4E20

Max. measurement value (LMS14x): Dez: 20 000 mm --> Hex: 4E20

Max. measurement value (LMS15x, LMS16x): Dez: 20 000 mm --> Hex: 4E20

Higher measurement values will be given out with a zero, that means no measurement value detected.

Calculation and amount of data

Example how to calculate the amount of data for a measurement telegram.

Sizes of values and telegram parts:

- one measurement value: 5 byte (4 byte value itself, 1 byte space after the value)
- one RSSI value: 3 byte (2 byte value itself, 1 byte space after the value)
- telegram header: 81 byte
- telegram end: 12 byte

Calculation of number of Measurement values depends always on the resolution:

0.5° = 2 measurements per degree
0.25° = 4 measurements per degree
Always one additional measurement for the last measurement
Number of measurement values = Number of degrees × measurements per degree + 1
Example for measurement of 56° in 0.5° resolution (without RSSI data):
56 × 2 + 1 = 113 Measurement values
Amount of Data for these measurement values:
113 × 5 Byte = 565 Byte

Calculation of amount of data per telegram:
Data of one Telegram = Header + Measurements + end of telegram
81 Byte + 113 Measurements + 12 Byte
81 Byte + (113 × 5 Byte) + 12 Byte = 658 Byte per Telegram (= 5264 Bit (658 × 8 Bit))

Possible amout for delivery with special Speed:
Number of telegrams per second = Speed ÷ telegram size
Speed Example:
115200 Bit/s = 11520 Byte/s = 11,52 Byte/s
11520 (Byte/s) ÷ 658 Byte = 17,5 Telegrams/s

Telegram size with **0.25°**resolution:
Degrees: 270°
Resolution: 0.25°
Measurement values = 270 × 4 + 1 = 1081
Data per telegram = 81 Byte + (1081 × 5 Byte) + 12 Byte = 5498 Byte (= 43984 Bit)
Telegram size with **0.5°**resolution:
Degrees: 270°
Resolution: 0.5°
Measurement values = 270 × 2 + 1 = 541
Data per telegram = 81 Byte + (541 × 5 Byte) + 12 Byte = 2798 Byte (= 22384 Bit)
As a result in that configuration a 10 MBit connection will not be enough. With a 100 MBit Hub, 3-4 scanner can be used, with a 1 GBit Hub accordingly more.

Example of a telegram stream

Table 126: Telegram structure: sRA LMDscandata (Example)

Telegram structure: sRA LMDscandata (Example)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRA	73 52 41
Command	Data telegram	String	11		LMDscandata	4C 4D 44 73 63 61 6E 64 61 74 61
Version number	For detecting format changes by the version	Uint_16	2		1	00 01
Device						
Device number	Defined with SOPAS	Uint_16	2		1	00 01
Serial number	Production period (year, calendar week, number): YYWWxxxx	Uint_32	4		89A27Fh	00 89 A2 7F
Device status	(See values column)	Uint_8	2 x 1	Ok:	0 0	00 00
Status info						

Telegram structure: sRA LMDscandata (Example)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Telegram counter	Number of measurement telegrams finished in the scanner and given to the interface. Does not count how many telegrams were really given out; is relevant if not all scans are delivered from the scan core.	Uint_16	2		343h	03 43
Scan counter	Number of scans which were created in the device; counts how many scans were really done.	Uint_16	2		347h	03 47
Time since start up in µs	Counting the time since power up the device; starting with 0. In the output telegram this is the time at the zero index before the measurement itself starts.	Uint_32	4		27477BA9h	27 47 7B A9
Time of transmission in µs	Time in µs when the complete scan is transmitted to the buffer for data output; starting with 0 at scanner bootup.	Uint_32	4		2747813Bh	27 47 81 3B
Status of digital inputs	Low byte represents input 1.	Uint_8	2 x 1		0 0	00 00
Status of digital outputs	Low byte represents output 1.	Uint_8	2 x 1	Corresponds to status 0111	7 0	07 00
Reserved	-	Uint_16	2	Always:	0	00 00
Frequencies						
Scan frequency	[1/100 Hz]	Uint_32	4	5000/100 = 50 Hz	1388 h	00 00 13 88
Measurement frequency	Inverse of the time between two measurement shots (in 100 Hz). Example: 50 Hz, 0.5° resolution → 720 shots/20 ms → 36 kHz Example: 12.5 Hz, 0.04° resolution → 9000 shots/80 ms → 112.5 kHz	Uint_32	4		168h	00 00 01 68
Encoder						
Amount of encoder	If 0, then next two values are missing.	Enum_16	2	No encoder data	0	00 00
16 bit output channel						
Amount of 16 bit channels	Number of 16 bit channels that provide measured data	Enum_16	2		1	00 01

Telegram structure: sRA LMDscandata (Example)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Content	Defines the content of the output channel Unit of radial distance values (DIST) is mm	String	5		DIST1	44 49 53 54 31
Scale factor	Scale factor or factor of the measurement values	Real as float according to IEEE754	4	Floating Point: Value = 1	3F800000h	3F 80 00 00
Scale factor offset	Sets starting point of measurement	Real as float according to IEEE754	4	Floating Point: Value = 0	00000000	00 00 00 00
Start angle	[1/10000°]	Uint_32	4	100000/10000 = 10°	186A0h	00 01 86 A0
Size of single angular step	Output format in degree: 1/10000°	Uint_16	2	5000/10000 = 0.5° 400/10000 = 0.04°	1388 h	13 88
Amount of data	Defines the number of items on measured output	Uint_16	2	21 Measurement points	15h	00 15
Data_1 Data_21	Data stream starting Data_1 to Data_21 No limit	Uint_16	2	Measurement data Min. 22 mm: 16h Max. 20000 mm: 4E20h Max. 40000 mm: 9C40h Max. 50000 mm: C350h	8A1 8A5 8AB 8AC 8A6 8AC 8B6 8C8 8C2 8C9 8CB 8C4 8E4 8E1 8EB 8E0 8F5 908 8FC 907 906	08 A1 08 A5 08 AB 08 AC 08 A6 08 AC 08 B6 08 C8 08 C2 08 C9 08 CB 08 C4 08 E4 08 E1 08 EB 08 E0 08 F5 09 08 08 FC 09 07 09 06
8 bit output channel						
Amount of 8 bit channels	Number of 8 bit channels that provide measured data	Enum_16	2	No 8 bit data	0	00 00
Position						
Position	Output of position data	Enum_16	2	No position data:	0	00 00
Name						
Name	Device name	Enum_16	2	No device name defined:	0	00 00
Comment						
Comment	Comment	Uint_16	2	No comment:	0	00 00
Time						
Time	Transmits a time stamp	Uint_16	2	No time:	0	00 00
Reserved						
Reserved	-	Uint_16	2	Always:	0	00 00

Table 127: Example: sRA LMDscandata

CoLa A	<STX>sRA{SPC}LMDscandata{SPC}1{SPC}1{SPC}89A27F{SPC}0{SPC}0{SPC}343{SPC}347{SPC}27477BA9{SPC}2747813B{SPC}0{SPC}0{SPC}7{SPC}0{SPC}0{SPC}1388{SPC}168{SPC}0{SPC}1{SPC}DIST1{SPC}3F800000{SPC}00000000{SPC}186A0{SPC}1388{SPC}15{SPC}8A1{SPC}8A5{SPC}8AB{SPC}8AC{SPC}8A6{SPC}8AC{SPC}8B6{SPC}8C8{SPC}8C2{SPC}8C9{SPC}8CB{SPC}8C4{SPC}8E4{SPC}8E1{SPC}8EB{SPC}8E0{SPC}8F5{SPC}908{SPC}8FC{SPC}907{SPC}906{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0<ETX>
CoLa B	02 02 02 02 00 00 00 83 73 52 41 20 4C 4D 44 73 63 61 6E 64 61 74 61 20 00 01 00 01 00 89 A2 7F 00 00 03 43 03 47 27 47 7B A9 27 47 81 3B 00 00 07 00 00 00 00 00 13 88 00 00 01 68 00 00 00 01 44 49 53 54 31 3F 80 00 00 00 00 00 00 00 01 86 A0 13 88 00 15 08 93 08 95 08 AF 08 B3 08 B0 08 A4 08 B0 08 BF 08 B9 08 BA 08 D0 08 D3 08 CF 08 DE 08 EB 08 E3 08 FE 08 EC 09 03 08 FD 08 FD 00 00 00 00 00 00 00 00 00 00 00 00 2B

12.2.1.4.4 Time stamp

12.2.1.4.4.1 Set time synchronization [sWN TSCRole]

Set the device synchronization mode.

Table 128: Telegram structure: sWN TSCRole

Telegram structure: sWN TSCRole (User Level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3		sWN	73 57 4E
Command	Set timestamp role	String	7		TSCRole	54 53 43 52 6F 6C 65
Status	Timestamp role	Uint_8	1	Off: Client: Server:	0 1 2	00 01 02

Table 129: Example: sWN TSCRole

CoLa A	<STX>sWN{SPC}TSCRole{SPC}1<ETX>	
	<STX>sWN TSCRole 1<ETX>	
	sWN TSCRole 1	
	02 73 57 4E 20 54 53 43 52 6F 6C 65 20 31 03	
CoLa B	02 02 02 02 00 00 00 0D 73 57 4E 20 54 53 43 52 6F 6C 65 20 01 1B	
	73 57 4E 20 54 53 43 52 6F 6C 65 20 01	

Table 130: Telegram structure: sWA TSCRole

Telegram structure: sWA TSCRole						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sWA	73 57 41
Command	Set NTP role	String	7		TSCRole	54 53 43 52 6F 6C 65

Table 131: Example: sWA TSCRole

CoLa A	<STX>sWA{SPC}TSCRole<ETX>
	02 73 57 41 20 54 53 43 52 6F 6C 65 03
CoLa B	02 02 02 02 00 00 00 0C 73 57 41 20 54 53 43 52 6F 6C 65 20 15

12.2.1.4.4.2 Set time stamp [sMN LSPsetdatetime]

The data format in the telegram is: +2009{SPC}+7{SPC}+22{SPC}+12{SPC}+0{SPC}+0{SPC}+0.
The numbers represent year, month, day, hour, minute, second, microsecond.

If plus is used up-front the data it is interpreted as an integer decimal number, without the plus it's the scanner reads the data as hex format.

The answer is always in ASCII format.

NOTE
There is no real time clock inside the device. When the scanner is switched off and after a reboot, the time has to be set again. However, it is possible to analyze the Off-time in order to evade this issue.

Table 132: Telegram structure: sMN LSPsetdatetime

Telegram structure: sMN LSPsetdatetime (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3		sMN	73 4D 4E
Command	Set time stamp	String	14		LSPsetdatetime	4C 53 50 73 65 74 64 61 74 65 74 69 6D 65
Year		Uint_16	2		+1970d ... +2099d (07B2h ... 0833h)	07 B2 ... 08 33
Month		Uint_8	1		+1d ... +12d (01h ... 0Ch)	01 ... 0C
Day		Uint_8	1		+1d ... +31d (01h ... 1Fh)	00 ... 1F
Hour		Uint_8	1		+0d ... +23d (00h ... 17h)	00 ... 17
Minute		Uint_8	1		+0d ... +59d (00h ... 3Bh)	00 ... 3B
Second		Uint_8	1		+0d ... +59d (00h ... 3Bh)	00 ... 3B
Microsec-ond		Uint_32	4		+0d ... +999999d (00000000h ... 000F423Fh)	00 00 00 00 ... 00 0F 42 3F

Table 133: Example 1: sMN LSPsetdatetime Date: 17.02.2009; Time: 16:34

CoLa A	<STX>sMN{SPC}LSPsetdatetime{SPC}7D9{SPC}2{SPC}11{SPC}10{SPC}22{SPC}0{SPC}0<ETX>
	<STX>sMN LSPsetdatetime 7D9 2 11 10 22 0 0<ETX>
	sMN LSPsetdatetime 7D9 2 11 10 22 0 0
	02 73 4D 4E 20 4C 53 50 73 65 74 64 61 74 65 74 69 6D 65 20 37 44 39 20 32 20 31 31 20 31 30 20 32 32 20 30 20 30 03
CoLa B	02 02 02 02 00 00 00 1E 73 4D 4E 20 4C 53 50 73 65 74 64 61 74 65 74 69 6D 65 20 07 D9 02 11 10 22 00 00 00 00 00 B3
	73 4D 4E 20 4C 53 50 73 65 74 64 61 74 65 74 69 6D 65 20 07 D9 02 11 10 22 00 00 00 00 00

Table 134: Example 2: sMN LSPsetdatetime Date: 26.01.2010; Time: 10:35

CoLa A	<STX>sMN{SPC}LSPsetdatetime{SPC}+2010{SPC}+01{SPC}+26{SPC}+10{SPC}+35{SPC}0{SPC}0<ETX>
	<STX>sMN LSPsetdatetime +2010 +01 +26 +10 +35 0 0<ETX>
	sMN LSPsetdatetime +2010 +01 +26 +10 +35 0 0
	02 73 4D 4E 20 4C 53 50 73 65 74 64 61 74 65 74 69 6D 65 20 2B 32 30 31 30 20 2B 30 31 20 2B 32 36 20 2B 31 30 20 2B 33 35 20 2B 30 30 20 2B 30 30 30 30 03
CoLa B	02 02 02 02 00 00 00 1E 73 4D 4E 20 4C 53 50 73 65 74 64 61 74 65 74 69 6D 65 20 07 DA 01 1A 0A 23 00 00 00 00 00 B3
	73 4D 4E 20 4C 53 50 73 65 74 64 61 74 65 74 69 6D 65 20 07 DA 01 1A 0A 23 00 00 00 00 00

Table 135: Telegram structure: sAN LSPsetdatetime

Telegram structure: sAN LSPsetdatetime						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sAN	73 41 4E
Command	Set time stamp	String	14		LSPsetdatetime	4C 53 50 73 65 74 64 61 74 65 74 69 6D 65
Status code	Code number	Enum_8	1	Success:	1	01

Table 136: Example 1, 2: sAN LSPsetdatetime

CoLa A	<STX>sAN{SPC}LSPsetdatetime{SPC}1<ETX>
	02 73 41 4E 20 4C 53 50 73 65 74 64 61 74 65 74 69 6D 65 20 31 03
CoLa B	02 02 02 02 00 00 00 14 73 41 4E 20 4C 53 50 73 65 74 64 61 74 65 74 69 6D 65 20 01 51

Activate time stamp in the output string format or on SOPAS page “data processing”.

12.2.1.4.4.3 Read time stamp and status of the measurement function [sRN STlms]

Table 137: Telegram structure: sRN STlms

Telegram structure: sRN STlms						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E
Command	Status and time	String	5		STlms	53 54 6C 6D 73

Table 138: Example: sRN STlms

CoLa A	<STX>sRN{SPC}STlms<ETX>	
	<STX>sRN STlms<ETX>	
	sRN STlms	
	02 73 52 4E 20 53 54 6C 6D 73 03	
CoLa B	02 02 02 02 00 00 00 09 73 52 4E 20 53 54 6C 6D 73 3A	
	73 52 4E 20 53 54 6C 6D 73	

Table 139: Telegram structure: sRA STlms

Telegram structure: sRA STlms						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sRA	73 52 41
Command	Status and time	String	5		STlms	53 54 6C 6D 73
Status code	Status of the measurement function.	Enum_16	2	Boot cycle: Ready: Measurement active: FW Update: Error:	0 ... 5 6 7 8 10	00 00 ... 00 05 00 06 00 07 00 08 00 0A
Reserved	-	Uint_8	1	Always:	0	00

Telegram structure: sRA STlms						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Length of time parameter		Uint_16	2		0d ... +65535d (00h ... FFFFh)	00 00 ... FF FF
Time	HH HH	Uint_16	2		0d ... 99d	00 00 ... 39 39
	:	Uint_8	1		:	3A
	MM MM	Uint_16	2		0d ... 99d	00 00 ... 39 39
	:	Uint_8	1		:	3A
	SS SS	Uint_16	2		0d ... 99d	00 00 ... 39 39
Length of date parameter		Uint_16	2		0d ... +65535d (00h ... FFFFh)	00 00 ... FF FF
Date	DD DD	Uint_16	2		0d ... 99d	00 00 ... 39 39
	.	Uint_8	1		.	2E
	MM MM	Uint_16	2		0d ... 99d	00 00 ... 39 39
	.	Uint_8	1		.	2E
	YY YY YY YY	Uint_32	4		0d ... 9999d	00 00 00 00 ... 39 39 39 39
LED1		Uint_32	4	Inactive: Active: Blinking:	1 2 3	00 00 00 01 00 00 00 02 00 00 00 03
LED2		Uint_32	4	Inactive: Active: Blinking:	1 2 3	00 00 00 01 00 00 00 02 00 00 00 03
LED3		Uint_32	4	Inactive: Active: Blinking:	1 2 3	00 00 00 01 00 00 00 02 00 00 00 03

Table 140: Example: sRA STlms

CoLa A	<STX>sRA{SPC}STlms{SPC}7{SPC}0{SPC}8{SPC}08:52:51{SPC}A{SPC}13.01.2025{SPC}1{SPC}1{SPC}2<ETX>
CoLa B	02 02 02 02 00 00 00 2F 73 52 41 20 53 54 6C 6D 73 20 00 07 00 00 08 30 38 3A 35 32 3A 35 31 00 0A 31 33 2E 30 31 2E 32 30 32 35 00 00 00 01 00 00 00 01 00 00 00 02 1F

12.2.1.4.4.4 Set NTP (Network Time Protocol) parameters

12.2.1.4.4.4.1 Set time synchronization interface [sWN TSCTCInterface]

Define the interface for device time synchronization.

Table 141: Telegram structure: sWN TSCTCInterface

Telegram structure: sWN TSCTCInterface (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3		sWN	73 57 4E
Command	Set time synchroniza-tion interface	String	14		TSCTCInterface	54 53 43 54 43 49 6E 74 65 72 66 61 63 65
Variable data	Time synchronization interface data	Uint_8	1	Ethernet: CAN:	0 1	00 01

Table 142: Example: sWN TSCTCInterface

CoLa A	<STX>sWN{SPC}TSCTCInterface{SPC}0<ETX>
	<STX>sWN TSCTCInterface 0<ETX>
	sWN TSCTCInterface 0
	02 73 57 4E 20 54 53 43 54 43 49 6E 74 65 72 66 61 63 65 20 30 03
CoLa B	02 02 02 02 00 00 00 14 73 57 4E 20 54 53 43 54 43 49 6E 74 65 72 66 61 63 65 20 00 7C
	73 57 4E 20 54 53 43 54 43 49 6E 74 65 72 66 61 63 65 20 00

Table 143: Telegram structure: sWA TSCTCInterface

Telegram structure: sWA TSCTCInterface						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sWA	73 57 41
Command	Set time synchroniza-tion interface	String	14		TSCTCInterface	54 53 43 54 43 49 6E 74 65 72 66 61 63 65

Table 144: Example: sWA TSCTCInterface

CoLa A	<STX>sWA{SPC}TSCTCInterface<ETX>
	02 73 57 41 20 54 53 43 54 43 49 6E 74 65 72 66 61 63 65 03
CoLa B	02 02 02 02 00 00 00 13 73 57 41 20 54 53 43 54 43 49 6E 74 65 72 66 61 63 65 20 73

12.2.1.4.4.4.2 Set time server IP address [sWN TSCTCSrvAddr]

Define the IP address from which the device will receive the time synchronization.

Table 145: Telegram structure: sWN TSCTCSrvAddr

Telegram structure: sWN TSCTCSrvAddr (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3		sWN	73 57 4E
Command	Set time server IP address	String	12		TSCTCSrvAddr	54 53 43 54 43 53 72 76 41 64 64 72
IP address data	Set values	Uint_8	1	First part of IP address	0 ...+255d (00... FF)	00 ... FF
				Second part of IP address	0 ...+255d (00... FF)	00 ... FF
				Third part of IP address	0 ...+255d (00... FF)	00 ... FF
				Fourth part of IP address	0 ...+255d (00... FF)	00 ... FF

Table 146: Example: sWN TSCTCSrvAddr 192.168.0.11

CoLa A	<STX>sWN{SPC}TSCTCSrvAddr{SPC}C0{SPC}A8{SPC}00{SPC}0B<ETX>
	<STX>sWN TSCTCSrvAddr C0 A8 00 0B<ETX>
	sWN TSCTCSrvAddr C0 A8 00 0B
	02 73 57 4E 20 54 53 43 54 43 53 72 76 41 64 64 72 20 43 30 20 41 38 20 30 30 20 30 42 03
CoLa B	02 02 02 02 00 00 00 15 73 57 4E 20 54 53 43 54 43 53 72 76 41 64 64 72 20 C0 A8 00 0B 3E
	73 57 4E 20 54 53 43 54 43 53 72 76 41 64 64 72 20 C0 A8 00 0B

Table 147: Telegram structure: sWA TSCTCSrvAddr

Telegram structure: sWA TSCTCSrvAddr						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sWA	73 57 41
Command	Set time server IP address	String	12		TSCTCSrvAddr	54 53 43 54 43 53 72 76 41 64 64 72

Table 148: Example: sWA TSCTCSrvAddr

CoLa A	<STX>sWA{SPC}TSCTCSrvAddr<ETX>
	02 73 57 41 20 54 53 43 54 43 53 72 76 41 64 64 72 03
CoLa B	02 02 02 02 00 00 00 11 73 57 41 20 54 53 43 54 43 53 72 76 41 64 64 72 20 52

12.2.1.4.4.4.3 Set time zone [sWN TSCTCtimezone]

Table 149: Telegram structure: sWN TSCTCtimezone

Telegram structure: sWN TSCTCtimezone (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3		sWN	73 57 4E
Command	Set time zone	String	13		TSCTCtimezone	54 53 43 54 43 74 69 6D 65 7A 6F 6E 65
Time zone data	Set values in number of hours relative to GMT, hex specially coded	Int_8	1	[GMT-12:00] ... [GMT] ... [GMT+12:00]	+0d (00h) ... +12d (0Ch) ... +24d (18h)	00 ... 0C ... 18

Table 150: Example: sWN TSCTCtimezone GMT -11 hours

CoLa A	<STX>sWN{SPC}TSCTCtimezone{SPC}+1<ETX>	
	<STX>sWN TSCTCtimezone +1<ETX>	
	sWN TSCTCtimezone +1	
	02 73 57 4E 20 54 53 43 54 43 74 69 6D 65 7A 6F 6E 65 20 2B 31 03	
CoLa B	02 02 02 02 00 00 00 13 73 57 4E 20 54 53 43 54 43 74 69 6D 65 7A 6F 6E 65 20 0D 3F	
	73 57 4E 20 54 53 43 54 43 74 69 6D 65 7A 6F 6E 65 20 0D	

Table 151: Telegram structure: sWA TSCTCtimezone

Telegram structure: sWA TSCTCtimezone						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sWA	73 57 41
Command	Set time zone	String	13		TSCTCtimezone	54 53 43 54 43 74 69 6D 65 7A 6F 6E 65

Table 152: Example: sWA TSCTCtimezone

CoLa A	<STX>sWA{SPC}TSCTCtimezone<ETX>
	02 73 57 41 20 54 53 43 54 43 74 69 6D 65 7A 6F 6E 65 03
CoLa B	02 02 02 02 00 00 00 12 73 57 41 20 54 53 43 54 43 74 69 6D 65 7A 6F 6E 65 20 3D

12.2.1.4.4.4 Set update time [sWN TSCTCupdatetime]

Define the time period after which the sensor will attempt to get the current system time for the TSC server.

