OPERATING INSTRUCTIONS

MRS1000

3D LiDAR sensor

SICK
Sensor Intelligence.



Described product

MRS1000

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 Explanation of symbols 6
1.3 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 Status indicators 11
3.3 Type code 12
3.4 Type label 12
3.5 Principle of operation 13
3.5.1 Measurement principle 13
3.5.2 Distance measurement 14
3.5.3 Direction measurement 14
3.5.4 Impact of object surfaces on the measurement 14
3.5.5 Scanning range 16
3.5.6 Object sizes 17
3.5.7 Multi-echo analysis 19
3.5.8 Filter 21
3.5.9 Multi-layer technology 26
3.5.10 Scan data output 26
3.5.11 Interlaced mode 27
3.5.12 RSSI values 30
3.5.13 Contamination indication 30
3.5.14 Calculation of the field size for mobile applications 31

4 Transport and storage 34
4.1 Transport 34
4.2 Unpacking 34
4.3 Transport inspection 34
4.4 Storage 34

5 Mounting 35
5.1 Mounting instructions 35
5.2 Mounting device 35

5.3 Mounting multiple devices ... 36

6 Electrical installation ... 37

6.1 Wiring instructions ... 37
6.2 Prerequisites for safe operation of the device ... 38
6.3 Connection diagram ... 41
6.4 Connection options ... 43
6.5 Connecting the device electrically ... 44

7 Operation ... 45

7.1 Operation with SOPAS ET ... 45
7.1.1 Description of the device window ... 46
7.2 Operation with teach-in (EasyTeach) ... 53
7.3 Opening the web server user interface (SOPASair) ... 53

8 Maintenance ... 54

8.1 Maintenance plan ... 54
8.2 Cleaning ... 54

9 Troubleshooting ... 55

9.1 General faults, warnings, and errors ... 55
9.2 Repairs ... 55
9.3 Returns ... 55
9.4 Disposal ... 55

10 Technical data ... 56

10.1 Features ... 56
10.2 Mechanics/electronics ... 58
10.3 Dimensional drawing ... 60
10.4 Performance ... 61
10.5 Interfaces ... 61
10.6 Ambient data ... 62

11 Accessories ... 63

12 Annex ... 64

12.1 Declarations of conformity and certificates ... 64
12.2 Licenses ... 64
12.3 Configuration without computer ... 64
12.3.1 Field evaluation ... 64
12.4 Configuration with computer ... 65
12.4.1 Output of measured values via terminal program ... 65
12.4.2 Measured value display and configuration via start window ... 67
12.4.3 Selecting the application ... 67
12.4.4 Field evaluation ... 67
12.4.5 Configuring interfaces ... 68
12.4.6 Some useful functions ... 68

12.4.7 Configuring the device 68

13 Telegram listing (EN) 69

13.1.1 About this document 71
13.1.2 Communication format 72
13.1.3 Workflows 75
13.1.4 Telegrams 77
13.1.5 Diagnostics 160

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 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.

1.3 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 MRS1000 3D LiDAR sensor features four scan layers and is designed for the following applications:

- Detection of objects during continuous output of measured data as required
- Field monitoring of freely defined areas with signaling of object detections via switching outputs

It is suitable for applications which demand precise, non-contact optical measuring contours and dimensioning.

Typical areas of application include property/building protection, access monitoring, and collision protection.

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 1: 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 2: Scope of delivery

Item	Component	Comments
1	Device in the version ordered	Depending on version Without connecting cables and brackets
1	Set of protective caps for electrical connections	Attached to the connections
1	Printed safety notes, multilingual	Quick guide and general safety notes