Table 153: Telegram structure: sWN TSCTCupdatetime

Telegram structure: sWN TSCTCupdatetime (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3		sWN	73 57 4E
Command	Set update time of synchronization	String	15		TSCTCupdate-time	54 53 43 54 43 75 70 64 61 74 65 74 69 6D 65
Update time of synchronization	Set values in seconds	Uint_32	4		+1d ... +3600d (01h ... 0E10h)	00 00 00 00 ... 00 00 0E 10

Table 154: Example: sWN TSCTCupdatetime 600 s

CoLa A	<STX>sWN{SPC}TSCTCupdatetime{SPC}+600<ETX>	
	<STX>sWN TSCTCupdatetime +600<ETX>	
	sWN TSCTCupdatetime +600	
	02 73 57 4E 20 54 53 43 54 43 75 70 64 61 74 65 74 69 6D 65 20 2B 36 30 30 03	
CoLa B	02 02 02 02 00 00 00 18 73 57 4E 20 54 53 43 54 43 75 70 64 61 74 65 74 69 6D 65 20 00 00 02 58 67	
	73 57 4E 20 54 53 43 54 43 75 70 64 61 74 65 74 69 6D 65 20 00 00 02 58	

Table 155: Telegram structure: sWA TSCTCupdatetime

Telegram structure: sWA TSCTCupdatetime						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sWA	73 57 41
Command	Set update time of synchronization	String	15		TSCTCupdate-time	54 53 43 54 43 75 70 64 61 74 65 74 69 6D 65

Table 156: Example: sWA TSCTCupdatetime

CoLa A	<STX>sWA{SPC}TSCTCupdatetime<ETX>
	02 73 57 41 20 54 53 43 54 43 75 70 64 61 74 65 74 69 6D 65 03
CoLa B	02 02 02 02 00 00 00 14 73 57 41 20 54 53 43 54 43 75 70 64 61 74 65 74 69 6D 65 20 32

12.2.1.4.4.5 Read for maximum offset time [sRN TSCTCmaxoffset]

Estimate the offset of the time synchronization.

Table 157: Telegram structure: sRN TSCTCmaxoffset

Telegram structure: sRN TSCTCmaxoffset						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E
Command	Read maximum offset time	String	14		TSCTCmaxoff-set	54 53 43 54 43 6D 61 78 6F 66 66 73 65 74

Table 158: Example: sRN TSCTCmaxoffset

CoLa A	<STX>sRN{SPC}TSCTCmaxoffset<ETX>	
	<STX>sRN TSCTCmaxoffset<ETX>	
	sRN TSCTCmaxoffset	
	02 73 52 4E 20 54 53 43 54 43 6D 61 78 6F 66 66 73 65 74 03	
CoLa B	02 02 02 02 00 00 00 12 73 52 4E 20 54 53 43 54 43 6D 61 78 6F 66 66 73 65 74 65	
	73 52 4E 20 54 53 43 54 43 6D 61 78 6F 66 66 73 65 74	

Table 159: Telegram structure: sRA TSCTCmaxoffset

Telegram structure: sRA TSCTCmaxoffset						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sRA	73 52 41
Command	Read maximum offset time	String	14		TSCTCmaxoff-set	54 53 43 54 43 6D 61 78 6F 66 66 73 65 74
Max. offset time	[Seconds as float according to IEEE754]	Real	4	Min Value: ~ -3.403*10^38 s Max Value: ~+3.403*10^38 s	0h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF

Table 160: Example: sRA TSCTCmaxoffset 18000 s

CoLa A	<STX>sRA{SPC}TSCTCmaxoffset{SPC}468CA000<ETX>
	02 73 52 41 20 54 53 43 54 43 6D 61 78 6F 66 66 73 65 74 20 34 36 38 43 41 30 30 30 03
CoLa B	02 02 02 02 00 00 00 17 73 52 41 20 54 53 43 54 43 6D 61 78 6F 66 66 73 65 74 20 46 8C A0 00 20

12.2.1.4.4.6 Read for delay time [sRN TSCTCdelay]

Read the network delay for the time synchronization.

Table 161: Telegram structure: sRN TSCTCdelay

Telegram structure: sRN TSCTCdelay						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E
Command	Read delay time	String	10		TSCTCdelay	54 53 43 54 43 64 65 6C 61 79

Table 162: Example: sRN TSCTCdelay

CoLa A	<STX>sRN{SPC}TSCTCdelay<ETX>
	<STX>sRN TSCTCdelay<ETX>
	sRN TSCTCdelay
	02 73 52 4E 20 54 53 43 54 43 64 65 6C 61 79 03
CoLa B	02 02 02 02 00 00 00 0E 73 52 4E 20 54 53 43 54 43 64 65 6C 61 79 69
	73 52 4E 20 54 53 43 54 43 64 65 6C 61 79

Table 163: Telegram structure: sRA TSCTCdelay

Telegram structure: sRA TSCTCdelay						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sRA	73 52 41
Command	Read delay time	String	10		TSCTCdelay	54 53 43 54 43 64 65 6C 61 79
Max. offset time	[Seconds as float according to IEEE754]	Real	4		0h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF

Table 164: Example: sRA TSCTCdelay 0.003 s

CoLa A	<STX>sRA{SPC}TSCTCdelay{SPC}3B435B02<ETX>
	02 73 52 41 20 54 53 43 54 43 64 65 6C 61 79 20 33 42 34 33 35 42 30 32 03
CoLa B	02 02 02 02 00 00 00 13 73 52 41 20 54 53 43 54 43 64 65 6C 61 79 20 3B 43 5B 02 67

12.2.1.4.4.7 Reset maximum offset time [sMN mResetMaxOff]

This command resets the maximum offset time, i.e. sets it to zero (0).

Table 165: Telegram structure: sMN mResetMaxOff

Telegram structure: sMN mResetMaxOff (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3		sMN	73 4D 4E
Command	Reset maximum offset time	String	12		mResetMaxOff	6D 52 65 73 65 74 4D 61 78 4F 66 66

Table 166: Example: sMN mResetMaxOff

CoLa A	<STX>sMN{SPC}mResetMaxOff<ETX>
	<STX>sMN mResetMaxOff<ETX>
	sMN mResetMaxOff
	02 73 4D 4E 20 6D 52 65 73 65 74 4D 61 78 4F 66 66 03
CoLa B	02 02 02 02 00 00 00 10 73 4D 4E 20 6D 52 65 73 65 74 4D 61 78 4F 66 66 73
	73 4D 4E 20 6D 52 65 73 65 74 4D 61 78 4F 66 66

Table 167: Telegram structure: sAN mResetMaxOff

Telegram structure: sAN mResetMaxOff						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sAN	73 41 4E
Command	Reset maximum offset time	String	12		mResetMaxOff	6D 52 65 73 65 74 4D 61 78 4F 66 66

Table 168: Example: sAN mResetMaxOff

CoLa A	<STX>sAN{SPC}mResetMaxOff<ETX>
	02 73 41 4E 20 6D 52 65 73 65 74 4D 61 78 4F 66 66 03
CoLa B	02 02 02 02 00 00 00 11 73 41 4E 20 6D 52 65 73 65 74 4D 61 78 4F 66 66 20 5F

12.2.1.4.5 Filters

12.2.1.4.5.1 Set particle filter [sWN LFPparticle]

Filter out disturbances in the meausurement data caused by particles such as dust, snow flakes or similar see "Particle filter", page 33.

Table 169: Telegram structure: sWN LFPparticle

Telegram structure: sWN LFPparticle (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3		sWN	73 57 4E
Command	Set particle filter	String	11		LFPparticle	4C 46 50 70 61 72 74 69 63 6C 65
Status code	Code number	Bool_1	1	Inactive: Active:	0 1	00 01
Threshold[1)]	Particle threshold in mm	Uint_16	2	(must be taken)	+500d (1F4h)	01 F4

1) Never change the threshold here, it is taken by the device to handle the particles.

Table 170: Example: sWN LFPparticle

CoLa A	<STX>sWN{SPC}LFPparticle{SPC}1{SPC}+500<ETX>	
	<STX>sWN LFPparticle 1 +500<ETX>	
	sWN LFPparticle 1 +500	
	02 73 57 4E 20 4C 46 50 70 61 72 74 69 63 6C 65 20 31 20 2B 35 30 30 03	
CoLa B	02 02 02 02 00 00 00 13 73 57 4E 20 4C 46 50 70 61 72 74 69 63 6C 65 20 01 01 F4 D0	
	73 57 4E 20 4C 46 50 70 61 72 74 69 63 6C 65 20 01 01 F4	

1) 1)

Table 171: Telegram structure: sWA LFPparticle

Telegram structure: sWA LFPparticle						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sWA	73 57 41
Command	Set particle filter	String	11		LFPparticle	4C 46 50 70 61 72 74 69 63 6C 65

Table 172: Example: sWA LFPparticle

CoLa A	<STX>sWA{SPC}LFPparticle<ETX>
	02 73 57 41 20 4C 46 50 70 61 72 74 69 63 6C 65 03
CoLa B	02 02 02 02 00 00 00 10 73 57 41 20 4C 46 50 70 61 72 74 69 63 6C 65 20 2B

12.2.1.4.5.2 Set average filter [sWN LFPmeanfilter]

This filter smooths the distance value by calculating the arithmetic average from the configured number of scans see "Average filter", page 34.

Table 173: Telegram structure: sWN LFPmeanfilter

Telegram structure: sWN LFPmeanfilter (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3		sWN	73 57 4E
Command	Set average filter	String	13		LFPmeanfilter	4C 46 50 6D 65 61 6E 66 69 6C 74 65 72
Status code	Code number	Bool_1	1	Inactive: Active:	0 1	00 01
Number of scans	Number	Uint_16	2		+2d ... +100d (00 02h ... 00 64h)	00 02 ... 00 64
Reserved	-	Enum_8	1	Always:	0	00

Table 174: Example: sWN LFPmeanfilter

CoLa A	<STX>sWN{SPC}LFPmeanfilter{SPC}1{SPC}+10{SPC}0<ETX>
	<STX>sWN LFPmeanfilter 1 +10 0<ETX>
	sWN LFPmeanfilter 1 +10 0
	02 73 57 4E 20 4C 46 50 6D 65 61 6E 66 69 6C 74 65 72 20 31 20 2B 31 30 20 30 03
CoLa B	02 02 02 02 00 00 00 16 73 57 4E 20 4C 46 50 6D 65 61 6E 66 69 6C 74 65 72 20 01 00 0A 00 3C
	73 57 4E 20 4C 46 50 6D 65 61 6E 66 69 6C 74 65 72 20 01 00 0A 00

Table 175: Telegram structure: sWA LFPmeanfilter

Telegram structure: sWA LFPmeanfilter						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sWA	73 57 41

Telegram structure: sWA LFPmeanfilter						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command	Set average filter	String	13		LFPmeanfilter	4C 46 50 6D 65 61 6E 66 69 6C 74 65 72

Table 176: Example: sWA LFPmeanfilter

CoLa A	<STX>sWA{SPC}LFPmeanfilter<ETX>
	02 73 57 41 20 4C 46 50 6D 65 61 6E 66 69 6C 74 65 72 03
CoLa B	02 02 02 02 00 00 00 12 73 57 41 20 4C 46 50 6D 65 61 6E 66 69 6C 74 65 72 20 38

12.2.1.4.5.3 Set n-pulse to 1-pulse filter (Echo filter) [sWN LFPnto1filter]

Echo filter for LMS1xx see "Echo filter", page 33.

Table 177: Telegram structure: sWN LFPnto1filter

Telegram structure: sWN LFPnto1filter (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3		sWN	73 57 4E
Command	Set n-to-1 filter	String	13		LFPnto1filter	4C 46 50 6E 74 6F 31 66 69 6C 74 65 72
Status code	Code number	Bool_1	1	Inactive: Active:	0 1	00 01

Table 178: Example: sWN LFPnto1filter

CoLa A	<STX>sWN{SPC}LFPnto1filter{SPC}1<ETX>	
	<STX>sWN LFPnto1filter 1<ETX>	
	sWN LFPnto1filter 1	
	02 73 57 4E 20 4C 46 50 6E 74 6F 31 66 69 6C 74 65 72 20 31 03	
CoLa B	02 02 02 02 00 00 00 13 73 57 4E 20 4C 46 50 6E 74 6F 31 66 69 6C 74 65 72 20 01 75	
	73 57 4E 20 4C 46 50 6E 74 6F 31 66 69 6C 74 65 72 20 01	

Table 179: Telegram structure: sWA LFPnto1filter

Telegram structure: sWA LFPnto1filter						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sWA	73 57 41
Command	Set n-to-1 filter	String	13		LFPnto1filter	4C 46 50 6E 74 6F 31 66 69 6C 74 65 72

Table 180: Example: sWA LFPnto1filter

CoLa A	<STX>sWA{SPC}LFPnto1filter<ETX>
	02 73 57 41 20 4C 46 50 6E 74 6F 31 66 69 6C 74 65 72 03
CoLa B	02 02 02 02 00 00 00 12 73 57 41 20 4C 46 50 6E 74 6F 31 66 69 6C 74 65 72 20 7B

12.2.1.4.5.4 Set fog filter [sWN MSsuppmode]

Activating this filter allows the device to eliminate unwanted echos at close range see "Fog filter", page 31.

Table 181: Telegram structure: sWN MSsuppmode

Telegram structure: sWN MSsuppmode (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3		sWN	73 57 4E
Command	Set fog filter	String	10		MSsuppmode	4D 53 73 75 70 70 6D 6F 64 65
Status code	Code number	Bool_1	1	Glitch: Fog:	0 1	00 01

Table 182: Example: sWN MSsuppmode

CoLa A	<STX>sWN{SPC}MSsuppmode{SPC}1<ETX>
	<STX>sWN MSsuppmode 1<ETX>
	sWN MSsuppmode 1
	02 73 57 4E 20 4D 53 73 75 70 70 6D 6F 64 65 20 31 03
CoLa B	02 02 02 02 00 00 00 10 73 57 4E 20 4D 53 73 75 70 70 6D 6F 64 65 20 01 70
	73 57 4E 20 4D 53 73 75 70 70 6D 6F 64 65 20 01

Table 183: Telegram structure: sWA MSsuppmode

Telegram structure: sWA MSsuppmode						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sWA	73 57 41
Command	Set fog filter	String	10		MSsuppmode	4D 53 73 75 70 70 6D 6F 64 65

Table 184: Example: sWA MSsuppmode

CoLa A	<STX>sWA{SPC}MSsuppmode<ETX>
	02 73 57 41 20 4D 53 73 75 70 70 6D 6F 64 65 03
CoLa B	02 02 02 02 00 00 00 0F 73 57 41 20 4D 53 73 75 70 70 6D 6F 64 65 20 7E

12.2.1.4.6 Encoder

12.2.1.4.6.1 Set increment source [sWN LICsrc]

Define if the device gets movement information via the encoder input or if a fixed speed is to be assumed.

Table 185: Telegram structure: sWN LICsrc

Telegram structure: sWN LICsrc (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3		sWN	73 57 4E
Command	Set increment source	String	6		LICsrc	4C 49 43 73 72 63

Telegram structure: sWN LICsrc (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Increment source		Enum_8	1	Fixed speed: Encoder:	0 1	00 01

Table 186: Example: sWN LICsrc

CoLa A	<STX>sWN{SPC}LICsrc{SPC}0<ETX>
	<STX>sWN LICsrc 0<ETX>
	sWN LICsrc 0
	02 73 57 4E 20 4C 49 43 73 72 63 20 30 03
CoLa B	02 02 02 02 00 00 00 0C 73 57 4E 20 4C 49 43 73 72 63 20 00 4E
	73 57 4E 20 4C 49 43 73 72 63 20 00

Table 187: Telegram structure: sWA LICsrc

Telegram structure: sWA LICsrc						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sWA	73 57 41
Command	Set increment source	String	6		LICsrc	4C 49 43 73 72 63

Table 188: Example: sWA LICsrc

CoLa A	<STX>sWA{SPC}LICsrc<ETX>
	02 73 57 41 20 4C 49 43 73 72 63 03
CoLa B	02 02 02 02 00 00 00 0B 73 57 41 20 4C 49 43 73 72 63 20 41

12.2.1.4.6.2 Set encoder settings [sWN LICencset]

Define if and which type of encoder is connected to the device so that input signals are processed correctly.

Table 189: Telegram structure: sWN LICencset

Telegram structure: sWN LICencset (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3		sWN	73 57 4E
Command	Encoder settings	String	9		LICencset	4C 49 43 65 6E 63 73 65 74
Encoder setting		Enum_8	1	Off: Single increment/INC1: Direction recognition (phase): Direction recognition (level):	0 1 2 3	00 01 02 03

Table 190: Example: sWN LICencset

CoLa A	<STX>sWN{SPC}LICencset{SPC}0<ETX>
	<STX>sWN LICencset 0<ETX>
	sWN LICencset 0
	02 73 57 4E 20 4C 49 43 65 6E 63 73 65 74 20 30 03

CoLa B	02 02 02 02 00 00 00 0F 73 57 4E 20 4C 49 43 65 6E 63 73 65 74 20 00 26 73 57 4E 20 4C 49 43 65 6E 63 73 65 74 20 00

Table 191: Telegram structure: sWA LICencset

Telegram structure: sWA LICencset						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sWA	73 57 41
Command	Encoder settings	String	9		LICencset	4C 49 43 65 6E 63 73 65 74

Table 192: Example: sWA LICencset

CoLa A	<STX>sWA{SPC}LICencset<ETX>
	02 73 57 41 20 4C 49 43 65 6E 63 73 65 74 03
CoLa B	02 02 02 02 00 00 00 0E 73 57 41 20 4C 49 43 65 6E 63 73 65 74 20 29

12.2.1.4.6.3 Set encoder resolution [sWN LICencres]

Determine the encoder resolution in mm/increment.

Table 193: Telegram structure: sWN LICencres

Telegram structure: sWN LICencres (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3		sWN	73 57 4E
Command	Set encoder resolution	String	9		LICencres	4C 49 43 65 6E 63 72 65 73
Encoder resolution	Resolution value in mm/Inc as float according to IEEE754	Real	4		+0.001d ... +2000d	3A 83 12 6F ... 44 FA 00 00 (see IEEE 754)

Table 194: Example: sWN LICencres

CoLa A	<STX>sWN{SPC}LICencres{SPC}+1000<ETX>
	<STX>sWN LICencres +1000<ETX>
	sWN LICencres +1000
	02 73 57 4E 20 4C 49 43 65 6E 63 72 65 73 20 2B 31 30 30 30 03
CoLa B	02 02 02 02 00 00 00 12 73 57 4E 20 4C 49 43 65 6E 63 72 65 73 20 44 7A 00 00 1E
	73 57 4E 20 4C 49 43 65 6E 63 72 65 73 20 44 7A 00 00

Table 195: Telegram structure: sWA LICencres

Telegram structure: sWA LICencres						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sWA	73 57 41
Command	Set encoder resolution	String	9		LICencres	4C 49 43 65 6E 63 72 65 73

Table 196: Example: sWA LICencres

CoLa A	<STX>sWA{SPC}LICencres<ETX>
	02 73 57 41 20 4C 49 43 65 6E 63 72 65 73 03
CoLa B	02 02 02 02 00 00 00 0E 73 57 41 20 4C 49 43 65 6E 63 72 65 73 20 2F

12.2.1.4.6.4 Set fixed speed [sWN LICFixVel]

Simulate fixed encoder speed in m/s.

Table 197: Telegram structure: sWN LICFixVel

Telegram structure: sWN LICFixVel (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3		sWN	73 57 4E
Command	Set fixed speed	String	9		LICFixVel	4C 49 43 46 69 78 56 65 6C
Fixed speed	Speed in m/s as float according to IEEE754	Real	4		+0.001d ... +10.0d	3A 83 12 6F ... 41 20 00 00

Table 198: Example: sWN LICFixVel

CoLa A	<STX>sWN{SPC}LICFixVel{SPC}+5<ETX>	
	<STX>sWN LICFixVel +5<ETX>	
	sWN LICFixVel +5	
	02 73 57 4E 20 4C 49 43 46 69 78 56 65 6C 20 2B 35 03	
CoLa B	02 02 02 02 00 00 00 12 73 57 4E 20 4C 49 43 46 69 78 56 65 6C 20 40 A0 00 00 C4	
	73 57 4E 20 4C 49 43 46 69 78 56 65 6C 20 40 A0 00 00	

Table 199: Telegram structure: sWA LICFixVel

Telegram structure: sWA LICFixVel						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sWA	73 57 41
Command	Set fixed speed	String	9		LICFixVel	4C 49 43 46 69 78 56 65 6C

Table 200: Example: sWA LICFixVel

CoLa A	<STX>sWA{SPC}LICFixVel<ETX>
	02 73 57 41 20 4C 49 43 46 69 78 56 65 6C 03
CoLa B	02 02 02 02 00 00 00 0E 73 57 41 20 4C 49 43 46 69 78 56 65 6C 20 2B

12.2.1.4.6.5 Read speed threshold [sRN LICSpTh]

Table 201: Telegram structure: sRN LICSpTh

Telegram structure: sRN LICSpTh						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E

Telegram structure: sRN LICSpTh						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command	Read speed threshold	String	7		LICSpTh	4C 49 43 53 70 54 68

Table 202: Example: sRN LICSpTh

CoLa A	<STX>sRN{SPC}LICSpTh<ETX>
	<STX>sRN LICSpTh<ETX>
	sRN LICSpTh
	02 73 52 4E 20 4C 49 43 53 70 54 68 03
CoLa B	02 02 02 02 00 00 00 0B 73 52 4E 20 4C 49 43 53 70 54 68 16
	73 52 4E 20 4C 49 43 53 70 54 68

Table 203: Telegram structure: sRA LICSpTh

Telegram structure: sRA LICSpTh						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sRA	73 52 41
Command	Read speed threshold	String	7		LICSpTh	4C 49 43 53 70 54 68
Speed threshold	Speed threshold in %	Uint_8	2		+1d ... +20d (01h ... 14h)	01 ... 14

Table 204: Example: sRA LICSpTh

CoLa A	<STX>sRA{SPC}LICSpTh{SPC}5<ETX>
	02 73 52 41 20 4C 49 43 53 70 54 68 20 35 03
CoLa B	02 02 02 02 00 00 00 0D 73 52 41 20 4C 49 43 53 70 54 68 20 05 3C

12.2.1.4.6.6 Read encoder speed [sRN LICencsp]

Read encoder speed in m/s.

Table 205: Telegram structure: sRN LICencsp

Telegram structure: sRN LICencsp						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E
Command	Read encoder speed	String	8		LICencsp	4C 49 43 65 6E 63 73 70

Table 206: Example: sRN LICencsp

CoLa A	<STX>sRN{SPC}LICencsp<ETX>
	<STX>sRN LICencsp<ETX>
	sRN LICencsp
CoLa B	02 02 02 02 00 00 00 0C 73 52 4E 20 4C 49 43 65 6E 63 73 70 62
	73 52 4E 20 4C 49 43 65 6E 63 73 70

Table 207: Telegram structure: sRA LICencsp

Telegram structure: sRA LICencsp						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sRA	73 52 41
Command	Read encoder speed	String	8		LICencsp	4C 49 43 65 6E 63 73 70
Encoder speed	[Speed in m/s as float according to IEEE754]	Real	4		0h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF

Table 208: Example: sRA LICencsp

CoLa A	<STX>sRA{SPC}LICencsp{SPC}0<ETX>
CoLa B	02 02 02 02 00 00 00 11 73 52 41 20 4C 49 43 65 6E 63 73 70 20 00 00 00 00 4D

12.2.1.4.7 Inputs and Outputs

12.2.1.4.7.1 Read state of the ports [sRN LIDportstate]

LIDportstate has to be available additionally or as successor of the LIDoutputstate telegram.

Valid for all sensors with Ethernet and ports (inputs / outputs).

Table 209: Telegram structure: sRN LIDportstate

Telegram structure: sRN LIDportstate						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E
Command	Ask for port configuration	String	12		LIDportstate	4C 49 44 70 6F 72 74 73 74 61 74 65

Table 210: Example: sRN LIDportstate

CoLa A	<STX>sRN{SPC}LIDportstate<ETX>
	<STX>sRN LIDportstate<ETX>
	sRN LIDportstate
	02 73 52 4E 20 4C 49 44 70 6F 72 74 73 74 61 74 65 03
CoLa B	02 02 02 02 00 00 00 10 73 52 4E 20 4C 49 44 70 6F 72 74 73 74 61 74 65 60
	73 52 4E 20 4C 49 44 70 6F 72 74 73 74 61 74 65

Table 211: Telegram structure: sRA LIDportstate

Telegram structure: sRA LIDportstate						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sRA	73 52 41
Command	Port state	String	12		LIDportstate	4C 49 44 70 6F 72 74 73 74 61 74 65

Telegram structure: sRA LIDportstate						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Status code	Version number	Uint_16	2	Current version:	0 ... FFFFh 1	00 01 ... FF FF
	System counter (time in µs since power up max. 71min then starting from 0 again)	Uint_32	4		0 ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
ARRAY which defines the number of internal ports*	0...n	Uint_16	2	Hex: Not available: Number of ports:	0000 - FFFF 00 01 ... n	00 00 - FF FF
State of the ports and count value in hex	Internal port state	Enum_8	1	Output voltage low: (Relays open) Output voltage high: (Relays closed) Tri-state: Input voltage high (level): Input voltage from low to high (edge) Input voltage low (level): Input voltage high to low (edge)	00 01 02 03 04	00 01 02 03 04
	Internal port counter	Uint_32	4		0 ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
	.....					
ARRAY which defines the number of external ports*	0...n	Uint_16	1	Hex: Not available: Numer of ports:	00 00 - FF FF 00 01 ... n	00 00 - FF FF
State of the ports and count value in hex	External port state	Enum_8	1	Output voltage low: (Relays open) Output voltage high: (Relays closed) Tri-state: Input voltage high (level): Input voltage from low to high (edge) Input voltage low (level): Input voltage high to low (edge)	00 01 02 03 04	00 01 02 03 04
	External port counter	Uint_32	4		0 ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
Time	States code	Enum_16	1	No time data: Time data:	00 00 00 01	00 00 00 01
Time Block (sensor time from the last change of min. one of the out-puts)	Year	Array	2	E.g.	1970	07 B2
	Month		1		1 ... 12	01 ... 0C
	Day		1		1 ... 31	01 ... 1F
	Hour		1		0 ... 23	00 ... 17
	Minute		1		0 ... 59	00 ... 3B
	Second		1		0 ... 59	00 ... 3B
	Microsecond		4		0 ... 999999	00 00 00 00 ... 00 0F 42 3F

Inputs/outputs: If the device has separate inputs and outputs (instead of general purpose ports) the ARRAY shall start with inputs followed by the outputs.

Example with 3 internal and 4 external ports:

Table 212: Example: sRA LIDportstate

CoLa A	<STX>sRA{SPC}LIDportstate{SPC}1{SPC}41F84EC5{SPC}3{SPC}1{SPC}20{SPC}1{SPC}20{SPC}3{SPC}20{SPC}4{SPC}1{SPC}20{SPC}1{SPC}20{SPC}0{SPC}20{SPC}03{SPC}20{SPC}1{SPC}7D9{SPC}2{SPC}12{SPC}C{SPC}29{SPC}E{SPC}975E0<ETX>
	02 73 52 41 20 4C 49 44 70 6F 72 74 73 74 61 74 65 20 31 20 34 31 46 38 34 45 43 35 20 33 20 31 20 32 30 20 31 20 32 30 20 33 20 32 30 20 34 20 31 20 32 30 20 31 20 32 30 20 30 20 32 30 20 30 33 20 32 30 20 31 20 37 44 39 20 32 20 31 32 20 43 20 32 39 20 45 20 39 37 35 45 30 03
CoLa B	02 02 02 02 00 00 00 5B 73 52 41 20 4C 49 44 70 6F 72 74 73 74 61 74 65 20 01 00 41 F8 4E C5 00 03 00 01 00 20 00 04 00 01 00 20 00 01 00 20 00 00 00 20 00 03 00 20 00 01 00 07 D9 00 02 00 12 00 0C 00 29 00 0E 00 09 75 E0 21

12.2.1.4.7.2 Read state of the inputs [sRN LIDinputstate]

Use `sEN LIDinputstate 1` to receive a telegram each time an input signal (e.g. by trigger) changes. Compare with chapter "Receive outputstate by event [sEN LIDoutputstate]", page 148.

Table 213: Telegram structure: sRN LIDinputstate

Telegram structure: sRN LIDinputstate						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E
Command	Input state	String	14		LIDinputstate	4C 49 44 69 6E 70 75 74 73 74 61 74 65

Table 214: Example: sRN LIDinputstate

CoLa A	<STX>sRN{SPC}LIDinputstate<ETX>	
	<STX>sRN LIDinputstate<ETX>	
	sRN LIDinputstate	
	02 73 52 4E 20 4C 49 44 69 6E 70 75 74 73 74 61 74 65 03	
CoLa B	02 02 02 02 00 00 00 11 73 52 4E 20 4C 49 44 69 6E 70 75 74 73 74 61 74 65 0F	
	73 52 4E 20 4C 49 44 69 6E 70 75 74 73 74 61 74 65	

Table 215: Telegram structure: sRA LIDinputstate

Telegram structure: sRA LIDinputstate						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sRA	73 52 41
Command	Output state	String	14		LIDinputstate	4C 49 44 69 6E 70 75 74 73 74 61 74 65
Status code	Version number	Uint_16	2		0h ... FFFFh	00 00 ... FF FF
	System counter (time in µs since power up max. 71min then starting from 0 again)	Uint_32	4		0 h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
State of the inputs 1 ... n	Amount of inputs (n) depending of device family	Enum_8	1	Not active: Active: Input not used:	0 1 2	00 01 02

Telegram structure: sRA LIDinputstate						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Time	States code	Uint_16	2	No time data: Time data:	0 1	00 00 00 01
Time Block (sensor-time from the last change of min. one of the out-puts)	Year	Array	2	E. g.	1970	07 B2
	Month		1		1 ... 12	01 ... 0C
	Day		1		1 ... 31	01 ... 1F
	Hour		1		0 ... 23	00 ... 17
	Minute		1		0 ... 59	00 ... 3B
	Second		1		0 ... 59	00 ... 3B
	Microsecond		4		0 ... 999999	00 00 00 00 ... 00 0F 42 3F

Table 216: Example: sRA LIDinputstate In1 active, In2 active, In3 inactive, In4 not used, In5 not used, In6 inactive, In7 inactive, In8 not used, time: 2022-04-25 14:10 46.946 sec

CoLa A	<STX>sRA{SPC}LIDinput-state{SPC}1{SPC}98E71BD7{SPC}1{SPC}1{SPC}0{SPC}2{SPC}2{SPC}0{SPC}0{SPC}2{SPC}1{SPC}7E6{SPC}4{SPC}19{SPC}E{SPC}A{SPC}2E{SPC}E6F50<ETX>
	02 73 52 41 20 4C 49 44 69 6E 70 75 74 73 74 61 74 65 20 31 20 39 38 45 37 31 42 44 37 20 31 20 31 20 30 20 32 20 32 20 30 20 30 20 32 20 31 20 37 45 36 20 34 20 31 39 20 45 20 41 20 32 45 20 45 36 46 35 30 03
CoLa B	02 02 02 02 00 00 00 2D 73 52 41 20 4C 49 44 69 6E 70 75 74 73 74 61 74 65 20 00 01 98 E7 1B D7 01 01 00 02 02 00 00 02 00 01 07 E6 04 19 0E 0A 2E 00 0E 6F 50 76

12.2.1.4.7.3 Read state of the outputs [sRN LIDoutputstate]

Status of all outputs

Table 217: Telegram structure: sRN LIDoutputstate

Telegram structure: sRN LIDoutputstate						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E
Command	Output state	String	14		LIDoutputstate	4C 49 44 6F 75 74 70 75 74 73 74 61 74 65

Table 218: Example: sRN LIDoutputstate

CoLa A	<STX>sRN{SPC}LIDoutputstate<ETX>
	<STX>sRN LIDoutputstate<ETX>
	sRN LIDoutputstate
	02 73 52 4E 20 4C 49 44 6F 75 74 70 75 74 73 74 61 74 65 03
CoLa B	02 02 02 02 00 00 00 12 73 52 4E 20 4C 49 44 6F 75 74 70 75 74 73 74 61 74 65 66
	73 52 4E 20 4C 49 44 6F 75 74 70 75 74 73 74 61 74 65

Table 219: Telegram structure: sRA LIDoutputstate

Telegram structure: sRA LIDoutputstate						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Complete telegram structure of the answer see "Receive outputstate by event [sEN LIDoutputstate]", page 148.						

12.2.1.4.7.4 Receive outputstate by event [sEN LIDoutputstate]

Output telegram is sent every time an output state changes.

Table 220: Telegram structure: sEN LIDoutputstate

Telegram structure: sEN LIDoutputstate						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Event	String	3		sEN	73 45 4E
Command	Output state	String	14		LIDoutputstate	4C 49 44 6F 75 74 70 75 74 73 74 61 74 65
	Start/stop	Enum_8	1	Start: Stop:	1 0	01 00

Table 221: Example: sEN LIDoutputstate

CoLa A	<STX>sEN{SPC}LIDoutputstate{SPC}1<ETX>	
	<STX>sEN LIDoutputstate 1<ETX>	
	sEN LIDoutputstate 1	
	02 73 45 4E 20 4C 49 44 6F 75 74 70 75 74 73 74 61 74 65 20 31 03	
CoLa B	02 02 02 02 00 00 00 14 73 45 4E 20 4C 49 44 6F 75 74 70 75 74 73 74 61 74 65 20 01 50	
	73 45 4E 20 4C 49 44 6F 75 74 70 75 74 73 74 61 74 65 20 01	

Table 222: Telegram structure: sRA/sSN LIDoutputstate

Telegram structure: sRA/sSN LIDoutputstate						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sRA / sSN	73 52 41 / 73 53 4E
Command	Output state	String	14		LIDoutputstate	4C 49 44 6F 75 74 70 75 74 73 74 61 74 65
Status code	Version number	Uint_16	2		0h ... FFFFh	00 00 ... FF FF
	System counter (time in µs since power up max. 71min then starting from 0 again)	Uint_32	4		0h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
State of the outputs 1 ... n and count value in hex. (values of an example) Amount of outputs (n) depending of device family	Output 1 ... n state	Enum_8	1	Not active: Active: Output not used:	0 1 2	00 01 02
	Output 1 ... n count	Uint_32	4		0h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF

Telegram structure: sRA/sSN LIDoutputstate						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
State of the external outputs 1 ... n and count value in hex. (values of an example) Amount of outputs (n) depending of device family	External Output 1 ... n state	Enum_8	1	Not active: Active: Output not used:	0 1 2	00 01 02
	External Output 1 ... n count	Uint_32	4		0 ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
Time	States code	Uint_16	2	No time data: Time data:	0 1	00 00 00 01
Time Block (sensor-time from the last change of min. one of the outputs)	Year	Array	2	E. g.	1970	07 B2
	Month		1		1 ... 12	01 ... 0C
	Day		1		1 ... 31	01 ... 1F
	Hour		1		0 ... 23	00 ... 17
	Minute		1		0 ... 59	00 ... 3B
	Second		1		0 ... 59	00 ... 3B
	Microsecond		4		0 ... 999999	00 00 00 00 ... 00 0F 42 3F

Table 223: Example: sRA LIDoutputstate Out1 active Count 5, Out2 active Count 5, Out3 active Count 5, all other Outputs not used, time: 2009-02-18 12:41 14.62 sec

CoLa A	<STX>sRA{SPC}LIDoutput-state{SPC}1{SPC}41F84EC5{SPC}1{SPC}5{SPC}1{SPC}5{SPC}1{SPC}5{SPC}2{SPC}0{SPC}2{SPC}0{SPC}2{SPC}0{SPC}2{SPC}0{SPC}2{SPC}0{SPC}2{SPC}0{SPC}2{SPC}0{SPC}2{SPC}0{SPC}2{SPC}0{SPC}1{SPC}7D9{SPC}2{SPC}12{SPC}C{SPC}29{SPC}E{SPC}975E0<ETX>
	02 73 52 41 20 4C 49 44 6F 75 74 70 75 74 73 74 61 74 65 20 31 20 41 F8 4E C5 20 31 20 35 20 31 20 35 20 31 20 35 20 32 20 30 20 32 20 30 20 32 20 30 20 32 20 30 20 32 20 30 20 32 20 30 20 32 20 30 20 32 20 30 20 31 20 37 44 39 20 32 20 31 32 20 43 20 32 39 20 45 20 39 37 35 45 30 03
CoLa B	02 02 02 02 00 00 00 5D 73 52 41 20 4C 49 44 6F 75 74 70 75 74 73 74 61 74 65 20 00 01 41 F8 4E C5 01 00 00 00 05 01 00 00 00 05 01 00 00 00 05 02 00 00 00 00 02 00 00 00 00 02 00 00 00 00 02 00 00 00 00 02 00 00 00 00 02 00 00 00 00 02 00 00 00 00 02 00 00 00 00 01 07 D9 02 12 0C 29 0E 00 09 75 E0 06

12.2.1.4.7.5 Set output state [sMN mDOSetOutput]

Set a specific output to high or low via software command.

NOTE
Output source needs to be set to "SOPAS command" and the port configured as Output (in case of I/O).

Table 224: Telegram structure: sMN mDOSetOutput

Telegram structure: sMN mDOSetOutput						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3		sMN	73 4D 4E

Telegram structure: sMN mDOSetOutput						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command	Set output state	String	12		mDOSetOutput	6D 44 4F 53 65 74 4F 75 74 70 75 74
Output number		Uint_8	1		1 ... 3	01 ... 03
Output state		Enum_8	1	Not active: Active:	0 1	00 01

Table 225: Example: sMN mDOSetOutput

CoLa A	<STX>sMN{SPC}mDOSetOutput{SPC}1{SPC}1<ETX>
	<STX>sMN mDOSetOutput 1 1<ETX>
	sMN mDOSetOutput 1 1
	02 73 4D 4E 20 6D 44 4F 53 65 74 4F 75 74 70 75 74 20 31 20 31 03
CoLa B	02 02 02 02 00 00 00 13 73 4D 4E 20 6D 44 4F 53 65 74 4F 75 74 70 75 74 20 01 01 6B
	73 4D 4E 20 6D 44 4F 53 65 74 4F 75 74 70 75 74 20 01 01

Table 226: Telegram structure: sAN mDOSetOutput

Telegram structure: sAN mDOSetOutput						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sAN	73 41 4E
Command	Set output state	String	12		mDOSetOutput	6D 44 4F 53 65 74 4F 75 74 70 75 74
Status Code	Status code	Bool_1	1	Error: Success:	0 1	00 01

Table 227: Example: sAN mDOSetOutput

CoLa A	<STX>sAN{SPC}mDOSetOutput{SPC}1<ETX>
	02 73 41 4E 20 6D 44 4F 53 65 74 4F 75 74 70 75 74 20 31 03
CoLa B	02 02 02 02 00 00 00 12 73 41 4E 20 6D 44 4F 53 65 74 4F 75 74 70 75 74 20 01 66

12.2.1.4.7.6 Set debouncing time for input x [sWN DI3DebTim]

The telegram applies for the inputs 1 to 4 (DIxDebTim, x = 1 ... 4). The following tables show the data for input 3.

Table 228: Telegram structure: sWN DI3DebTim

Telegram structure: sWN DI3DebTim (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3		sWN	73 57 4E
Command	Set debouncing time for input 3	String	9		DI3DebTim	44 49 33 44 65 62 54 69 6D
Debouncing time data	[ms]	Uint_16	2		0d ... +10000d (00h ... 2710h)	00 00 ... 27 10

Table 229: Example: sWN DI3DebTim

CoLa A	<STX>sWN{SPC}DI3DebTim{SPC}+10<ETX>
	<STX>sWN DI3DebTim +10<ETX>
	sWN DI3DebTim +10
	02 73 57 4E 20 44 49 33 44 65 62 54 69 6D 20 2B 31 30 03
CoLa B	02 02 02 02 00 00 00 10 73 57 4E 20 44 49 33 44 65 62 54 69 6D 20 00 0A 77
	73 57 4E 20 44 49 33 44 65 62 54 69 6D 20 00 0A

Table 230: Telegram structure: sWA DI3DebTim

Telegram structure: sWA DI3DebTim						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sWA	73 57 41
Command	Set debouncing time for input 3	String	9		DI3DebTim	44 49 33 44 65 62 54 69 6D

Table 231: Example: sWA DI3DebTim

CoLa A	<STX>sWA{SPC}DI3DebTim<ETX>
	02 73 57 4E 20 44 49 33 44 65 62 54 69 6D 03
CoLa B	02 02 02 02 00 00 00 0E 73 57 41 20 44 49 33 44 65 62 54 69 6D 20 48

12.2.1.4.7.7 Reset output counter [sMN LIDrstoutpcnt]

Reset the counter which keeps track of how often an digital output has been active (and not active). Information from the counter is included in `LIDoutputstate` (see "Receive outputstate by event [sEN LIDoutputstate]", page 148).

Table 232: Telegram structure: sMN LIDrstoutpcnt

Telegram structure: sMN LIDrstoutpcnt (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3		sMN	73 4D 4E
Command	Reset output counter	String	13		LIDrstoutpcnt	4C 49 44 72 73 74 6F 75 74 70 63 6E 74

Table 233: Example: sMN LIDrstoutpcnt

CoLa A	<STX>sMN{SPC}LIDrstoutpcnt<ETX>
	<STX>sMN LIDrstoutpcnt<ETX>
	sMN LIDrstoutpcnt
	02 73 4D 4E 20 4C 49 44 72 73 74 6F 75 74 70 63 6E 74 03
CoLa B	02 02 02 02 00 00 00 11 73 4D 4E 20 4C 49 44 72 73 74 6F 75 74 70 63 6E 74 03
	73 4D 4E 20 4C 49 44 72 73 74 6F 75 74 70 63 6E 74

Table 234: Telegram structure: sAN LIDrstoutpcnt

Telegram structure: sAN LIDrstoutpcnt						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sAN	73 41 4E
Command	Reset output counter	String	13		LIDrstoutpcnt	4C 49 44 72 73 74 6F 75 74 70 63 6E 74
Status code	Code number	Bool_1	1	Success: Error:	0 1	00 01

Table 235: Example: sAN LIDrstoutpcnt

CoLa A	<STX>sAN{SPC}LIDrstoutpcnt{SPC}0<ETX>
	02 73 41 4E 20 4C 49 44 72 73 74 6F 75 74 70 63 6E 74 20 30 03
CoLa B	02 02 02 02 00 00 00 13 73 41 4E 20 4C 49 44 72 73 74 6F 75 74 70 63 6E 74 20 00 2F

12.2.1.4.8 Status

12.2.1.4.8.1 Read firmware version [sRN DeviceIdent]

Table 236: Telegram structure: sRN DeviceIdent

Telegram structure: sRN DeviceIdent						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E
Command	Read firmware version	String	11		DeviceIdent	44 65 76 69 63 65 49 64 65 6E 74

Table 237: Example: sRN DeviceIdent

CoLa A	<STX>sRN{SPC}DeviceIdent<ETX>	
	<STX>sRN DeviceIdent<ETX>	
	sRN DeviceIdent	
	02 73 52 4E 20 44 65 76 69 63 65 49 64 65 6E 74 03	
CoLa B	02 02 02 02 00 00 00 0F 73 52 4E 20 44 65 76 69 63 65 49 64 65 6E 74 25	
	73 52 4E 20 44 65 76 69 63 65 49 64 65 6E 74	

Table 238: Telegram structure: sRA DeviceIdent

Telegram structure: sRA DeviceIdent						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sRA	73 52 41
Command		String	11		DeviceIdent	44 65 76 69 63 65 49 64 65 6E 74
Value	Length of firmware designation	Enum_16	2		0 ... 22h	0 ... 22h

Telegram structure: sRA DeviceIdent						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Value	Firmware designation for device family	String			(See example)	(See example)
Value	Length of firmware version	Enum_16	2		0 ... 22h	0 ... 22h
Value	Firmware version	String			(See example)	(See example)

Table 239: Example: sRA DeviceIdent

CoLa A	<STX>sRA{SPC}DeviceIdent{SPC}10{SPC}LMS10x_FieldEval{SPC}10{SPC}V1.36-21.10.2010<ETX>
CoLa B	02 02 02 02 00 00 00 34 73 52 41 20 44 65 76 69 63 65 49 64 65 6E 74 20 00 10 4C 4D 53 31 30 78 5F 46 69 65 6C 64 45 76 61 6C 00 10 56 31 2E 33 36 2D 32 31 2E 31 30 2E 32 30 31 30 62

12.2.1.4.8.2 Read the device state [sRN SCdevicestate]

This telegram reads the general device state.

NOTE
The status of the measurement function can be read separately with the telegram `STlms`.

Table 240: Telegram structure: sRN SCdevicestate

Telegram structure: sRN SCdevicestate						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E
Command	Read the device state	String	13		SCdevicestate	53 43 64 65 76 69 63 65 73 74 61 74 65

Table 241: Example: sRN SCdevicestate

CoLa A	<STX>sRN{SPC}SCdevicestate<ETX>
	<STX>sRN SCdevicestate<ETX>
	sRN SCdevicestate
	02 73 52 4E 20 53 43 64 65 76 69 63 65 73 74 61 74 65 03
CoLa B	02 02 02 02 00 00 00 11 73 52 4E 20 53 43 64 65 76 69 63 65 73 74 61 74 65 30
	73 52 4E 20 53 43 64 65 76 69 63 65 73 74 61 74 65

Table 242: Telegram structure: sRA SCdevicestate

Telegram structure: sRA SCdevicestate						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sRA	73 52 41
Command	Read the device state	String	13		SCdevicestate	53 43 64 65 76 69 63 65 73 74 61 74 65

Telegram structure: sRA SCdevicestate						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Status code	Code number	Enum_8	1	Busy / logged-in: Ready: Error: Standby:	0 1 2 3	00 01 02 03

Table 243: Example: sRA SCdevicestate

CoLa A	<STX>sRA{SPC}SCdevicestate{SPC}1<ETX>
	02 73 52 41 20 53 43 64 65 76 69 63 65 73 74 61 74 65 20 31 03
CoLa B	02 02 02 02 00 00 00 13 73 52 41 20 53 43 64 65 76 69 63 65 73 74 61 74 65 20 01 1E

12.2.1.4.8.3 Read device order number [sRN DIornr]

This telegram reads the device order number.

Table 244: Telegram structure: sRN DIornr

Telegram structure: sRN DIornr						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E
Command	Read device order number	String	6		DIornr	44 49 6F 72 6E 72

Table 245: Example: sRN DIornr

CoLa A	<STX>sRN{SPC}DIornr<ETX>
	<STX>sRN DIornr<ETX>
	sRN DIornr
	02 73 52 4E 20 44 49 6F 72 6E 72 03
CoLa B	02 02 02 02 00 00 00 0A 73 52 4E 20 44 49 6F 72 6E 72 43
	73 52 4E 20 44 49 6F 72 6E 72

Table 246: Telegram structure: sRA DIornr

Telegram structure: sRA DIornr						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sRA	73 52 41
Command	Read device order number	String	6		DIornr	44 49 6F 72 6E 72
Order number	Order number in 7 digits	String	7		0000000 ... 9999999	30 30 30 30 30 30 30 ... 39 39 39 39 39 39 39

12.2.1.4.8.4 Read device type [sRN DItype]

This telegram asks for the device type of the product familiy.

Table 247: Telegram structure: sRN DItype

Telegram structure: sRN DItype						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E
Command	Ask state	String	6		DItype	44 49 74 79 70 65

Table 248: Example: sRN DItype

CoLa A	<STX>sRN{SPC}DItype<ETX>
	<STX>sRN DItype<ETX>
	sRN DItype
	02 73 52 4E 20 44 49 74 79 70 65 03
CoLa B	02 02 02 02 00 00 00 0A 73 52 4E 20 44 49 74 79 70 65 5A
	73 52 4E 20 44 49 74 79 70 65

Table 249: Telegram structure: sRA DItype

Telegram structure: sRA DItype						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sRA	73 52 41
Command	Ask state	String	6		DItype	44 49 74 79 70 65
Length of type key	Number of digits of the following type code length	Uint_8	1		0d ... 255d (0h ... FFh)	00 ... FF
Device type	Type code of the device	String	(var.)		(Device type)	(Device type)

Table 250: sRA DItype Example for LMS111-10100:

CoLa A	<STX>sRA{SPC}DItype{SPC}C{SPC}LMS111-10100<ETX>
	02 73 52 41 20 44 49 74 79 70 65 20 43 20 4C 4D 53 31 31 31 2D 31 30 31 30 30 03
CoLa B	02 02 02 02 00 00 00 18 73 52 41 20 44 49 74 79 70 65 20 00 0C 4C 4D 53 31 31 31 2D 31 30 31 30 30 07

12.2.1.4.8.5 Read operating hours [sRN ODoprh]

Table 251: Telegram structure: sRN ODoprh

Telegram structure: sRN ODoprh						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E
Command	Read operating hours	String	6		ODoprh	4F 44 6F 70 72 68

Table 252: Example: sRN ODoprh

CoLa A	<STX>sRN{SPC}ODoprh<ETX>
	<STX>sRN ODoprh<ETX>
	sRN ODoprh
	02 73 52 4E 20 4F 44 6F 70 72 68 03
CoLa B	02 02 02 02 00 00 00 0A 73 52 4E 20 4F 44 6F 70 72 68 41
	73 52 4E 20 4F 44 6F 70 72 68

Table 253: Telegram structure: sRA ODoprh

Telegram structure: sRA ODoprh						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sRA	73 52 41
Command	Read operating hours	String	6		ODoprh	4F 44 6F 70 72 68
Value	Operating hours in 1/10 h	Uint_32	4		0h … FFFFFFFFh	00 00 00 00 … FF FF FF FF

Table 254: Example: sRA ODoprh

CoLa A	<STX>sRA{SPC}ODoprh{SPC}2DC8B<ETX>
	02 73 52 41 20 4F 44 6F 70 72 68 20 32 44 43 38 42 03
CoLa B	02 02 02 02 00 00 00 0F 73 52 41 20 4F 44 6F 70 72 68 20 00 02 DC 8B 36

Calculation of the value: 2DC8B (hex) → 187531 (dez) × 1/10 h = 18753.1 h

12.2.1.4.8.6 Read power on counter [sRN ODpwrc]

Shows the number of power on cycles.

Table 255: Telegram structure: sRN ODpwrc

Telegram structure: sRN ODpwrc						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E
Command	Read power on counter	String	6		ODpwrc	4F 44 70 77 72 63

Table 256: Example: sRN ODpwrc

CoLa A	<STX>sRN{SPC}ODpwrc<ETX>
	<STX>sRN ODpwrc<ETX>
	sRN ODpwrc
	02 73 52 4E 20 4F 44 70 77 72 63 03
CoLa B	02 02 02 02 00 00 00 0A 73 52 4E 20 4F 44 70 77 72 63 52
	73 52 4E 20 4F 44 70 77 72 63

Table 257: Telegram structure: sRA ODpwrc

Telegram structure: sRA ODpwrc						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sRA	73 52 41
Command	Read power on counter	String	6		ODpwrc	4F 44 70 77 72 63
Value	Power on counter	Uint_32	4		0h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF

Table 258: Example: sRA ODpwrc

CoLa A	<STX>sRA{SPC}ODpwrc{SPC}752D<ETX>
	02 73 52 41 20 4F 44 70 77 72 63 20 752D 03
CoLa B	02 02 02 02 00 00 00 0F 73 52 41 20 4F 44 70 77 72 63 20 00 00 75 2D 36

12.2.1.4.8.7 Read temperature [sRN OPcurtmpdev]

With this command the internal temperature of the device can be identified. Please note that it does not give an indication of the current ambient temperature.

Table 259: Telegram structure: sRN OPcurtmpdev

Telegram structure: sRN OPcurtmpdev						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E
Command	Read temperature of the device	String	11		OPcurtmpdev	4F 50 63 75 72 74 6D 70 64 65 76

Table 260: Example: sRN OPcurtmpdev

CoLa A	<STX>sRN{SPC}OPcurtmpdev<ETX>	
	<STX>sRN OPcurtmpdev<ETX>	
	sRN OPcurtmpdev	
	02 73 52 4E 20 4F 50 63 75 72 74 6D 70 64 65 76 03	
CoLa B	02 02 02 02 00 00 00 0F 73 52 4E 20 4F 50 63 75 72 74 6D 70 64 65 76 2A	
	73 52 4E 20 4F 50 63 75 72 74 6D 70 64 65 76	

Table 261: Telegram structure: sRA OPcurtmpdev

Telegram structure: sRA OPcurtmpdev						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sRA	73 52 41
Command	Read temperature of the device	String	11		OPcurtmpdev	4F 50 63 75 72 74 6D 70 64 65 76

Telegram structure: sRA OPcurtmpdev						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Tempera-ture data	[°C]	Real as float accord-ing to IEEE754	4	(-50°C ... +100°C)	C2480000h ... 42C80000h	C2 48 00 00 ... 42 C8 00 00

Example: sRA OPcurtmpdev (35°C)

Table 262: Example: sRA OPcurtmpdev

CoLa A	<STX>sRA{SPC}OPcurtmpdev{SPC}420C0000<ETX>
	02 73 52 41 20 4F 50 63 75 72 74 6D 70 64 65 76 20 34 32 30 43 30 30 30 30 03
CoLa B	02 02 02 02 00 00 00 14 73 52 41 20 4F 50 63 75 72 74 6D 70 64 65 76 20 42 0C 00 00 4B

12.2.1.4.8.8 Set device name [sWN LocationName]

Give the device a specific description name such as its location.

Table 263: Telegram structure: sWN LocationName

Telegram structure: sWN LocationName (User level 'Maintenance' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3		sWN	73 57 4E
Command	Set device name	String	12		LocationName	4C 6F 63 61 74 69 6F 6E 4E 61 6D 65
Value	Number of characters of the following device name	Uint_16	2		0d ... +16d (0h ... 10h)	00 00 ... 00 10
Value	Device name	String	16		[Device name]	[Device name]

Table 264: Example: sWN LocationName +13 OutdoorDevice

CoLa A	<STX>sWN{SPC}LocationName{SPC}+13{SPC}OutdoorDevice<ETX>
	<STX>sWN LocationName +13 OutdoorDevice<ETX>
	sWN LocationName +13 OutdoorDevice
	02 73 57 4E 20 4C 6F 63 61 74 69 6F 6E 4E 61 6D 65 20 2B 31 33 20 4F 75 74 64 6F 6F 72 44 65 76 69 63 65 03
CoLa B	02 02 02 02 00 00 00 20 73 57 4E 20 4C 6F 63 61 74 69 6F 6E 4E 61 6D 65 20 00 0D 4F 75 74 64 6F 6F 72 44 65 76 69 63 65 1D
	73 57 4E 20 4C 6F 63 61 74 69 6F 6E 4E 61 6D 65 20 00 0D 4F 75 74 64 6F 6F 72 44 65 76 69 63 65

Table 265: Telegram structure: sWA LocationName

Telegram structure: sWA LocationName						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sWA	73 57 41
Command	Set device name	String	12		LocationName	4C 6F 63 61 74 69 6F 6E 4E 61 6D 65

Table 266: Example: sWA LocationName

CoLa A	<STX>sWA{SPC}LocationName<ETX>
	02 73 57 41 20 4C 6F 63 61 74 69 6F 6E 4E 61 6D 65 03
CoLa B	02 02 02 02 00 00 00 11 73 57 41 20 4C 6F 63 61 74 69 6F 6E 4E 61 6D 65 20 7F

12.2.1.4.8.9 Read device name [sRN LocationName]

Read the given name of the device (Default is the serial number of the device).

Table 267: Telegram structure: sRN LocationName

Telegram structure: sRN LocationName						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E
Command	Read device name	String	12		LocationName	4C 6F 63 61 74 69 6F 6E 4E 61 6D 65

Table 268: Example: sRN LocationName

CoLa A	<STX>sRN{SPC}LocationName<ETX>	
	<STX>sRN LocationName<ETX>	
	sRN LocationName	
	02 73 52 4E 20 4C 6F 63 61 74 69 6F 6E 4E 61 6D 65 03	
CoLa B	02 02 02 02 00 00 00 10 73 52 4E 20 4C 6F 63 61 74 69 6F 6E 4E 61 6D 65 55	
	73 52 4E 20 4C 6F 63 61 74 69 6F 6E 4E 61 6D 65	

Table 269: Telegram structure: sRA LocationName

Telegram structure: sRA LocationName						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Find complete telegram structure of the answer in see table 263, page 158						

12.2.1.4.8.10 Read heating state [sRN OPheatstateext]

Read if the heating is connected and its status.

NOTE
It is not allowed to request this telegram in a faster cycle than 10 ms!

Table 270: Telegram structure: sRN OPheatstateext

Telegram structure: sRN OPheatstateext						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E
Command		String	14		OPheatstateext	4F 50 68 65 61 74 73 74 61 74 65 65 78 74

Table 271: Example: sRN OPheatstateext

CoLa A	<STX>sRN{SPC}OPheatstateext<ETX>	
	<STX>sRN OPheatstateext<ETX>	
	sRN OPheatstateext	
	02 73 52 4E 20 4F 50 68 65 61 74 73 74 61 74 65 65 78 74 03	
CoLa B	02 02 02 02 00 00 00 12 73 52 4E 20 4F 50 68 65 61 74 73 74 61 74 65 65 78 74 56	
	73 52 4E 20 4F 50 68 65 61 74 73 74 61 74 65 65 78 74	

Table 272: Telegram structure: sRA OPheatstateext

Telegram structure: sRA OPheatstateext						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sRA	73 52 41
Command		String	14		OPheatstateext	4F 50 68 65 61 74 73 74 61 74 65 65 78 74
Heating state	Heating state	Enum_8	1	Heating active: Electrical current for heating too low: Heating inactive: Power supply for heating not connected:	0d 1d 2d 3d	00 01 02 03

Table 273: Example: sRA OPheatstateext 2

CoLa A	<STX>sRA{SPC}OPheatstateext{SPC}2<ETX>
	02 73 52 41 20 4F 50 68 65 61 74 73 74 61 74 65 65 78 74 20 32 03
CoLa B	02 02 02 02 00 00 00 14 73 52 41 20 4F 50 68 65 61 74 73 74 61 74 65 65 78 74 20 02 7B

12.2.1.4.9 Interfaces

12.2.1.4.9.1 Set IP address [sWN EIIpAddr]

NOTE

- Save permanently to set values. Changes will be active after rebooting the device.
- Settings must correspond with network in which scanner is used. Else device cannot be found any more.

Table 274: Telegram structure: sWN EIIpAddr

Telegram structure: sWN EIIpAddr (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3		sWN	73 57 4E
Command	Set IP address	String	8		EIIpAddr	45 49 49 50 41 64 64 72

Telegram structure: sWN EIIpAddr (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
IP address	Set values	Uint_8	1	First part of IP adress	0 ...+255d (00 ... FF)	00 ... FF
				Second part of IP adress	0 ...+255d (00 ... FF)	00 ... FF
				Third part of IP adress	0 ...+255d (00 ... FF)	00 ... FF
				Fourth part of IP adress	0 ...+255d (00 ... FF)	00 ... FF

Table 275: Example: sWN EIIpAddr 192.168.0.2

CoLa A	<STX>sWN{SPC}EIIpAddr{SPC}C0{SPC}A8{SPC}0{SPC}2<ETX>
	<STX>sWN EIIpAddr C0 A8 0 2<ETX>
	sWN EIIpAddr C0 A8 0 2
	02 73 57 4E 20 45 49 49 70 41 64 64 72 20 43 30 20 41 38 20 30 20 32 03
CoLa B	02 02 02 02 00 00 00 11 73 57 4E 20 45 49 49 70 41 64 64 72 20 C0 A8 00 02 06
	73 57 4E 20 45 49 49 70 41 64 64 72 20 C0 A8 00 02

Table 276: Telegram structure: sWA EIIpAddr

Telegram structure: sWA EIIpAddr						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sWA	73 57 41
Command	Set IP address	String	8		EIIpAddr	45 49 49 50 41 64 64 72

Table 277: Example: sWA EIIpAddr

CoLa A	<STX>sWA{SPC}EIIpAddr<ETX>
	02 73 57 41 20 45 49 49 70 41 64 64 72 03
CoLa B	02 02 02 02 00 00 00 0D 73 57 41 20 45 49 49 70 41 64 64 72 20 63

12.2.1.4.9.2 Read IP address [sRN EIIpAddr]

Read the IP address of the device.

Table 278: Telegram structure: sRN EIIpAddr

Telegram structure: sRN EIIpAddr						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E
Command	Read IP address	String	8		EIIpAddr	45 49 49 50 41 64 64 72

Table 279: Example: sRN EIIpAddr

CoLa A	<STX>sRN{SPC}EIIpAddr<ETX>
	<STX>sRN EIIpAddr<ETX>
	sRN EIIpAddr
	02 73 52 4E 20 45 49 49 70 41 64 64 72 03

CoLa B	02 02 02 02 00 00 00 0C 73 52 4E 20 45 49 49 70 41 64 64 72 49 73 52 4E 20 45 49 49 70 41 64 64 72

Table 280: Telegram structure: sRA EIIpAddr

Telegram structure: sRA EIIpAddr						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sRA	73 52 41
Command	Read IP address	String	8		EIIpAddr	45 49 49 50 41 64 64 72
IP address	Default: 192.168.0.1	Uint_8	1	First part of IP address	0 ...+255d (00 ... FF)	00 ... FF
				Second part of IP address	0 ...+255d (00 ... FF)	00 ... FF
				Third part of IP address	0 ...+255d (00 ... FF)	00 ... FF
				Fourth part of IP address	0 ...+255d (00 ... FF)	00 ... FF

Table 281: Example: sRA EIIpAddr 192.168.0.2

CoLa A	<STX>sRA{SPC}EIIpAddr{SPC}C0{SPC}A8{SPC}00{SPC}02<ETX>
	02 73 57 41 20 45 49 49 70 41 64 64 72 20 C0 20 A8 20 00 20 02 03
CoLa B	02 02 02 02 00 00 00 11 73 52 41 20 45 49 49 70 41 64 64 72 20 C0 A8 00 02 0C

12.2.1.4.9.3 Set Ethernet gateway [sWN Elgate]

Change Ethernet gateway IP address (TCP/IP)

NOTE

- Save permanently to set values. Changes will be active after rebooting the device.
- Settings must correspond with network in which scanner is used. Else device cannot be found any more.

Table 282: Telegram structure: sWN Elgate

Telegram structure: sWN Elgate (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3		sWN	73 57 4E
Command	Set gateway adress	String	6		Elgate	45 49 67 61 74 65
Gateway address	Set values	Uint_8	1	First part of gateway address	0 ...+255d (00... FF)	00 ...FF
				Second part of gateway address	0 ...+255d (00... FF)	00 ...FF
				Third part of gateway address	0 ...+255d (00... FF)	00 ...FF
				Fourth part of gateway address	0 ...+255d (00... FF)	00 ...FF

Table 283: Example: sWN Elgate 192.168.0.1

CoLa A	<STX>sWN{SPC}Elgate{SPC}C0{SPC}A8{SPC}00{SPC}01<ETX>
	<STX>sWN Elgate C0 A8 00 01<ETX>
	sWN Elgate C0 A8 00 01
	02 73 57 4E 20 45 49 67 61 74 65 20 43 30 20 41 38 20 30 30 20 30 31 03
CoLa B	02 02 02 02 00 00 00 0F 73 57 4E 20 45 49 67 61 74 65 20 C0 A8 00 01 18
	73 57 4E 20 45 49 67 61 74 65 20 C0 A8 00 01

Table 284: Telegram structure: sWA Elgate

Telegram structure: sWA Elgate						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sWA	73 57 41
Command	Set gateway adress	String	6		Elgate	45 49 67 61 74 65

Table 285: Example: sWA Elgate

CoLa A	<STX>sWA{SPC}Elgate<ETX>
	02 73 57 41 20 45 49 67 61 74 65 03
CoLa B	02 02 02 02 00 00 00 0B 73 57 41 20 45 49 67 61 74 65 20 7E

12.2.1.4.9.4 Read Ethernet gateway [sRN Elgate]

Read for the Ethernet gateway (TCP/IP)

Table 286: Telegram structure: sRN Elgate

Telegram structure: sRN Elgate						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E
Command	Read gateway address	String	6		Elgate	45 49 67 61 74 65

Table 287: Example: sRN Elgate

CoLa A	<STX>sRN{SPC}Elgate<ETX>
	<STX>sRN Elgate<ETX>
	sRN Elgate
	02 73 52 4E 20 45 49 67 61 74 65 03
CoLa B	02 02 02 02 00 00 00 0A 73 52 4E 20 45 49 67 61 74 65 54
	73 52 4E 20 45 49 67 61 74 65

Table 288: Telegram structure: sRA Elgate

Telegram structure: sRA Elgate						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sRA	73 52 41

Telegram structure: sRA Elgate						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command	Read gateway address	String	6		Elgate	45 49 67 61 74 65
Gateway address	Default: 0.0.0.0	Uint_8	1	First part of gateway address	0 ...+255d (00... FF)	00 ... FF
				Second part of gateway address	0 ...+255d (00... FF)	00 ... FF
				Third part of gateway address	0 ...+255d (00... FF)	00 ... FF
				Fourth part of gateway address	0 ...+255d (00... FF)	00 ... FF

Table 289: Example: sRA Elgate 192.168.0.1

CoLa A	<STX>sRA{SPC}Elgate{SPC}C0{SPC}A8{SPC}00{SPC}01<ETX>
	02 73 52 41 20 45 49 67 61 74 65 20 C0 A8 00 01 03
CoLa B	02 02 02 02 00 00 00 0F 73 52 41 20 45 49 67 61 74 65 20 C0 A8 00 01 12

12.2.1.4.9.5 Set IP mask [sWN EImask]

Define the subnet mask of the device.

NOTE

- Save permanently to set values. Changes will be active after rebooting the device.
- Settings must correspond with network in which scanner is used. Else device cannot be found any more.

Table 290: Telegram structure: sWN EImask

Telegram structure: sWN EImask (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3		sWN	73 57 4E
Command	Set IP mask	String	6		EImask	45 49 6D 61 73 6B
IP mask	Set values	Uint_8	1	First part of IP mask	0 ...+255d (00... FF)	00 ...FF
				Second part of IP mask	0 ...+255d (00... FF)	00 ...FF
				Third part of IP mask	0 ...+255d (00... FF)	00 ...FF
				Fourth part of IP mask	0 ...+255d (00... FF)	00 ...FF

Table 291: Example: sWN EImask 255.255.254.0

CoLa A	<STX>sWN{SPC}EImask{SPC}FF{SPC}FF{SPC}FE{SPC}00<ETX>
	<STX>sWN EImask FF FF FE 00<ETX>
	sWN EImask FF FF FE 00
	02 73 57 4E 20 45 49 6D 61 73 6B 20 46 46 20 46 46 20 46 45 20 30 30 03
CoLa B	02 02 02 02 00 00 00 0F 73 57 4E 20 45 49 6D 61 73 6B 20 FF FF FE 00 8C
	73 57 4E 20 45 49 6D 61 73 6B 20 FF FF FE 00

Table 292: Telegram structure: sWA EImask

Telegram structure: sWA EImask						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sWA	73 57 41
Command	Set IP mask	String	6		EImask	45 49 6D 61 73 6B

Table 293: Example: sWA EImask

CoLa A	<STX>sWA{SPC}EImask<ETX>
	02 73 57 41 20 45 49 6D 61 73 6B 03
CoLa B	02 02 02 02 00 00 00 0B 73 57 41 20 45 49 6D 61 73 6B 20 7D

12.2.1.4.9.6 Read IP mask [sRN EImask]

Read the subnet mask of the device.

Table 294: Telegram structure: sRN EImask

Telegram structure: sRN EImask						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E
Command	Read IP mask	String	6		EImask	45 49 6D 61 73 6B

Table 295: Example: sRN EImask

CoLa A	<STX>sRN{SPC}EImask<ETX>
	<STX>sRN EImask<ETX>
	sRN EImask
	02 73 52 4E 20 45 49 6D 61 73 6B 03
CoLa B	02 02 02 02 00 00 00 0A 73 52 4E 20 45 49 6D 61 73 6B 57
	73 52 4E 20 45 49 6D 61 73 6B

Table 296: Telegram structure: sRA EImask

Telegram structure: sRA EImask						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sRA	73 52 41
Command	Read IP mask	String	6		EImask	45 49 6D 61 73 6B
IP mask	Default: 255.255.255.0	Uint_8	1	First part of IP mask	0 ...+255d (00... FF)	00 ... FF
				Second part of IP mask	0 ...+255d (00... FF)	00 ... FF
				Third part of IP mask	0 ...+255d (00... FF)	00 ... FF
				Fourth part of IP mask	0 ...+255d (00... FF)	00 ... FF

Table 297: Example: sRA EImask 255.255.254.0

CoLa A	<STX>sRA{SPC}EImask{SPC}FF{SPC}FF{SPC}FE{SPC}00<ETX>
	02 73 52 41 20 45 49 6D 61 73 6B 20 45 49 6D 61 73 6B 03
CoLa B	02 02 02 02 00 00 00 0F 73 52 41 20 45 49 6D 61 73 6B 20 FF FF FE 00 86

12.2.1.4.9.7 Read MAC address [sRN EIMacAdr]

Read the MAC address of the device.

Table 298: Telegram structure: sRN EIMacAdr

Telegram structure: sRN EIMacAdr						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E
Command	Read MAC address of the device	String	8		EIMacAdr	45 49 4D 61 63 41 64 72

Table 299: Example: sRN EIMacAdr

CoLa A	<STX>sRN{SPC}EIMacAdr<ETX>	
	<STX>sRN EIMacAdr<ETX>	
	sRN EIMacAdr	
	02 73 57 4E 20 45 49 4D 61 63 41 64 72 03	
CoLa B	02 02 02 02 00 00 00 0C 73 52 4E 20 45 49 4D 61 63 41 64 72 5B	
	73 52 4E 20 45 49 4D 61 63 41 64 72	

Table 300: Telegram structure: sRA EIMacAdr

Telegram structure: sRA EIMacAdr						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sRA	73 52 41
Command	Read MAC address of the device	String	8		EIMacAdr	45 49 4D 61 63 41 64 72
MAC address	Values	Uint_8	1	First part of MAC address	0 ...+255d (00... FF)	00 ... FF
		Uint_8	1	Second part of MAC address	0 ...+255d (00... FF)	00 ... FF
		Uint_8	1	Third part of MAC address	0 ...+255d (00... FF)	00 ... FF
		Uint_8	1	Fourth part of MAC address	0 ...+255d (00... FF)	00 ... FF
		Uint_8	1	Fifth part of MAC address	0 ...+255d (00... FF)	00 ... FF
		Uint_8	1	Sixth part of MAC address	0 ...+255d (00... FF)	00 ... FF

Table 301: Example: sRA EIMacAdr 00:06:77:22:40:EA

CoLa A	<STX>sRA{SPC}EIMacAdr{SPC}0{SPC}6{SPC}77{SPC}22{SPC}40{SPC}EA<ETX>
	02 73 52 41 20 45 49 4D 61 63 41 64 72 20 30 20 36 20 37 37 20 32 32 20 34 30 20 45 41 03
CoLa B	02 02 02 02 00 00 00 13 73 52 41 20 45 49 4D 61 63 41 64 72 20 00 06 77 22 40 EA 8D

12.2.1.4.9.8 Set baud rate for host interface [sWN SIHstBaud]

Define the rate in bits/s for the serial host interface.

Table 302: Telegram structure: sWN SIHstBaud

Telegram structure: sWN SIHstBaud (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3		sWN	73 57 4E
Command	Set baud rate for host interface	String	9		SIHstBaud	53 49 48 73 74 42 61 75 64
Baud rate data	Baud rate data for host interface	Enum_8	1	9600: 19200: 38400: 57600: 115200: 250000: 500000:	+5d (05h) +6d (06h) +7d (07h) +8d (08h) +9d (09h) +10d (0Ah) +11d (0Bh)	05 06 07 08 09 0A 0B

Table 303: Example: sWN SIHstBaud

CoLa A	<STX>sWN{SPC}SIHstBaud{SPC}+8<ETX>	
	<STX>sWN SIHstBaud +8<ETX>	
	sWN SIHstBaud +8	
	02 73 57 4E 20 53 49 48 73 74 42 61 75 64 20 08 03	
CoLa B	02 02 02 02 00 00 00 0F 73 57 4E 20 53 49 48 73 74 42 61 75 64 20 08 05	
	73 57 4E 20 53 49 48 73 74 42 61 75 64 20 08	

Table 304: Telegram structure: sWA SIHstBaud

Telegram structure: sWA SIHstBaud						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sWA	73 57 41
Command	Set baud rate for host interface	String	9		SIHstBaud	53 49 48 73 74 42 61 75 64

Table 305: Example: sWA SIHstBaud

CoLa A	<STX>sWA{SPC}SIHstBaud<ETX>
	02 73 57 41 20 53 49 48 73 74 42 61 75 64 03
CoLa B	02 02 02 02 00 00 00 0E 73 57 41 20 53 49 48 73 74 42 61 75 64 20 02

12.2.1.4.9.9 Read baud rate of host interface [sRN SIHstBaud]

Read the configured rate in bits/s of the serial host interface.

Table 306: Telegram structure: sRN SIHstBaud

Telegram structure: sRN SIHstBaud						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E

Telegram structure: sRN SIHstBaud						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command	Read baud rate of host interface	String	9		SIHstBaud	53 49 48 73 74 42 61 75 64

Table 307: Example: sRN SIHstBaud

CoLa A	<STX>sRN{SPC}SIHstBaud<ETX>
	<STX>sRN SIHstBaud<ETX>
	sRN SIHstBaud
	02 73 52 4E 20 53 49 48 73 74 42 61 75 64 03
CoLa B	02 02 02 02 00 00 00 0D 73 52 4E 20 53 49 48 73 74 42 61 75 64 28
	73 52 4E 20 53 49 48 73 74 42 61 75 64

Table 308: Telegram structure: sRA SIHstBaud

Telegram structure: sRA SIHstBaud						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sRA	73 52 41
Command	Read baud rate of host interface	String	9		SIHstBaud	53 49 48 73 74 42 61 75 64
Baud rate data	Baud rate data of host interface	Enum_8	1	9600: 19200: 38400: 57600: 115200: 250000: 500000:	5d (05h) 6d (06h) 7d (07h) 8d (08h) 9d (09h) 10d (0Ah) 11d (0Bh)	05 06 07 08 09 0A 0B

Table 309: Example: sRA SIHstBaud

CoLa A	<STX>sRA{SPC}SIHstBaud{SPC}8<ETX>
	02 73 52 41 20 53 49 48 73 74 42 61 75 64 20 08 03
CoLa B	02 02 02 02 00 00 00 0F 73 52 41 20 53 49 48 73 74 42 61 75 64 20 08 0F

12.2.1.4.9.10 Set Host port number [sWN EIHstPort]

Define the TCP/IP host port number.

Table 310: Telegram structure: sWN EIHstPort

Telegram structure: sWN EIHstPort (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3		sWN	73 57 4E
Command	Read hardware settings of host interface	String	7	Ethernet host TCP/ IP	EIHstPort	45 49 48 73 74 50 6F 72 74
Port number	Host port number (Default = 2112)	Uint_16	2		+0d ... +65535d (0h ... FF FFh)	00 00 ... FF FF

Example: sWN EIHstPort +2110 (Host Port 2110)

Table 311: Example: sWN EIHstPort +2110

CoLa A	<STX>sWN{SPC}EIHstPort{SPC}+2110<ETX>
	<STX>sWN EIHstPort +2110<ETX>
	sWN EIHstPort +2110
	02 73 57 4E 20 45 49 48 73 74 50 6F 72 74 20 2B 32 31 31 30 03
CoLa B	02 02 02 02 00 00 00 10 73 57 4E 20 45 49 48 73 74 50 6F 72 74 20 08 3E 26
	73 57 4E 20 45 49 48 73 74 50 6F 72 74 20 08 3E

Table 312: Telegram structure: sWA EIHstPort

Telegram structure: sWA EIHstPort						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sWA	73 57 41
Command	Read hardware settings of host interface	String	7	Ethernet host TCP/ IP	EIHstPort	45 49 48 73 74 50 6F 72 74

Example: sWA EIHstPort (Host Port)

Table 313: Example: sWA EIHstPort

CoLa A	<STX>sWA{SPC}EIHstPort<ETX>
	02 73 57 41 20 45 49 48 73 74 50 6F 72 74 03
CoLa B	02 02 02 02 00 00 00 0E 73 57 41 20 45 49 48 73 74 50 6F 72 74 20 1F

12.2.1.4.9.11 Set Host port Command Language (CoLa dialect) [sWN EIHstCola]

Define the communcation language (SICK specific) of the TCP/IP host port.

NOTE

It is not allowed to use this telegram in a faster cycle than 10 ms!

After switching the CoLa dialect by this telegram, you have to store the changes permanentely and reboot the sensor to activate the chosen CoLA dialect.

Binary CRC32 is available since the firmware version V1.80.0

Table 314: Telegram structure: sWN EIHstCola

Telegram structure: sWN EIHstCola (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3		sWN	73 57 4E
Command	Set hardware settings of host interface	String	7		EIHstCola	45 49 48 73 74 43 6F 6C 61
Command language	Host port Command language	Enum_8	1	CoLa ASCII: CoLa binary: Binary CRC32:	0 1 2	00 01 02

Example: sWN EIHstCola 1 (Host Port CoLa binary)

Table 315: Example: sWN EIHstCola 1

CoLa A	<STX>sWN{SPC}EIHstCola{SPC}1<ETX>
	<STX>sWN EIHstCola 1<ETX>
	sWN EIHstCola 1
	02 73 57 4E 20 45 49 48 73 74 43 6F 6C 61 20 31 03
CoLa B	02 02 02 02 00 00 00 0F 73 57 4E 20 45 49 48 73 74 43 6F 6C 61 20 01 09
	73 57 4E 20 45 49 48 73 74 43 6F 6C 61 20 01

Table 316: Telegram structure: sWA EIHstCola

Telegram structure: sWA EIHstCola						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sWA	73 57 41
Command	Set hardware settings of host interface	String	7		EIHstCola	45 49 48 73 74 43 6F 6C 61

Example: sWA EIHstCola (Host Port)

Table 317: Example: sWA EIHstCola

CoLa A	<STX>sWA{SPC}EIHstCola<ETX>
	02 73 57 41 20 45 49 48 73 74 43 6F 6C 61 03
CoLa B	02 02 02 02 00 00 00 0E 73 57 41 20 45 49 48 73 74 43 6F 6C 61 20 07

12.2.1.4.9.12 Enable/Disable Front Panel [sWN LMLfpen]

The following telegram `LMLfpen` corresponds to the SOPAS GUI check box “Display and serial aux interface active”.

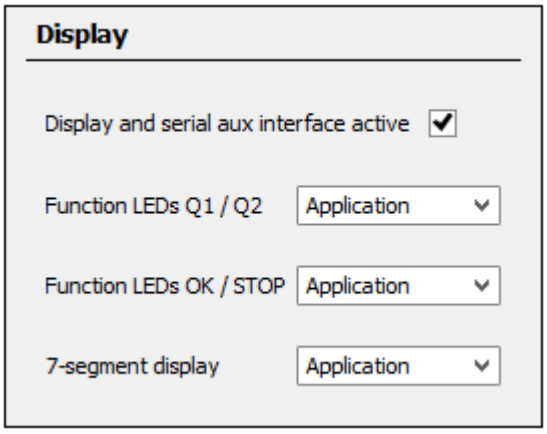


Table 318: Telegram structure: sWN LMLfpen

Telegram structure: sWN LMLfpen (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3		sWN	73 57 4E
Command	Set front panel display and serial aux interface	String	7		LMLfpen	4C 4D 4C 66 70 65 6E

Telegram structure: sWN LMLfpen (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Status	Enable/Disa-ble(Active/inactive)	Bool_1	1	Enable (Active): Disable (Inactive):	1 0	01 00

Table 319: Example: sWN LMLfpen 1 (Enable Front Panel)

CoLa A	<STX>sWN{SPC}LMLfpen{SPC}1<ETX>
	<STX>sWN LMLfpen 1<ETX>
	sWN LMLfpen 1
	02 73 57 4E 20 4C 4D 4C 66 70 65 6E 20 31 03
CoLa B	02 02 02 02 00 00 00 0D 73 57 4E 20 4C 4D 4C 66 70 65 6E 20 01 3B
	73 57 4E 20 4C 4D 4C 66 70 65 6E 20 01

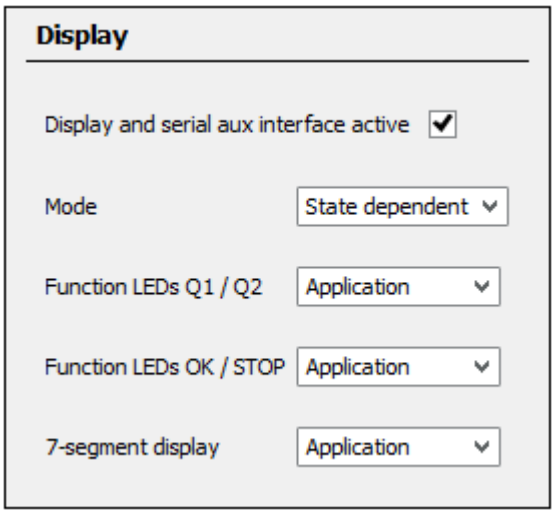
Table 320: Telegram structure: sWA LMLfpen

Telegram structure: sWA LMLfpen						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sWA	73 57 41
Command	Set front panel display and serial aux interface	String	7		LMLfpen	4C 4D 4C 66 70 65 6E

Table 321: Example: sWA LMLfpen

CoLa A	<STX>sWA{SPC}LMLfpFen<ETX>
	02 73 57 41 20 4C 4D 4C 66 70 65 6E 03
CoLa B	02 02 02 02 00 00 00 0C 73 57 41 20 4C 4D 4C 66 70 65 6E 20 35

For LMS1xx Security sensors (LMS12x, LMS13x, LMS14x), telegram `LLMfpmode` corresponds to the selection of `Mode` between `State dependent` (the mode is dependent on the state of the selection of `Arm` / `Disarmed` / `Walk Test`) or `Permanent` (the always active mode).



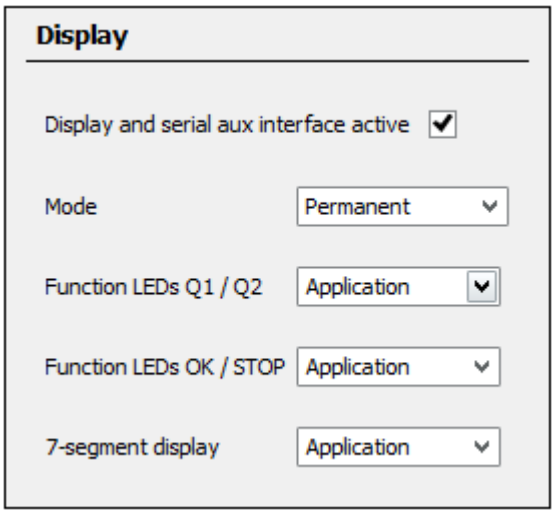


Table 322: Telegram structure: sWN LLMfpmode

Telegram structure: sWN LLMfpmode (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	LMS1xx Security	sWN	73 57 4E
Command	Set front panel display and serial aux interface	String	7	LMS1xx Security	LLMfpmode	4C 4C 4D 66 70 6D 6F 64 65
Status	Enable/Disable(Active/inactive)	Bool_1	1	LMS1xx Security Permanent (active): State dependent (Inactive):	 0 1	 00 01

Example: sWN LLMfpmode

Table 323: Example: sWN LLMfpmode 0 (set to “Permanent”)

CoLa A	<STX>sWN{SPC}LLMfpmode {SPC}1<ETX>	
	<STX>sWN LLMfpmode 1<ETX>	
	sWN LLMfpmode 1	
	02 73 57 4E 20 4C 4C 4D 66 70 6D 6F 64 65 20 31 03	
CoLa B	02 02 02 02 00 00 00 0D 73 57 4E 20 4C 4C 4D 66 70 6D 6F 64 65 20 00 12	
	73 57 4E 20 4C 4C 4D 66 70 6D 6F 64 65 20 00	

Table 324: Telegram structure: sWA LLMfpmode

Telegram structure: sWA LLMfpmode						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	LMS1xx Security	sWA	73 57 41
Command	Set front panel display and serial aux interface	String	7	LMS1xx Security	LLMfpmode	4C 4C 4D 66 70 6D 6F 64 65

Table 325: Example: sWA LLMfpmode

CoLa A	<STX>sWA{SPC}LLMfpmode <ETX>
	02 73 57 41 20 4C 4C 4D 66 70 6D 6F 64 65 03
CoLa B	02 02 02 02 00 00 00 0C 73 57 41 20 4C 4D 4C 66 70 65 6E 20 3D

12.2.1.4.9.13 Set function front panel [sWN LMLfpFcn]

Configure the function of the status LEDs and the 7-segment display on the device housing.

Table 326: Telegram structure: sWN LMLfpFcn

Telegram structure: sWN LMLfpFcn (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3		sWN	73 57 4E
Command	Set function of the front panel	String	8		LMLfpFcn	4C 4D 4C 66 70 46 63 6E
Reserved	Reserved	Bool_1	1		1	01
LED function Q1/Q2	Code number	Enum_8	1	No function: Application: Command:	0 1 2	00 01 02

Telegram structure: sWN LMLfpFcn (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
LED function OK/ Stop	Code number	Enum_8	1	Application: Command:	0 1	00 01
Display function	Code number	Enum_8	1	No function: Application: Command:	0 1 2	00 01 02

Table 327: Example: sWN LMLfpFcn

CoLa A	<STX>sWN{SPC}LMLfpFcn{SPC}1{SPC}1{SPC}0{SPC}1<ETX>
	<STX>sWN LMLfpFcn 1 1 0 1<ETX>
	sWN LMLfpFcn 1 1 0 1
	02 73 57 4E 20 4C 4D 4C 66 70 46 63 6E 20 31 20 31 20 30 20 31 03
CoLa B	02 02 02 02 00 00 00 11 73 57 4E 20 4C 4D 4C 66 70 46 63 6E 20 01 01 00 01 7B
	73 57 4E 20 4C 4D 4C 66 70 46 63 6E 20 01 01 00 01

Table 328: Telegram structure: sWA LMLfpFcn

Telegram structure: sWA LMLfpFcn						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sWA	73 57 41
Command	Set function of the front panel	String	8		LMLfpFcn	4C 4D 4C 66 70 46 63 6E

Table 329: Example: sWA LMLfpFcn

CoLa A	<STX>sWA{SPC}LMLfpFcn<ETX>
	02 73 57 41 20 4C 4D 4C 66 70 46 63 6E 03
CoLa B	02 02 02 02 00 00 00 0D 73 57 41 20 4C 4D 4C 66 70 46 63 6E 75

12.2.1.4.9.14 Set front LEDs [sMN mLMLSetLed]

To use this command, it is necessary to set the function of the LED to “Command” (use `sWN LMLfpFcn`), otherwise this command will have no influence to the LEDs.

OK and Stop LED can only alternate, if one is switched on, the other will turn automatically off.

Table 330: Telegram structure: sMN mLMLSetLed

Telegram structure: sMN mLMLSetLed						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3		sMN	73 4D 4E
Command	Set front LED	String	10		mLMLSetLed	6D 4C 4D 4C 53 65 74 4C 65 64
LED	LED to turn on/off	Int_8	1	Stop: OK: Q1: Q2:	1 2 3 4	01 02 03 04

Telegram structure: sMN mLMLSetLed						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Status	On or Off	Int_8	1	On: Off:	1 0	01 00

Table 331: Example: sMN mLMLSetLed 1 1 (Stop LED)

CoLa A	<STX>sMN{SPC}mLMLSetLed{SPC}1{SPC}1<ETX>
	<STX>sMN mLMLSetLed 1 1<ETX>
	sMN mLMLSetLed 1 1
CoLa B	02 02 02 02 00 00 00 12 73 4D 4E 20 6D 4C 4D 4C 53 65 74 4C 65 64 20 01 20 01 7F
	73 4D 4E 20 6D 4C 4D 4C 53 65 74 4C 65 64 20 01 20 01

Table 332: Telegram structure: sAN mLMLSetLed

Telegram structure: sAN mLMLSetLed						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sAN	73 41 4E
Command	Set front LED	String	10		mLMLSetLed	6D 4C 4D 4C 53 65 74 4C 65 64
Status code	Code number	Bool_1	1	Error: Success:	0 1	00 01

Table 333: Example: sAN mLMLSetLed

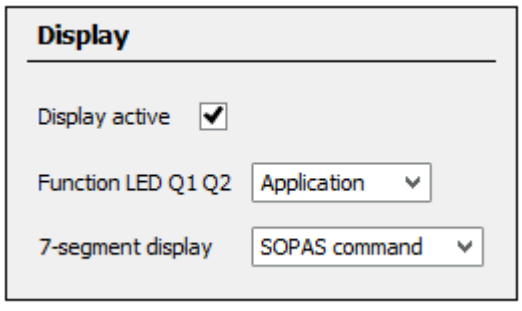
CoLa A	<STX>sAN{SPC}mLMLSetLed{SPC}0<ETX>
CoLa B	02 02 02 02 00 00 00 10 73 41 4E 20 6D 4C 4D 4C 53 65 74 4C 65 64 20 00 53

12.2.1.4.9.15 Set 7-segment display to specific symbol or number [sMN mLMLSetDisp]

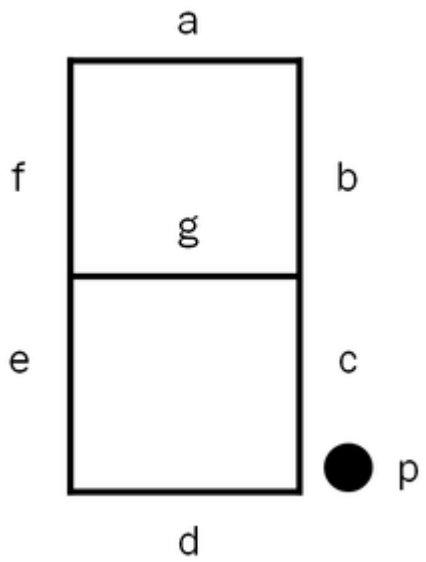
NOTE

Preconditions:

It is mandatory to define that the 7-segment display should react to SOPAS commands. This option needs to be activated via `sWN LMLfpFcn` or using the configuration software SOPAS ET. Choose “SOPAS command” in the drop-down list for the 7-segment display (as shown in the figure below). Afterwards the segments of the display can be set via telegram.



The display is consisting of the segments a-p:



The segments a-g are operated using the bits 0 to 6, the segment p using bit 7:

Segment	p	g	f	e	d	c	b	a
Related bit	7	6	5	4	3	2	1	0

Example: Showing the number “7” on the display:

Segment	p	g	f	e	d	c	b	a
On/Off for showing the required symbol	off	off	off	off	off	on	on	on
Binary	0	0	0	0	0	1	1	1

Transfer binary into ASCII:

00000111 bin ≙ 07 ASCII

Table 334: Telegram structure: sMN mLMLSetDisp

Telegram structure: sMN mLMLSetDisp						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Request (SOPAS method by name)	String	3		sMN	73 4D 4E
Command	Set 7-segment display	String	11		mLMLSetDisp	6D 4C 4D 4C 53 65 74 44 69 73 70
Display	7-segment display in the display of the LMS. The segments A-G are operated using the bits 0 to 6, the point P using bit 7.	Uint_8	1	Display off: ... Display shows 2: Display shows 7: ... Display completely on (8.):	0 ... 5Bh 7 ... FFh	00 ... 5B 07 ... FF

Table 335: Example: sMN mLMLSetDisp 07 (Showing the number “7” on the display)

CoLa A	<STX>sMN{SPC}mLMLSetDisp{SPC}07<ETX>
	<STX>sMN mLMLSetDisp 07<ETX>
	sMN mLMLSetDisp 07
	02 73 4D 4E 20 6D 4C 4D 4C 53 65 74 44 69 73 70 20 30 37 03

CoLa B	02 02 02 02 00 00 00 11 73 4D 4E 20 6D 4C 4D 4C 53 65 74 44 69 73 70 20 07 3B 73 4D 4E 20 6D 4C 4D 4C 53 65 74 44 69 73 70 20 07

Table 336: Telegram structure: sAN mLMLSetDisp

Telegram structure: sAN mLMLSetDisp						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer (SOPAS answer)	String	3		sAN	73 41 4E
Command	Set 7-segment display	String	11		mLMLSetDisp	6D 4C 4D 4C 53 65 74 44 69 73 70
ErrorCode	The command has been accepted if the error code 1 is returned.	Enum8	1	Error: No error:	0 1	00 01

Table 337: Example: sAN mLMLSetDisp 1

CoLa A	<STX>sAN{SPC}mLMLSetDisp{SPC}1<ETX>
	02 73 41 4E 20 6D 4C 4D 4C 53 65 74 44 69 73 70 20 31 03
CoLa B	02 02 02 02 00 00 00 11 73 41 4E 20 6D 4C 4D 4C 53 65 74 44 69 73 70 20 01 31

12.2.1.4.10 Application

12.2.1.4.10.1 Request status change of monitoring fields on event [sEN ECRChangeArr]

NOTE

Preconditions:

Necessary sensor setup:

- Setup detection fields.
- Setup evaluation cases and assign outputs to the evaluation cases.

Advantage of this telegram:

- Status of the evaluation case (1=field free, 2=detecting or 3=infringement) is transmitted if there is a status change of the monitoring field within the application.
- For example if the object size in the field exceeds the set object size parameter but the duration of the object inside the field is below the set time parameter. Then the status change from “field free” to “detecting” will be given out via telegram.

Necessary procedure after each power up of the device:

- Establish Ethernet connection to the device
- Activate Ethernet output

Table 338: Telegram structure: sEN ECRChangeArr

Telegram structure: sEN ECRChangeArr						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sEN	73 45 4E
Command	Data telegram	String	12		ECRChangeArr	45 43 52 43 68 61 6E 67 65 41 72 72

Telegram structure: sEN ECRChangeArr						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Reporting	Start/stop	Enum_8	1	Stop: Start:	0 1	00 01

Table 339: Example: sEN ECRChangeArr 1

CoLa A	<STX>sEN{SPC}ECRChangeArr{SPC}1<ETX>
	<STX>sEN ECRChangeArr 1<ETX>
	sEN ECRChangeArr 1
	02 73 45 4E 20 45 43 52 43 68 61 6E 67 65 41 72 72 20 31 03
CoLa B	02 02 02 02 00 00 00 12 73 45 4E 20 45 43 52 43 68 61 6E 67 65 41 72 72 20 01 4A
	73 45 4E 20 45 43 52 43 68 61 6E 67 65 41 72 72 20 01

Table 340: Telegram structure: sEA ECRChangeArr

Telegram structure: sEA ECRChangeArr						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sEA	73 45 41
Command	Data telegram	String	12		ECRChangeArr	45 43 52 43 68 61 6E 67 65 41 72 72
Reporting	Start/stop	Enum_8	1	Stop: Start:	0 1	00 01

Table 341: Example: sEA ECRChangeArr 1

CoLa A	<STX>sEA{SPC}ECRChangeArr{SPC}1<ETX>
	02 73 45 41 20 45 43 52 43 68 61 6E 67 65 41 72 72 20 31 03
CoLa B	02 02 02 02 00 00 00 12 73 45 41 20 45 43 52 43 68 61 6E 67 65 41 72 72 20 01 45

NOTE

The answer to the telegram will be followed by data that is sent on event.

The sensor only sends the following answer if there is a status change of the evaluation case within the application.

Table 342: Telegram structure: sSN ECRChangeArr

Telegram structure: sSN ECRChangeArr						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sSN	73 53 4E
Command	Data telegram	String	12		ECRChangeArr	45 43 52 43 68 61 6E 67 65 41 72 72
System counter	Time in µs since power up max. 71 min then starting from 0 again	Uint_32	4		0 ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
Array	1-10	UInt_16	2		0h ... Ah (0d ... 10d)	00 01 ... 00 0A

Telegram structure: sSN ECRChangeArr						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
EVC number	1-10	UInt_8	1		0h ... Ah (0d ... 10d)	00 01 ... 00 0A
Object detection		Enum_8	1	field free detecting infrigement	1 2 3	01 02 03
Year		Uint_16	2		0000h ... 270Fh (0d...9999d)	00 00 ... 27 0F
Month	1 to 12	Uint_8	1		00h ... 0Ch (0d ... 12d)	00 ... 0C
Day	Day of month 1 to 31	Uint_8	1		00h ... 1Fh (0d ... 31d)	00 ... 1F
Hour	0 to 23	Uint_8	1		00h ... 17h (0d ... 23d)	00 ... 17
Minute	0 to 59	Uint_8	1		00h ... 3Bh (0d ... 59d)	00 ... 3B
Second	0 to 59	Uint_8	1		00h ... 3Bh (0d ... 59d)	00 ... 3B
µSecond	0 to 9999999	Uint_32	4		00000000h ... 000F423Fh (0d ... 999999d)	00 00 00 00 ... 00 0F 42 3F

Table 343: Example: sSN ECRChangeArr 1

CoLa A	<STX>sSN{SPC}ECRChangeArr{SPC}97F8C2E3{SPC}1{SPC}1{SPC}1{SPC}7B2{SPC}1{SPC}1{SPC}0{SPC}2A{SPC}1D{SPC}63DA8<ETX>
	02 73 53 4E 20 45 43 52 43 68 61 6E 67 65 41 72 72 20 39 37 46 38 43 32 45 33 20 31 20 31 20 31 20 37 42 32 20 31 20 31 20 30 20 32 41 20 31 44 20 36 33 44 41 38 03
CoLa B	02 02 02 02 00 00 00 24 73 53 4E 20 45 43 52 43 68 61 6E 67 65 41 72 72 20 CF 09 10 99 00 01 01 02 07 B2 01 01 02 09 1C 00 03 47 D8 2E

12.2.1.4.10.2 Individual request of monitoring fields to their status changes – ECR xy [sRN ECRxy]

NOTE

The telegram “ECRxy” is available with firmware versions > V1.32 (LMS12x, LMS13x, LMS14x); > V2.00.1 (LMS11x) and > V2.00.3 (LMS16x).

Preconditions:

Necessary sensor setup:

- Setup detection fields.
- Setup evaluation cases.
- Setup output assignments to the evaluation cases.

Necessary proceeding after each Laser Scanner power up:

- Establish Ethernet connection to the device.
- If the user wants to register an event, it is necessary to send a telegram to register it each time after establishing connection.

Read/Poll the status of an evaluation case

It is possible to send (poll repeatedly) this telegram to Ethernet port 2111 or Ethernet port 2112 to check the evaluation status.

Table 344: Telegram structure: sRN ECRxy

Telegram structure: sRN ECRxy						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E
Command	Data telegram	String	12	Concrete number should be used for xy as shown below: Evaluation1: Evaluation2: ... Evaluation10:	ECRxy ECR01 ECR02 ... ECR10	 45 43 52 30 31 45 43 52 30 32 ... 45 43 52 31 30

Table 345: Example: sRN ECR01

CoLa A	<STX>sRN{SPC}ECR01<ETX>
	<STX>sRN ECR01<ETX>
	sRN ECR01
	02 73 52 4E 20 45 43 52 30 31 03
CoLa B	02 02 02 02 00 00 00 09 73 52 4E 20 45 43 52 30 31 1A
	73 52 4E 20 45 43 52 30 31

The response telegram is shown in Telegram structure: sRA/ sSN ECRxy

Register an event to get the status update of an evaluation

It is also possible to register an event by “sEN ECRxy” for receiving the change of the an evaluation. So, when there a change happens in the field of this evaluation, an update telegram will be sent out automatically.

The registration telegram format is: sEN ECRxy 0/1 (shown below)

Table 346: Telegram structure: sEN ECRxy

Telegram structure: sEN ECRxy						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sEN	73 45 4E
Command	Data telegram	String	12	Concrete number should be used for xy as shown below: Evaluation1: Evaluation2: ... Evaluation10:	ECRxy ECR01 ECR02 ... ECR10	 45 43 52 30 31 45 43 52 30 32 ... 45 43 52 31 30
Reporting	Start/stop	Enum_8	1	Stop: Start:	0 1	00 01

Table 347: Example: sEN ECR011

CoLa A	<STX>sEN{SPC}ECR01{SPC}1<ETX>
	<STX>sEN ECR011<ETX>
	sEN ECR011
	02 73 45 4E 20 45 43 52 30 31 20 31 03
CoLa B	02 02 02 02 00 00 00 0B 73 45 4E 20 45 43 52 30 31 20 01 2C
	73 45 4E 20 45 43 52 30 31 20 01

Table 348: Telegram structure: sEA ECRxy

Telegram structure: sEA ECRxy						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sEA	73 45 41
Command	Data telegram	String	12	Concrete number should be used for xy as shown below: Evaluation1: Evaluation2: ... Evaluation10:	ECRxy ECR01 ECR02 ... ECR10	 45 43 52 30 31 45 43 52 30 32 ... 45 43 52 31 30
Reporting	Start/stop	Enum_8	1	Stop: Start:	0 1	00 01

Table 349: Example: sEA ECR01 1

CoLa A	<STX>sEA{SPC}ECR01{SPC}1<ETX>
	02 73 45 41 20 45 43 52 30 31 20 31 03
CoLa B	02 02 02 02 00 00 00 0B 73 45 41 20 45 43 52 30 31 20 01 23

NOTE

- The event registration is not a parameter and cannot be saved permanently. Therefor it is necessary to send the telegram to register an event each time after establishing a connection to the device (especially, when the Ethernet port is in server mode).
- Ethernet port 2112 has a special function and can be set as client mode (as shown in Figure 1). Then, the corresponding telegram can be selected for the expected evaluation (for example evaluation 1, as shown below). There is no further need to manually register an event by sending a separate telegram. When there is a change in the registered field evaluation, LMS12x, LMS13x, LMS14x will automatically connect to the specified server (such as 192.168.0.1, as shown below) to send an update telegram.If the settings are saved permanently, this is the same situation even after the sensor power cycle.

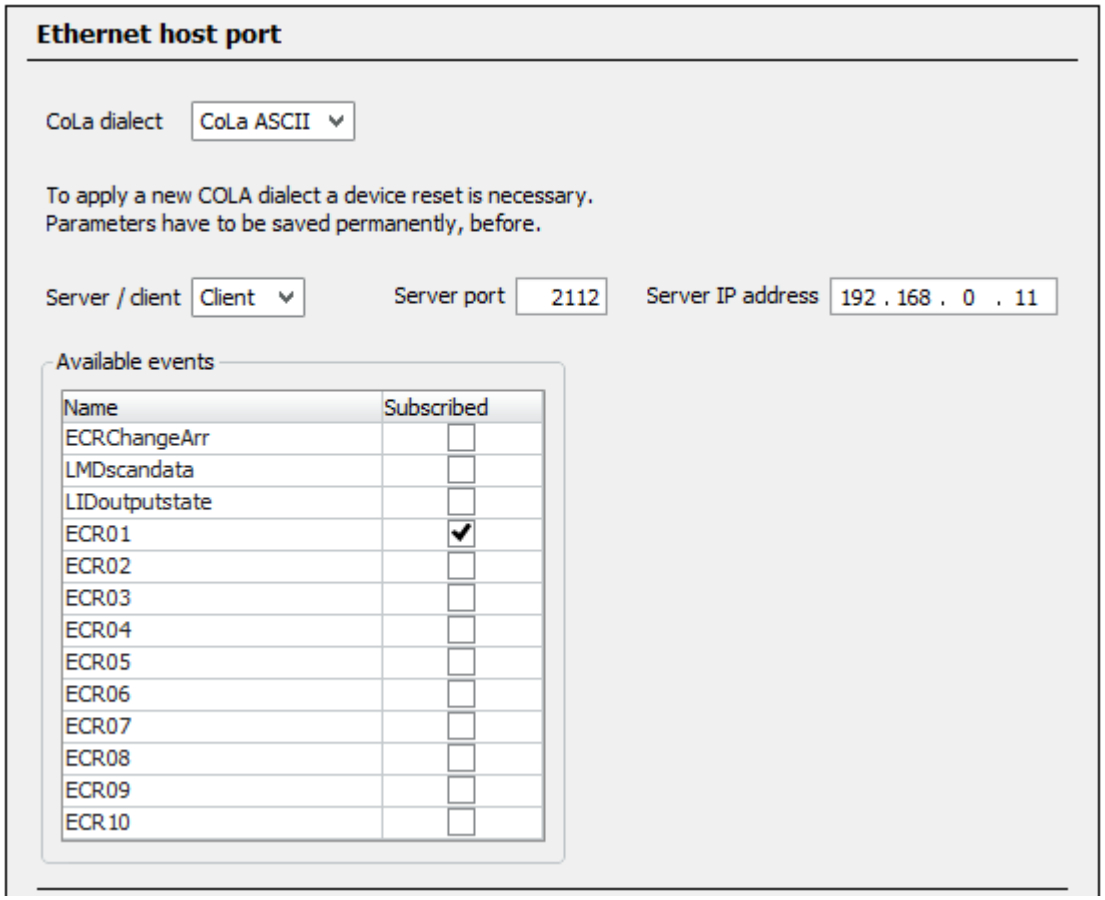


Figure 60: Ethernet host port setting (LMS12x, LMS13x, LMS14x)

The response telegram to "`sRN ECRxy`" and the event status update telegram for "`sEA ECRxy 1`" follows the format in table 7 as shown below.

Table 350: Telegram structure: sRA/sSN ECRxy

Telegram structure: sRA/sSN ECRxy						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	for sRN ECRxy: for sEN ECRxy:	sRA sSN	73 52 41 73 53 4E
Command	Data telegram	String	12	Concrete number should be used for xy as shown below: Evaluation1: Evaluation2: ... Evaluation10:	ECRxy ECR01 ECR02 ... ECR10	 45 43 52 30 31 45 43 52 30 32 ... 45 43 52 31 30
Reporting	Object detection	Enum_8	1	DON'T CARE: FIELD FREE: DETECTING: INFRINGEMENT:	0 1 2 3	00 01 02 03
Time-stamp	Transmisssion time stamp of the current telegram (unit:µs)	Uint_32	4		00000000h.... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
Year		Uint_16	2		0000h ... 270Fh (0d...9999d)	00 00 ... 27 0F

Telegram structure: sRA/sSN ECRxy						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Month	1 to 12	Uint_8	1		00h ... 0Ch (0d ... 12d)	00 ... 0C
Day	Day of month 1 to 31	Uint_8	1		00h ... 1Fh (0d ... 31d)	00 ... 1F
Hour	0 to 23	Uint_8	1		00h ... 17h (0d ... 23d)	00 ... 17
Minute	0 to 59	Uint_8	1		00h ... 3Bh (0d ... 59d)	00 ... 3B
Second	0 to 59	Uint_8	1		00h ... 3Bh (0d ... 59d)	00 ... 3B
µSecond	0 to 9999999	Uint_32	4		00000000h ... 000F423Fh (0d ... 999999d)	00 00 00 00 ... 00 0F 42 3F

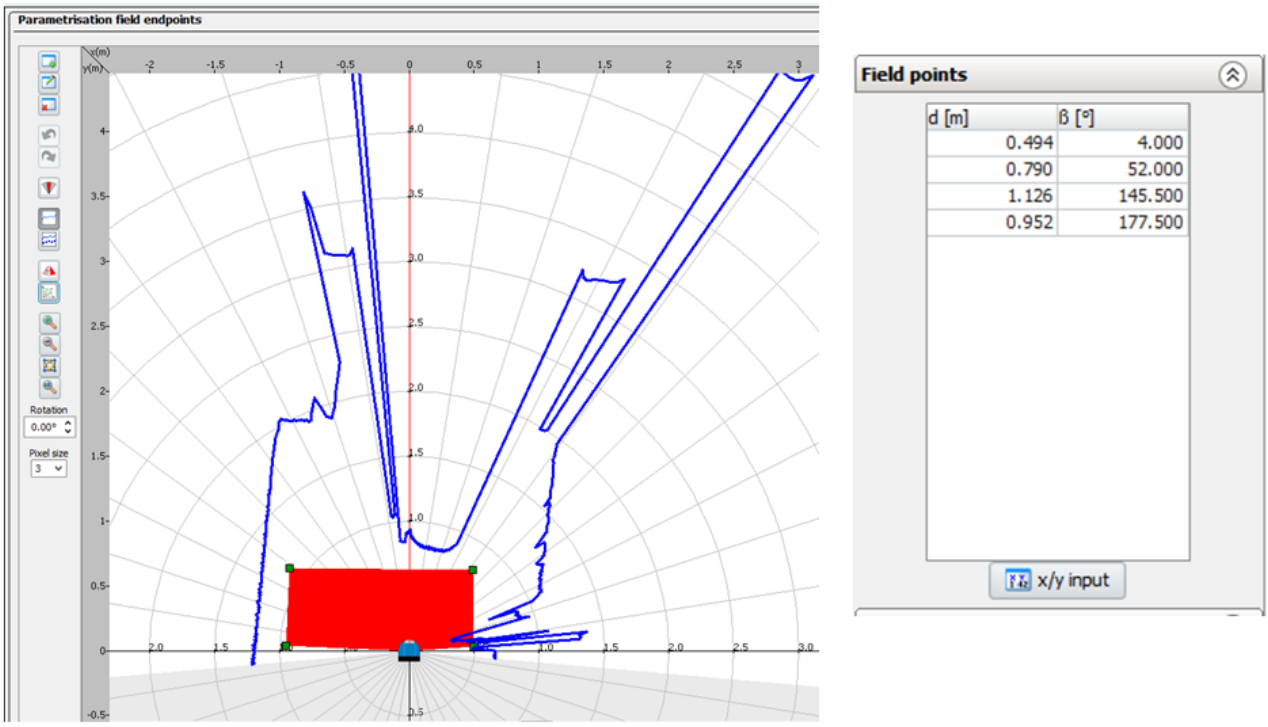
Table 351: Example: sRA ECR01

CoLa A	<STX>sRA{SPC}ECR01{SPC}1{SPC}684E451D 7B2 1 1 0 1D 9 D7938<ETX>
CoLa B	02 02 02 02 00 00 00 1A 73 53 4E 20 45 43 52 30 31 20 01 68 4E 45 1D 07 B2 01 01 00 1D 09 00 0D 79 38 A9

12.2.1.4.10.3 Request SOPAS field data structure [sMN mLFEgetField]

The SOPAS telegram `mLFEgetField` requests a field number as parameter and returns the corresponding SOPAS field data structure. If the field number is number is not configured, the answer telegram will be filled with 0.

Example – Request the field data structure of an evaluation field (field number: 1) that has been parameterized in the engineering tool SOPAS.



The answer telegram will include information regarding the field data structure of the requested evaluation field (type: segmented, number of field points etc.)

Please note: The sensor will switch to the state “Stop measurement” during read out. After the read out you have to switch the sensor back to “Run measurement”.

Table 352: Telegram structure: sMN mLFEgetField

Telegram structure: sMN mLFEgetField (User level 'Authorized Client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Request	String	3		sMN	73 4D 4E
Command	Only one telegram	String	14		mLFEgetField	6D 4C 46 45 67 65 74 46 69 65 6C 64
Command		UInt_32	4		1 ... 10	00 00 00 01 ... 00 00 01 01

Table 353: Example: sMN mLFEgetField 1

CoLa A	<STX>sMN{SPC}mLFEgetField{SPC}1<ETX>	
	<STX>sMN mLFEgetField 1<ETX>	
	sMN mLFEgetField 1	
	02 73 4D 4E 20 6D 4C 46 45 67 65 74 46 69 65 6C 64 20 31 03	
CoLa B	02 02 02 02 00 00 00 15 73 4D 4E 20 6D 4C 46 45 67 65 74 46 69 65 6C 64 20 00 00 00 01 67	
	73 4D 4E 20 6D 4C 46 45 67 65 74 46 69 65 6C 64 20 00 00 00 01	

Table 354: Example: sAN mLFEgetField

Telegram structure: sAN mLFEgetField						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (binary)**
Read		String	3		sAN	73 41 4E
Command	Only one telegram	String	14		mLFEgetField	6D 4C 46 45 67 65 74 46 69 65 6C 64
Field header						
Distance Scale Factor	Scale factor or factor of the measurement values	Real as float according to IEEE754	4	Factor x 1: Factor x 2:	3F800000h 40000000h	3F 80 00 00 40 00 00 00
Distance Scale Offset	Sets starting point of measurement	Real as float according to IEEE754	4	No offset:	00000000	00 00 00 00
Angle Scale Factor	Angle resolution	Uint_32	4	0.25°: 0.5°:	+2500d (9C4h) +5000d (1388h)	00 00 09 C4 00 00 13 88
Angle Scale Offset	Starting angle	Int_32	4		-450000d ... +2250000d (FFF92230h ... 225510h)	FF F9 22 30 ... 00 22 55 10
Field type		Enum_8	1	Radial: Rectangle: Segmented: Dynamic:	0 1 2 3	00 01 02 03
Field number		Uint_8	1		0 ... Ah	00 ... 0A

Telegram structure: sAN mLFEgetField						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (binary)**
Segmented field						
Segmented field configured		Uint_16	2	No data for segmented field available, i.e. field type is not segmented field: Data for segmented field available:	0 1	00 00 00 01
Only shown if field type is configured	Number of field points	Uint_16	2	0 ... 571:	0h ... 23Bh	00 00 ... 02 3B
	Angle index n	Uint_16	2	0 ... 1081:	0h ... 0439h	00 00 ... 04 39
	Start distance	Uint_16	2	0 ... 65535:	0h ... FFFFh	00 00 ... FF FF
	End distance	Uint_16	2	0 ... 65535:	0h ... FFFFh	00 00 ... FF FF
	Angle index n+1	...	...	...	...	...
	Start distance	...	...	...	...	...
	End distance	...	...	...	...	...
	...	...	...	...	...	...
Rectangular field						
Rectangular field configured		Uint_16	2	No data for rectangular field available, i.e. field type is not rectangular field: Data for rectangular field available:	0 1	00 00 00 01
Only shown if field type is configured	Angle of reference point	Int_32	4		-450000d ... +2250000d (FFF92230h ... 225510h)	FF F9 22 30 ... 00 22 55 10
	Distance of reference point	Uint_16	2	0 ... 65535:	0h ... FFFFh	00 00 ... FF FF
	Rotation angle	Int_32	4		-1800000d ... 1800000d (FFE488C0h ... 001B7740h)	FF E4 88 C0 ... 00 1B 77 40
	Width	Uint_32	4		0 ... 4294967295d (0 ... FFFFFFFFh)	00 00 00 00 ... FF FF FF FF
	Length	Uint_32	4		0 ... 4294967295d (0 ... FFFFFFFFh)	00 00 00 00 ... FF FF FF FF
Radial field						
Radial field configured		Uint_16	2	Radial fields are not available	0	00 00
Dynamic field						
Dynamic field configured		Uint_16	2	No data for dynamic field available, i.e. field type is not dynamic field Data for dynamic field available	0 1	00 00 00 01

Telegram structure: sAN mLFEgetField						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (binary)**
Only shown if field type is config-ured	Angle of reference point	Int_32			-450000d ... +2250000d (FFF92230h ... 225510h)	FF F9 22 30 ... 00 22 55 10
	Distance of reference point	Uint_16	2	0 ... 65535:	0h ... FFFFh	00 00 ... FF FF
	Rotation angle	Int_32	4		-1800000d ... 1800000d (FFE488C0h ... 001B7740h)	FF E4 88 C0 ... 00 1B 77 40
	Width	Uint_32	4		0 ... 4294967295d (0 ... FFFFFFFFh)	00 00 00 00 ... FF FF FF FF
	Length	Uint_32	4		0 ... 4294967295d (0 ... FFFFFFFFh)	00 00 00 00 ... FF FF FF FF
	Maximum speed	Int_16	2		0 ... 30000d (0 ... 7530h)	00 00 ... 75 30
	Maximum length	Uint_32	4		0 ... 4294967295d	00 00 00 00 ... FF FF FF FF
				General		
Version number		Uint_16	2		0 ... 65535	00 00 ... FF FF
Length of field name		Uint_16	2		0 ... 32	00 00 ... 00 20
Field name		String	0 ... 32		FIELD1	46 49 45 4C 44 31
Length of comment	If no comment, also no length with a string type ...	Uint_16	2		0 ... 128d (0 ... 80h)	00 00 ... 00 80
Comment		String	0 ... 128			

Table 355: Example: sAN mLFEgetField

CoLa A	<STX>sAN{SPC}mLFEget-Field{SPC}3F800000{SPC}00000000{SPC}9C4{SPC}FFF92230{SPC}1{SPC}1{SPC}0{SPC}1{SPC}1B7740{SPC}3E8{SPC}0{SPC}7D0{SPC}3E8{SPC}0{SPC}0{SPC}1{SPC}6{SPC}FIELD1{SPC}0<ETX>
	02 73 41 4E 20 6D 4C 46 45 67 65 74 46 69 65 6C 64 20 33 46 38 30 30 30 30 30 20 30 30 30 30 30 30 30 30 20 39 43 34 20 46 46 46 39 32 32 33 30 20 31 20 31 20 30 20 31 20 31 42 37 37 34 30 20 33 45 38 20 30 20 37 44 30 20 33 45 38 20 30 20 30 20 31 20 36 20 46 49 45 4C 44 31 20 30 03
CoLa B	02 02 02 02 00 00 00 51 73 41 4E 20 6D 4C 46 45 67 65 74 46 69 65 6C 64 20 3F 80 00 00 00 00 00 00 00 00 00 09 C4 FF F9 22 30 01 01 00 00 00 01 00 1B 77 40 03 E8 00 00 00 00 00 00 07 D0 00 00 03 E8 00 00 00 00 00 01 00 06 46 49 45 4C 44 31 00 00 E8

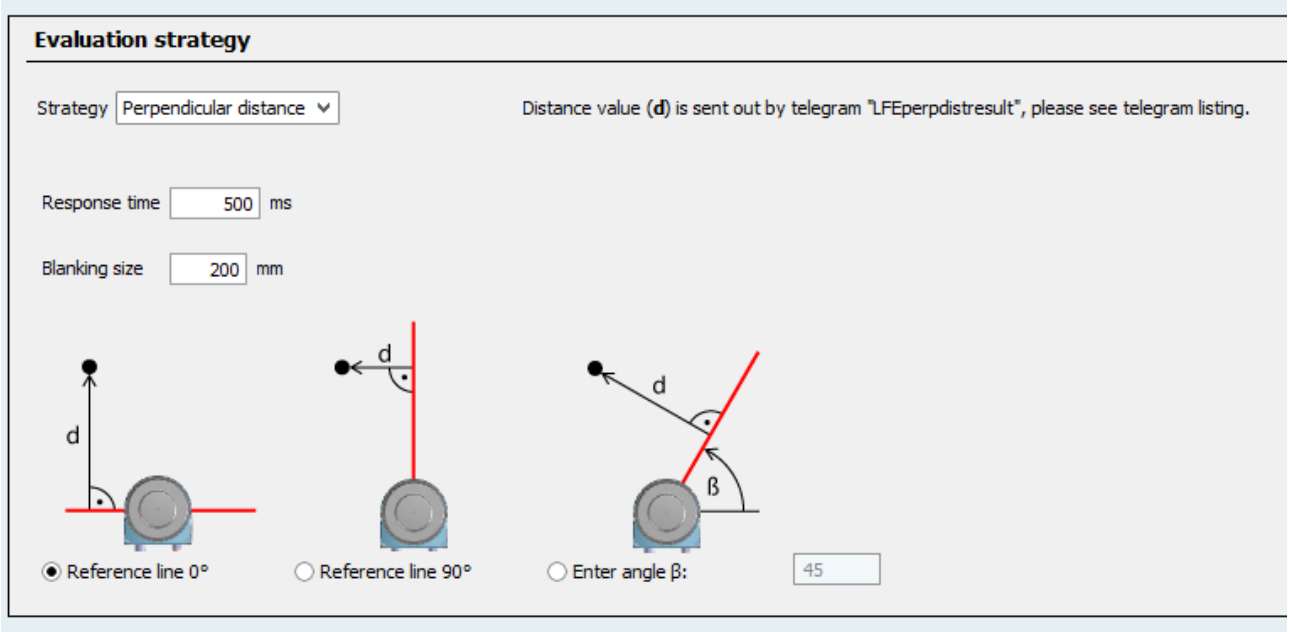
12.2.1.4.10.4 Request minimal and maximal perpendicular distance once [sRN LFEperpdistre-sult]

Set the EVC to evaluation strategy “Min. perpendicular distance” or “Max. perpendicular distance” and choose the field which you would like to read out. The field evalutation will only be activated by choosing an output.

NOTE

Precondition:

Either evaluation strategy "Minimal perpendicular distance" or "Maximal perpendicular distance" has to be activated in the engineering tool SOPAS. After the according perpendicular distance value is displayed in SOPAS (as shown below), you are able to read out the telegram.



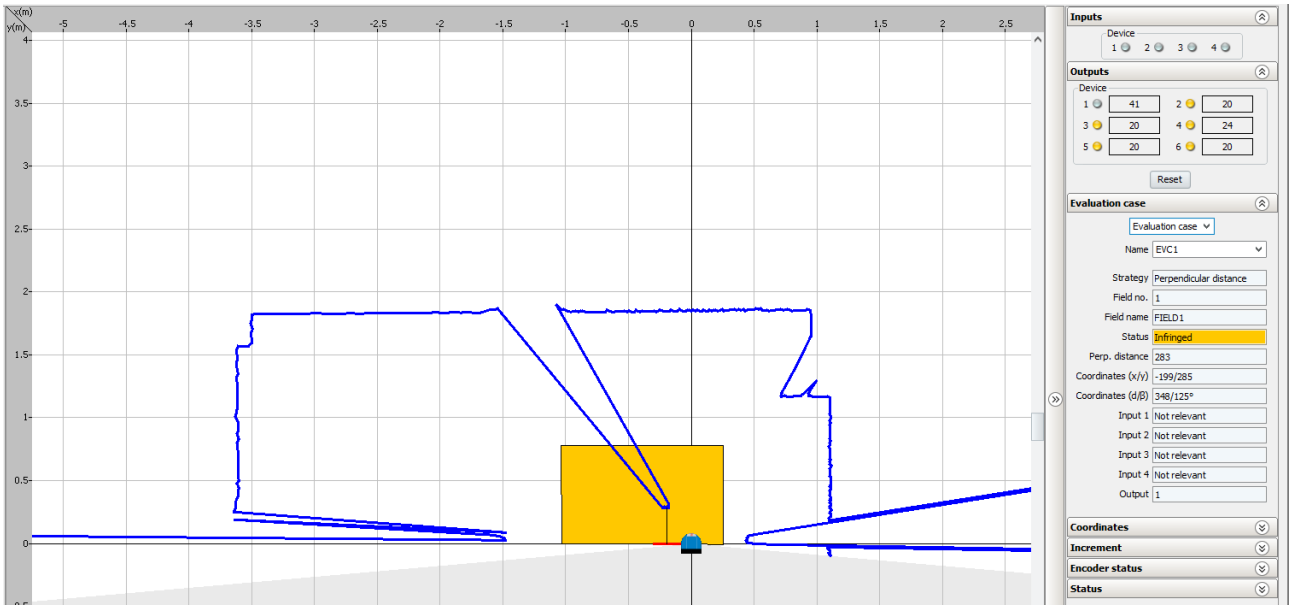


Table 356: Telegram structure: sRN LFEperpdistresult

Telegram structure: sRN LFEperpdistresult (no required user level)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E

Telegram structure: sRN LFEperpdistresult (no required user level)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command	Only one telegram	String	14		LFEperpdistre-sult	4C 46 45 70 65 72 70 64 69 73 74 72 65 73 75 6C 74

Table 357: Example: sRN LFEperpdistresult

CoLa A	<STX>sRN{SPC}LFEperpdistresult<ETX>
	<STX>sRN LFEperpdistresult<ETX>
	sRN LFEperpdistresult
	02 73 52 4E 20 4C 46 45 70 65 72 70 64 69 73 74 72 65 73 75 6C 74 03
CoLa B	02 02 02 02 00 00 00 15 73 52 4E 20 4C 46 45 70 65 72 70 64 69 73 74 72 65 73 75 6C 74 14
	73 52 4E 20 4C 46 45 70 65 72 70 64 69 73 74 72 65 73 75 6C 74

Table 358: Telegram structure: sRA LFEperpdistresult

Telegram structure: sRA LFEperpdistresult (no required user level)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRA	73 52 41
Command	Only one telegram	String	14		LFEperpdistre-sult	4C 46 45 70 65 72 70 64 69 73 74 72 65 73 75 6C 74
Array	1-10	UInt_16	2		0 ... 10d (0 ... Ah)	00 00 ... 00 0A
EVC number	1-10	UInt_8	1		0 ... 10d (0 ... Ah)	00 ... 0A
Perpendicular distance	0 m ... 20 m 0 m ... 40 m 0 m ... 50 m	UInt_32	4	LMS10x, LMS11x, LMS12x, LMS13x: LMS14x: LMS15x, LMS16x:	0 ... 20000d (0 ... 4E20h) 0 ... 40000d (0 ... 9C40h) 0 ... 50000d (0 ... C350h)	00 00 00 00 ... 00 00 4E 20 00 00 00 00 ... 00 00 9C 40 00 00 00 00 ... 00 00 C3 50
Reserved	default 0	UInt_32	4		0	00 00 00 00
Reserved	default 0	UInt_32	4		0	00 00 00 00
X-Pos [mm]	-20 m ... 20 m -40 m ... 40 m -50 m ... 50 m	Int_32	4	LMS10x, LMS11x, LMS12x, LMS13x: LMS14x: LMS15x, LMS16x:	-20000d ... 20000d (FFFFB1E0h ... 4E20h) -40000d ... 40000d (FFFF63C0h... 9C40h) -50000d ... 50000d (FFFF3CB0h... C350h)	FF FF B1 E0 ... 00 00 4E 20 FF FF 63 C0 ... 00 00 9C 40 FF FF 3C B0 ... 00 00 C3 50

Telegram structure: sRA LFEperpdistresult (no required user level)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Y-Pos [mm]	-20 m ... 20 m -40 m ... 40 m -50 m ... 50 m	Int_32	4	LMS10x, LMS11x, LMS12x, LMS13x: LMS14x: LMS15x, LMS16x:	-20000d ... 20000d (FFFFB1E0h ... 4E20h) -40000d ... 40000d (FFFF63C0h... 9C40h) -50000d ... 50000d (FFFF3CB0h... C350h)	FF FF B1 E0 ... 00 00 4E 20 FF FF 63 C0 ... 00 00 9C 40 FF FF 3C B0 ... 00 00 C3 50
Reserved	default 0	Int_32	4		0	00 00 00 00
Radial distance	0 m ... 20 m 0 m ... 40 m 0 m ... 50 m	UInt_32	4	LMS10x, LMS11x, LMS12x, LMS13x: LMS14x: LMS15x, LMS16x:	0 ... 20000d (0 ... 4E20h) 0 ... 40000d (0 ... 9C40h) 0 ... 50000d (0 ... C350h)	00 00 00 00 ... 00 00 4E 20 00 00 00 00 ... 00 00 9C 40 00 00 00 00 ... 00 00 C3 50
Radial distance angle	-45°... 225°	Int_32	4		-450000d ... +2250000d (FFF92230h ... 225510h)	FF F9 22 30 ... 00 22 55 10
Reserved	default 0	Int_32	4		0	00 00 00 00
Reference line angle	-45°... 225°	Int_32	4		-450000d ... +2250000d (FFF92230h ... 225510h)	FF F9 22 30 ... 00 22 55 10
Reserved	default 0	Int_32	4		0	00 00 00 00
Reserved	default 0	Int_32	4		0	00 00 00 00
Reserved	default 0	Int_32	4		0	00 00 00 00
Reserved	default 0	Int_32	4		0	00 00 00 00
Reserved	default 0	Int_32	4		0	00 00 00 00
Reserved	default 0	Int_32	4		0	00 00 00 00
Reserved	default 0	Int_32	4		0	00 00 00 00
Reserved	default 0	Int_32	4		0	00 00 00 00

Table 359: Example: sRA LFEperpdistresult

CoLa A	<STX>sRA{SPC} LFEperpdistresult{SPC}1{SPC}2{SPC}21A{SPC}0{SPC}0{SPC}0{SPC}21A{SPC}0{SPC}21A{SPC}DBBA0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0<ETX>
	02 73 41 4E 20 6D 4C 46 45 67 65 74 46 69 65 6C 64 20 34 30 30 30 30 30 30 30 20 30 30 30 30 30 30 30 30 20 31 33 38 38 20 46 46 46 46 33 43 42 30 20 32 20 31 20 31 20 34 20 31 32 20 46 46 46 46 20 46 37 20 37 32 20 46 46 46 46 20 31 38 42 20 31 32 44 20 46 46 46 46 20 32 33 33 20 31 36 44 20 46 46 46 46 20 31 44 43 20 30 20 30 20 30 20 31 20 36 20 46 49 45 4C 44 31 20 30 03
CoLa B	02 02 02 02 00 00 00 61 73 52 41 20 4C 46 45 70 65 72 70 64 69 73 74 72 65 73 75 6C 74 20 00 01 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 38

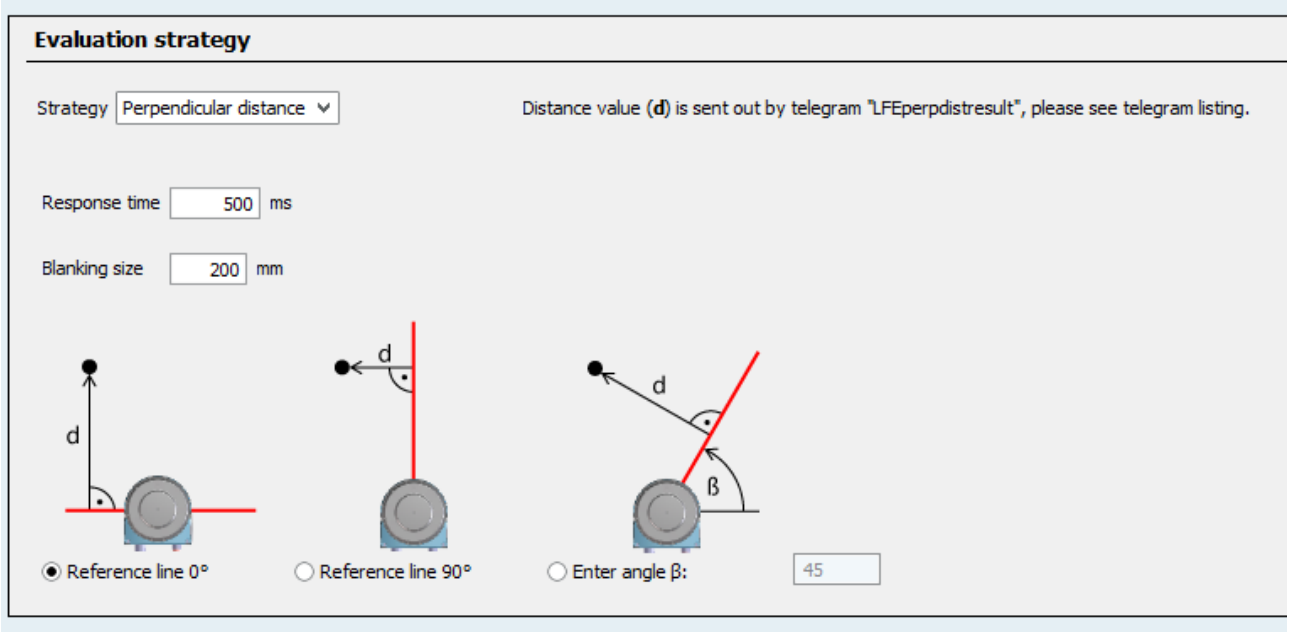
12.2.1.4.10.5 Request minimal and maximal perpendicular distance continiously on event [sEN LFEperpdistresult]

Set the EVC to evaluation strategy “Min. perpendicular distance” or “Max. perpendicular distance” and choose the field which you would like to read out. The field evalutation will only be activated by choosing an output.

NOTE

Precondition:

Either evaluation strategy "Minimal perpendicular distance" or "Maximal perpendicular distance" has to be activated in the engineering tool SOPAS. After the according perpendicular distance value is displayed in SOPAS (as shown below), you are able to read out the telegram.



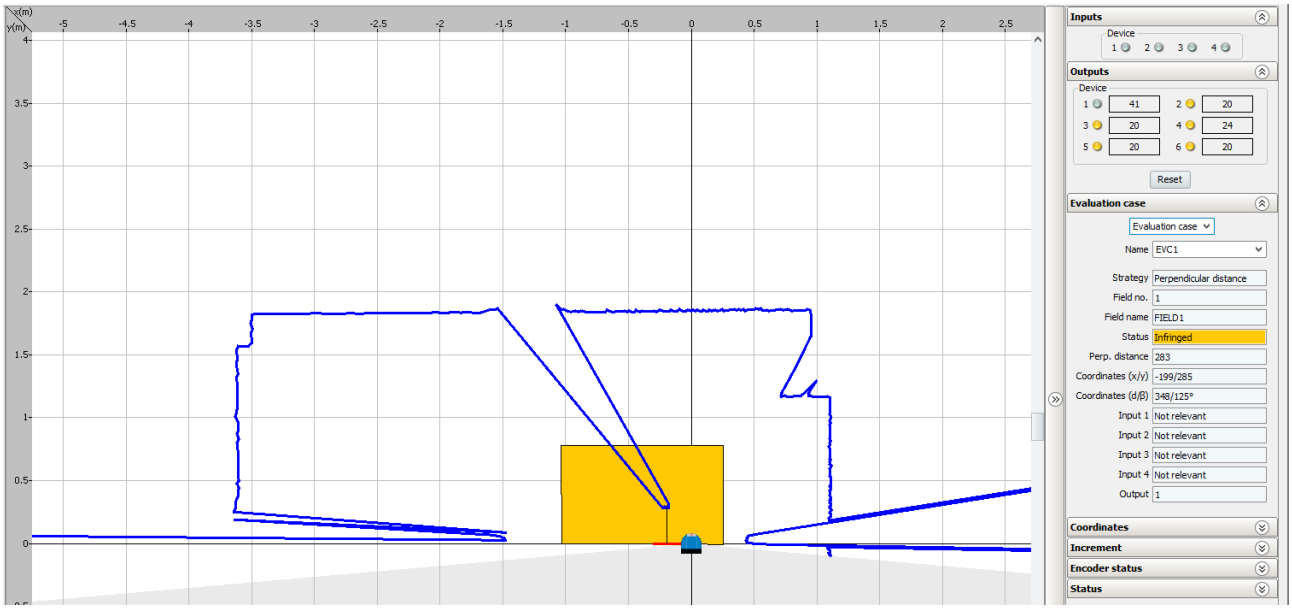


Table 360: Telegram structure: sEN LFEperpdistresult

Telegram structure: sEN LFEperpdistresult						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sEN	73 45 4E

Telegram structure: sEN LFEperpdistresult						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command	Only one telegram	String	14		LFEperpdistre-sult	4C 46 45 70 65 72 70 64 69 73 74 72 65 73 75 6C 74
Reporting	Start/stop	Enum_8	1	Stop: Start:	0 1	00 01

Table 361: Example: sEN LFEperpdistresult 1

CoLa A	<STX>sEN{SPC}LFEperpdistresult{SPC}1<ETX>
	<STX>sEN LFEperpdistresult 1<ETX>
	sEN LFEperpdistresult 1
	02 73 45 4E 20 4C 46 45 70 65 72 70 64 69 73 74 72 65 73 75 6C 74 20 31 03
CoLa B	02 02 02 02 00 00 00 17 73 45 4E 20 4C 46 45 70 65 72 70 64 69 73 74 72 65 73 75 6C 74 20 01 22
	73 45 4E 20 4C 46 45 70 65 72 70 64 69 73 74 72 65 73 75 6C 74 20 01

Table 362: Telegram structure: sEA LFEperpdistresult

Telegram structure: sEA LFEperpdistresult						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sEA	73 45 41
Command	Only one telegram	String	14		LFEperpdistre-sult	4C 46 45 70 65 72 70 64 69 73 74 72 65 73 75 6C 74
Reporting	Start/stop	Enum_8	1	Stop: Start:	0 1	00 01

Table 363: Example: sEA LFEperpdistresult 1

CoLa A	<STX>sEA{SPC}LFEperpdistresult{SPC}1<ETX>
	02 73 45 41 20 4C 46 45 70 65 72 70 64 69 73 74 72 65 73 75 6C 74 20 31 03
CoLa B	02 02 02 02 00 00 00 17 73 45 41 20 4C 46 45 70 65 72 70 64 69 73 74 72 65 73 75 6C 74 20 01 2D

NOTE

The answer to the telegram will be followed by data that is sent on event.

The sensor only sends the following answer if there are perpendicular distance values calculated within the application.

Table 364: Telegram structure: sSN LFEperpdistresult

Telegram structure: sSN LFEperpdistresult						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sSN	73 53 4E
Command	Only one telegram	String	14		LFEperpdistre-sult	4C 46 45 70 65 72 70 64 69 73 74 72 65 73 75 6C 74

Telegram structure: sSN LFEperpdistresult						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Array	1-10	UInt_16	2		0 ... 10d (0 ... Ah)	00 00 ... 00 0A
EVC number	1-10	UInt_8	1		0 ... 10d (0 ... Ah)	00 ... 0A
Perpendicular distance	0 m ... 20 m 0 m ... 40 m 0 m ... 50 m	UInt_32	4	LMS10x, LMS11x, LMS12x, LMS13x: LMS14x: LMS15x, LMS16x:	0 ... 20000d (0 ... 4E20h) 0 ... 40000d (0 ... 9C40h) 0 ... 50000d (0 ... C350h)	00 00 00 00 ... 00 00 4E 20 00 00 00 00 ... 00 00 9C 40 00 00 00 00 ... 00 00 C3 50
Reserved	default 0	UInt_32	4		0	00 00 00 00
Reserved	default 0	UInt_32	4		0	00 00 00 00
X-Pos [mm]	-20 m ... 20 m -40 m ... 40 m -50 m ... 50 m	Int_32	4	LMS10x, LMS11x, LMS12x, LMS13x: LMS14x: LMS15x, LMS16x:	-20000d ... 20000d (FFFFB1E0h ... 4E20h) -40000d ... 40000d (FFFF63C0h... 9C40h) -50000d ... 50000d (FFFF3CB0h ... C350h)	FF FF B1 E0 ... 00 00 4E 20 FF FF 63 C0 ... 00 00 9C 40 FF FF 3C B0 ... 00 00 C3 50
Y-Pos [mm]	-20 m ... 20 m -40 m ... 40 m -50 m ... 50 m	Int_32	4	LMS10x, LMS11x, LMS12x, LMS13x: LMS14x: LMS15x, LMS16x:	-20000d ... 20000d (FFFFB1E0h ... 4E20h) -40000d ... 40000d (FFFF63C0h... 9C40h) -50000d ... 50000d (FFFF3CB0h... C350h)	FF FF B1 E0 ... 00 00 4E 20 FF FF 63 C0 ... 00 00 9C 40 FF FF 3C B0 ... 00 00 C3 50
Reserved	default 0	Int_32	4		0	00 00 00 00
Radial distance	0 m ... 20 m 0 m ... 40 m 0 m ... 50 m	UInt_32	4	LMS10x, LMS11x, LMS12x, LMS13x: LMS14x: LMS15x, LMS16x:	0 ... 20000d (0 ... 4E20h) 0 ... 40000d (0 ... 9C40h) 0 ... 50000d (0 ... C350h)	00 00 00 00 ... 00 00 4E 20 00 00 00 00 ... 00 00 9C 40 00 00 00 00 ... 00 00 C3 50
Radial distance angle	-45°... 225°	Int_32	4		-450000d ... +2250000d (FFF92230h ... 225510h)	FF F9 22 30 ... 00 22 55 10
Reserved	default 0	Int_32	4		0	00 00 00 00
Reference line angle	-45°... 225°	Int_32	4		-450000d ... +2250000d (FFF92230h ... 225510h)	FF F9 22 30 ... 00 22 55 10
Reserved	default 0	Int_32	4		0	00 00 00 00
Reserved	default 0	Int_32	4		0	00 00 00 00
Reserved	default 0	Int_32	4		0	00 00 00 00
Reserved	default 0	Int_32	4		0	00 00 00 00
Reserved	default 0	Int_32	4		0	00 00 00 00

Telegram structure: sSN LFEperpdistresult						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Reserved	default 0	Int_32	4		0	00 00 00 00
Reserved	default 0	Int_32	4		0	00 00 00 00
Reserved	default 0	Int_32	4		0	00 00 00 00

Table 365: Example: sSN LFEperpdistresult

CoLa A	<STX>sSN{SPC}LFEperpdistresult{SPC}1{SPC}2{SPC}B0{SPC}0{SPC}0{SPC}FFFFFFE8{SPC}B0{SPC}0{SPC}B2{SPC}EF420{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0<ETX>
	02 73 53 4E 20 4C 46 45 70 65 72 70 64 69 73 74 72 65 73 75 6C 74 20 31 20 32 20 42 30 20 30 20 30 20 46 46 46 46 46 46 45 38 20 42 30 20 30 20 42 32 20 45 46 34 32 30 20 30 20 30 20 30 20 30 20 30 20 30 20 30 20 30 20 30 20 30 03
CoLa B	02 02 02 02 00 00 00 61 73 53 4E 20 4C 46 45 70 65 72 70 64 69 73 74 72 65 73 75 6C 74 20 00 01 01 00 00 01 5E 00 00 00 00 00 00 00 00 00 00 00 4A 00 00 01 5E 00 00 00 00 00 00 01 66 00 0B E6 E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 15

12.2.1.4.10.6 Request latest field infringement info [sRN LFEinfringementinfo]

This command is used to request entries from the “field infringement logging” via telegram. Using this command, only the latest field infringement is given out. The answer telegram includes information regarding date and time of the infringement as well as the associated EVC and infringement position.

NOTE

Only activated EVCs will be considered. An EVC is only valid, when an output is defined in the EVC. Using this telegram, an EVC without a defined output is not taken into considaration.

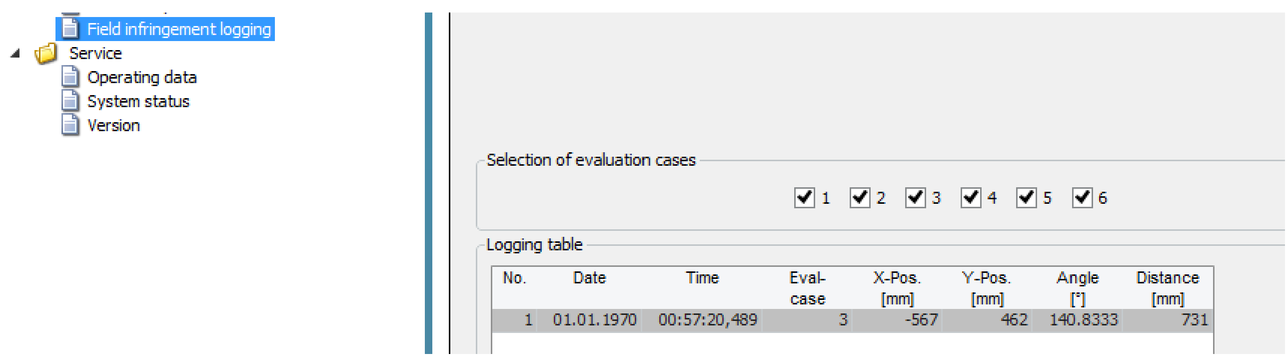


Table 366: Telegram structure: sRN LFEinfringementinfo

Telegram structure: sRN LFEinfringementinfo						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E
Command	Only one telegram	String	19		LFEinfringe-mentinfo	4C 46 45 69 6E 66 72 69 6E 67 65 6D 65 6E 74 69 6E 66 6F

Table 367: Example: sEN LFEinfringementinfo

CoLa A	<STX>sRN{SPC}LFEinfringementinfo<ETX>	
	<STX>sRN LFEinfringementinfo<ETX>	
	sRN LFEinfringementinfo	
	02 73 52 4E 20 4C 46 45 69 6E 66 72 69 6E 67 65 6D 65 6E 74 69 6E 66 6F 03	
CoLa B	02 02 02 02 00 00 00 17 73 52 4E 20 4C 46 45 69 6E 66 72 69 6E 67 65 6D 65 6E 74 69 6E 66 6F 0A	
	73 52 4E 20 4C 46 45 69 6E 66 72 69 6E 67 65 6D 65 6E 74 69 6E 66 6F	

NOTE

The answer telegram refers to the latest recorded field infringement. Therefore the device outputs the latest field infringement entry from the logging table.

Table 368: Telegram structure: sRA LFEinfringementinfo

Telegram structure: sRA LFEinfringementinfo						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRA	73 52 4E
Command	Only one telegram	String	19		LFEinfringe-mentinfo	4C 46 45 69 6E 66 72 69 6E 67 65 6D 65 6E 74 69 6E 66 6F
Time info						
Counter	continious counter of infrigements	Uint_16	2		0 ... 9999d (0 ... 270Fh)	00 00 ... 27 0F
Year		Uint_16	2		0 ... 9999d (0 ... 270Fh)	00 00 ... 27 0F
Month	1 to 12	Uint_8	1		0 ... 12d (0 ... Ch)	00 ... 0C
Day	Day of month 1 to 31	Uint_8	1		0 ... 31d (0 ... 1Fh)	00 ... 1F
Hour	0 to 23	Uint_8	1		0 ... 23d (0 ... 17h)	00 ... 17
Minute	0 to 59	Uint_8	1		0 ... 59d (0 ... 3Bh)	00 ... 3B
Second	0 to 59	Uint_8	1		0 ... 59d (0 ... 3Bh)	00 ... 3B
µSecond	0 to 999999	Uint_32	4		0 ... 999999d (0 ... 000F423Fh)	00 00 00 00 ... 00 0F 42 3F
Infringement-info						
EVC number	1-10	Uint_8	1		0 ... 10d (0 ... Ah)	00 ... 0A
X-Pos [mm]	-20 m ... 20 m -40 m ... 40 m -50 m ... 50 m	Int_32	4	LMS10x, LMS11x, LMS12x, LMS13x: LMS14x: LMS15x, LMS16x:	-20000d ... 20000d (FFFFB1E0h ... 4E20h) -40000d ... 40000d (FFFF63C0h... 9C40h) -50000d ... 50000d (FFFF3CB0h... C350h)	FF FF B1 E0 ... 00 00 4E 20 FF FF 63 C0 ... 00 00 9C 40 FF FF 3C B0 ... 00 00 C3 50

Telegram structure: sRA LFEinfringementinfo						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Y-Pos [mm]	-20 m ... 20 m -40 m ... 40 m -50 m ... 50 m	Int_32	4	LMS10x, LMS11x, LMS12x, LMS13x: LMS14x: LMS15x, LMS16x:	-20000d ... 20000d (FFFFB1E0h ... 4E20h) -40000d ... 40000d (FFFF63C0h... 9C40h) -50000d ... 50000d (FFFF3CB0h... C350h)	FF FF B1 E0 ... 00 00 4E 20 FF FF 63 C0 ... 00 00 9C 40 FF FF 3C B0 ... 00 00 C3 50
Distance [mm]	0 ... 20 m 0 ... 40 m 0 ... 50 m	Uint_32	4	LMS10x, LMS11x, LMS12x, LMS13x: LMS14x: LMS15x, LMS16x:	0 ... 20000d (0h ...4E20h) 0 ... 40000d (0 ... 9C40h) 0 ... 50000d (0h...C350h)	00 00 00 00 ... 00 00 4E 20 00 00 00 00 ... 00 00 9C 40 00 00 00 00 ... 00 00 C3 50
Angle [°]	-45° ... 225°	Int_32	4		-450000d ... +2250000d (FFF92230h ... 225510h)	FF F9 22 30 ... 00 22 55 10
Angle of Layer [°]	-45° ... 225°	Int_32	4		-450000d ... +2250000d (FFF92230h ... 225510h)	FF F9 22 30 ... 00 22 55 10
Object size [mm]	0 ... 20 m 0 ... 40 m 0 ... 50 m	Uint_32	4	LMS10x, LMS11x, LMS12x, LMS13x: LMS14x: LMS15x, LMS16x:	0 ... 20000d (0h ...4E20h) 0 ... 40000d (0 ... 9C40h) 0 ... 50000d (0h...C350h)	00 00 00 00 ... 00 00 4E 20 00 00 00 00 ... 00 00 9C 40 00 00 00 00 ... 00 00 C3 50

Table 369: Example: sRA LFEinfringementinfo

CoLa A	<STX>sRA{SPC}LFEinfringemen- tinfo{SPC}4{SPC}7B2{SPC}1{SPC}1{SPC}2{SPC}32{SPC}1F{SPC}3A1B0{SPC}1{SPC}FFFFFDAE{SPC}D6{SPC}277{SPC}18 7083{SPC}4A<ETX>
	02 73 52 41 20 4C 46 45 69 6E 66 72 69 6E 67 65 6D 65 6E 74 69 6E 66 6F 20 34 20 37 42 32 20 31 20 31 20 32 20 33 32 20 31 46 20 33 41 31 42 30 20 31 20 46 46 46 46 46 44 41 45 20 44 36 20 32 37 37 20 31 38 37 30 38 33 20 34 41 03
CoLa B	02 02 02 02 00 00 00 3A 73 52 41 20 4C 46 45 69 6E 66 72 69 6E 67 65 6D 65 6E 74 69 6E 66 6F 20 00 04 07 B2 01 01 02 32 1F 00 03 A1 B0 01 FF FF FD AE 00 00 00 D6 00 00 02 77 00 18 70 83 00 00 00 4A F9

12.2.1.4.10.7 Request field infringement info continiously on event [sEN LFEinfringementinfo]

The command is used to request entries from the “field infringement logging” via telegram. Using this command, a telegram will be sent from the sensor on event (in case of a new field infringement). The answer telegram includes information regarding date and time of the infringement as well as the associated EVC and infringement position.

NOTE
Only activated EVCs will be considered. An EVC is only valid, when an output is defined in the EVC. Using this telegram, an EVC without a defined output is not taken into considaration.

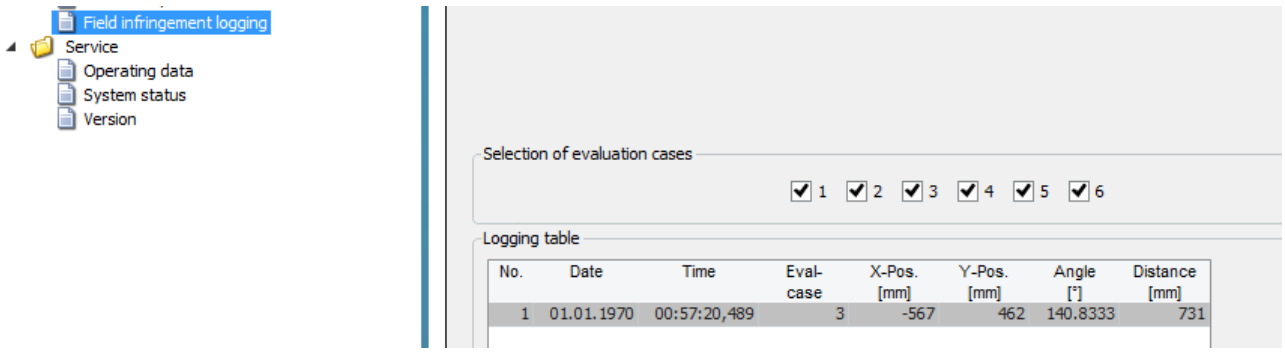


Table 370: Telegram structure: sEN LFEinfringementinfo

Telegram structure: sEN LFEinfringementinfo						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sEN	73 45 4E
Command	Data telegram	String	19		LFEinfringe-mentinfo	4C 46 45 69 6E 66 72 69 6E 67 65 6D 65 6E 74 69 6E 66 6F
Reporting	Start/stop	Enum_8	1	Stop: Start:	0 1	00 01

Table 371: Example: sEN LFEinfringementinfo 1

CoLa A	<STX>sEN{SPC}LFEinfringementinfo{SPC}1<ETX>
	<STX>sEN LFEinfringementinfo 1<ETX>
	sEN LFEinfringementinfo 1
	02 73 45 4E 20 4C 46 45 69 6E 66 72 69 6E 67 65 6D 65 6E 74 69 6E 66 6F 20 31 03
CoLa B	02 02 02 02 00 00 00 19 73 45 4E 20 4C 46 45 69 6E 66 72 69 6E 67 65 6D 65 6E 74 69 6E 66 6F 20 01 3C
	73 45 4E 20 4C 46 45 69 6E 66 72 69 6E 67 65 6D 65 6E 74 69 6E 66 6F 20 01

Table 372: Telegram structure: sEA LFEinfringementinfo

Telegram structure: sEA LFEinfringementinfo						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Event	String	3		sEA	73 45 41
Command	Data telegram	String	19		LFEinfringe-mentinfo	4C 46 45 69 6E 66 72 69 6E 67 65 6D 65 6E 74 69 6E 66 6F
Reporting	Start/stop	Enum_8	1	Stop: Start:	0 1	00 01

Table 373: Example: sEA LFEinfringementinfo 1

CoLa A	<STX>sEA{SPC}LFEinfringementinfo{SPC}1<ETX>
	02 73 45 41 20 4C 46 45 69 6E 66 72 69 6E 67 65 6D 65 6E 74 69 6E 66 6F 20 31 03
CoLa B	02 02 02 02 00 00 00 19 73 45 41 20 4C 46 45 69 6E 66 72 69 6E 67 65 6D 65 6E 74 69 6E 66 6F 20 01 33

NOTE

The answer to the telegram is followed by data that is sent on event.

The sensor only sends the following data if there is a new field infringement detected and noted in the field infringement logging.

Table 374: Telegram structure: sSN LFEinfringementinfo

Telegram structure: sSN LFEinfringementinfo						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sSN	73 53 4E
Command	Data telegram	String	19		LFEinfringe-mentinfo	4C 46 45 69 6E 66 72 69 6E 67 65 6D 65 6E 74 69 6E 66 6F
Time info						
Counter	continious counter of infrigements	Uint_16	2		0 ... 9999d (0 ... 270Fh)	00 00 ... 27 0F
Year		Uint_16	2		0 ... 9999d (0 ... 270Fh)	00 00 ... 27 0F
Month	1 to 12	Uint_8	1		0 ... 12d (0 ... Ch)	00 ... 0C
Day	Day of month 1 to 31	Uint_8	1		0 ... 31d (0 ... 1Fh)	00 ... 1F
Hour	0 to 23	Uint_8	1		0 ... 23d (0 ... 17h)	00 ... 17
Minute	0 to 59	Uint_8	1		0 ... 59d (0 ... 3Bh)	00 ... 3B
Second	0 to 59	Uint_8	1		0 ... 59d (0 ... 3Bh)	00 ... 3B
µSecond	0 to 999999	Uint_32	4		0 ... 999999d (0 ... 000F423Fh)	00 00 00 00 ... 00 0F 42 3F
Infringement-info						
EVC number	1-10	Uint_8	1		0 ... 10d (0 ... Ah)	00 ... 0A
X-Pos [mm]	-20 m ... 20 m -40 m ... 40 m -50 m ... 50 m	Int_32	4	LMS10x, LMS11x, LMS12x, LMS13x: LMS14x: LMS15x, LMS16x:	-20000d ... 20000d (FFFFB1E0h ... 4E20h) -40000d ... 40000d (FFFF63C0h... 9C40h) -50000d ... 50000d (FFFF3CB0h... C350h)	FF FF B1 E0 ... 00 00 4E 20 FF FF 63 C0 ... 00 00 9C 40 FF FF 3C B0 ... 00 00 C3 50
Y-Pos [mm]	-20 m ... 20 m -40 m ... 40 m -50 m ... 50 m	Int_32	4	LMS10x, LMS11x, LMS12x, LMS13x: LMS14x: LMS15x, LMS16x:	-20000d ... 20000d (FFFFB1E0h ... 4E20h) -40000d ... 40000d (FFFF63C0h... 9C40h) -50000d ... 50000d (FFFF3CB0h... C350h)	FF FF B1 E0 ... 00 00 4E 20 FF FF 63 C0 ... 00 00 9C 40 FF FF 3C B0 ... 00 00 C3 50

Telegram structure: sSN LFEinfringementinfo						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Distance [mm]	0 ... 20 m 0 ... 40 m 0 ... 50 m	Uint_32	4	LMS10x, LMS11x, LMS12x, LMS13x: LMS14x: LMS15x, LMS16x:	0 ... 20000d (0h ...4E20h) 0 ... 40000d (0 ... 9C40h) 0 ... 50000d (0h...C350h)	00 00 00 00 ... 00 00 4E 20 00 00 00 00 ... 00 00 9C 40 00 00 00 00 ... 00 00 C3 50
Angle [°]	-45° ... 225°	Int_32	4		-450000d ... +2250000d (FFF92230h ... 225510h)	FF F9 22 30 ... 00 22 55 10
Angle of Layer [°]	-45° ... 225°	Int_32	4		-450000d ... +2250000d (FFF92230h ... 225510h)	FF F9 22 30 ... 00 22 55 10
Object size [mm]	0 ... 20 m 0 ... 40 m 0 ... 50 m	Uint_32	4	LMS10x, LMS11x, LMS12x, LMS13x: LMS14x: LMS15x, LMS16x:	0 ... 20000d (0h ...4E20h) 0 ... 40000d (0 ... 9C40h) 0 ... 50000d (0h...C350h)	00 00 00 00 ... 00 00 4E 20 00 00 00 00 ... 00 00 9C 40 00 00 00 00 ... 00 00 C3 50

Table 375: Example: sSN LFEinfringementinfo

CoLa A	<STX>sSN{SPC}LFEinfringemen-tinfo{SPC}6{SPC}7B2{SPC}1{SPC}1{SPC}0{SPC}19{SPC}21{SPC}9D788{SPC}1{SPC}41{SPC}C3{SPC}0{SPC}CE{SPC}AE8 F8{SPC}0{SPC}A8<ETX>
	02 73 53 4E 20 4C 46 45 69 6E 66 72 69 6E 67 65 6D 65 6E 74 69 6E 66 6F 20 36 20 37 42 32 20 31 20 31 20 30 20 31 39 20 32 31 20 39 44 37 38 38 20 31 20 34 31 20 43 33 20 30 20 43 45 20 41 45 38 46 38 20 30 20 41 38 03
CoLa B	02 02 02 02 00 00 00 40 73 53 4E 20 4C 46 45 69 6E 66 72 69 6E 67 65 6D 65 6E 74 69 6E 66 6F 20 00 06 07 B2 01 01 00 19 21 00 09 D7 88 01 00 00 00 41 00 00 00 C3 00 00 00 00 00 00 00 CE 00 0A E8 F8 00 00 00 00 00 A8 09

12.2.1.5 Diagnostics

12.2.1.5.1 SOPAS error codes

sFA ErrorCode

Table 376: SOPAS error codes

Telegram structure: sFA ErrorCode			
Error code	**Description**	**Dec.**	**Hex.**
Sopas_Ok	No error	0	0
Sopas_Error_METHODIN_ACCESSDENIED	Wrong userlevel, access to method not allowed	1	1
Sopas_Error_METHODIN_UNKNOWNINDEX	Trying to access a method with an unknown Sopas index	2	2
Sopas_Error_VARIABLE_UNKNOWNINDEX	Trying to access a variable with an unknown Sopas index	3	3
Sopas_Error_LOCALCONDITIONFAILED	Local condition violated, e.g. giving a value that exceeds the minimum or maximum allowed value for this variable	4	4
Sopas_Error_INVALID_DATA	Invalid data given for variable, this errorcode is deprecated (is not used anymore).	5	5
Sopas_Error_UNKNOWN_ERROR	An error with unknown reason occurred, this errorcode is deprecated.	6	6
Sopas_Error_BUFFER_OVERFLOW	The communication buffer was too small for the amount of data that should be serialised.	7	7

Telegram structure: sFA ErrorCode			
Error code	**Description**	**Dec.**	**Hex.**
Sopas_Error_BUFFER_UNDERFLOW	More data was expected, the allocated buffer could not be filled.	8	8
Sopas_Error_ERROR_UNKNOWN_TYPE	The variable that shall be serialised has an unknown type. This can only happen when there are variables in the firmware of the device that do not exist in the released description of the device. This should never happen.	9	9
Sopas_Error_VARIABLE_WRITE_ACCESSDENIED	It is not allowed to write values to this variable. Probably the variable is defined as read-only.	10	A
Sopas_Error_UNKNOWN_CMD_FOR_NAMESERVER	When using names instead of indices, a command was issued that the nameserver does not understand.	11	B
Sopas_Error_UNKNOWN_COLA_COMMAND	The CoLa protocol specification does not define the given command, command is unknown.	12	C
Sopas_Error_METHODIN_SERVER_BUSY	It is not possible to issue more than one command at a time to an SRT device.	13	D
Sopas_Error_FLEX_OUT_OF_BOUNDS	An array was accessed over its maximum length.	14	E
Sopas_Error_EVENTREG_UNKNOWNINDEX	The event you wanted to register for does not exist, the index is unknown.	15	F
Sopas_Error_COLA_A_VALUE_OVERFLOW	The value does not fit into the value field, it is too large.	16	10
Sopas_Error_COLA_A_INVALID_CHARACTER	Character is unknown, probably not alphanumeric.	17	11
Sopas_Error_OSAI_NO_MESSAGE	Only when using SRTOS in the firmware and distributed variables this error can occur. It is an indication that no operating system message could be created. This happens when trying to GET a variable.	18	12
Sopas_Error_OSAI_NO_ANSWER_MESSAGE	This is the same as `Sopas_Error_OSAI_NO_MESSAGE` with the difference that it is thrown when trying to PUT a variable.	19	13
Sopas_Error_INTERNAL	Internal error in the firmware, problably a pointer to a parameter was null.	20	14
Sopas_Error_HubAddressCorrupted	The Sopas Hubaddress is either too short or too long.	21	15
Sopas_Error_HubAddressDecoding	The Sopas Hubaddress is invalid, it can not be decoded (Syntax).	22	16
Sopas_Error_HubAddressAddressExceeded	Too many hubs in the address	23	17
Sopas_Error_HubAddressBlankExpected	When parsing a HubAddress an expected blank was not found. The HubAddress is not valid.	24	18
Sopas_Error_AsyncMethodsAreSuppressed	An asynchronous method call was made although the device was built with “AsyncMethodsSuppressed”. This is an internal error that should never happen in a released device.	25	19
Sopas_Error_ComplexArraysNotSupported	Device was built with „ComplexArraysSuppressed“ because the compiler does not allow recursions. But now a complex array was found. This is an internal error that should never happen in a released device.	26	1A

Table 377: Example: sFA ErrorCode Wrong userlevel

CoLa A	<STX>sFA{SPC}1<ETX>
	02 73 46 41 20 31 03
CoLa B	02 02 02 02 00 00 00 05 73 46 41 20 00 01 75

SICK AG
WALDKIRCH
GERMANY
SICK.COM

SICK AT A GLANCE

SICK is a globally leading company in intelligent sensors and sensor solutions. With over 10,000 employees, more than 60 subsidiaries and equity investments as well as numerous agencies worldwide, SICK is always close to its customers.

With a unique range of products and services for factory and logistics automation, SICK creates added value along the entire value chain of its customers. In addition, SICK brings extensive industry and application experience as well as a deep understanding of its customers' processes and requirements.

Partnering for the better, optimizing productivity, elevating quality, and protecting health and safety – this is what SICK stands for.

THAT IS "SENSOR INTELLIGENCE".

8012471/1U1F/2026-02-18/en

SICK
Sensor Intelligence