3.2 Status indicators

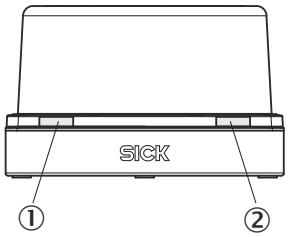


Figure 1: Status indicators

① LED1

② LED2

LED1 (color)	LED2 (color)	Description
● (Red)	● (Red)	Start-up, parameterization, firmware update, correctable error, measurement stopped
-	-	Off
● (Green)	● (Yellow)	Field breached
● (Green)	● (Green)	Field free
☀ (Red)	☀ (Red)	Fatal error
● (Green)	● (Green)	On / Ready for operation
☀ (Green)	☀ (Green)	Standby
● (Green)	☀ (Yellow)	Restart after time; input
☀ (Yellow)	● (Green)	Contamination warning
☀ (Yellow)	☀ (Red)	Contamination error
☀ (Yellow)	☀ (Yellow)	EasyTeach/Teach In
☀ (Green)	☀ (Red)	Alignment mode
● (Green) ● (Yellow) ● (Red)	● (Green) ● (Yellow) ● (Red)	Identifying the device

● = illuminated; ☀ = flashing

3.3 Type code

Type code structure

MRS abccd – efghij

Position	Description	Characteristic
-	Device name type	MRS: Multilayer Range Sensor
a	Device type	1: MRS1000
b	Version (10% range & field of view (FOV))	1: 16m @10%, FOV 275°
c	Module (sender and receiver units (SRU))	04: 4 SEE (-2.5°; 0°; 2.5°; 5°)
d	Application	C: Combined (detection & ranging)
–		
e	Housing version	0: IP65 enclosure rating 1: IP67 enclosure rating 9: Special version
f	Connection	1: Rotatable connector, 3 x M12 9: Special version
g	Version	1: Standard 9: Special version
h	Laser type	0: Laser class 1, infrared
i	Motor frequency	1: 12.5 Hz
j	Color tone / label	0: blue (RAL 5012) 1: gray (RAL 7042)

3.4 Type label

The type label gives information for identification of the sensor.

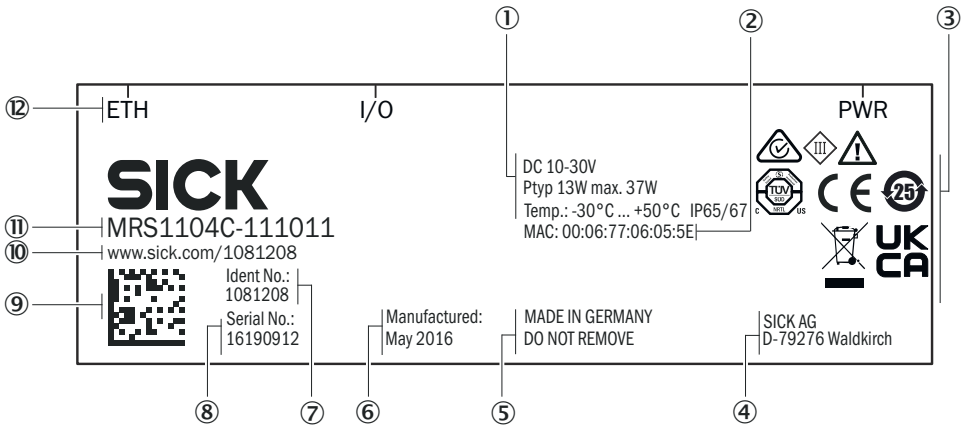


Figure 2: MRS1000 type label (example)

① Voltage supply, typical power, max. power, operating temperature, enclosure rating
② MAC address
③ Conformity mark/certification mark, symbol: Observe the operating instructions!
④ Manufacturer
⑤ Production location, note: Do not remove type label
⑥ Production date
⑦ Part number
⑧ Serial number
⑨ Data Matrix code with product data and link to product page
⑩ Web address of product page
⑪ Type code
⑫ Labeling of connections

3.5 Principle of operation

3.5.1 Measurement principle

The device is an optoelectronic LiDAR sensor that contactlessly scans the outline of its surroundings with the help of laser beams. The device measures its surroundings in two-dimensional polar coordinates, relative to its measurement origin. This is marked by a circular indentation in the center of the optics cover. If a laser beam strikes an object, the position of that object is determined in terms of distance and angle.

With the MRS1000, this is done in 4 spread-out scan layers (layers 1 to 4).

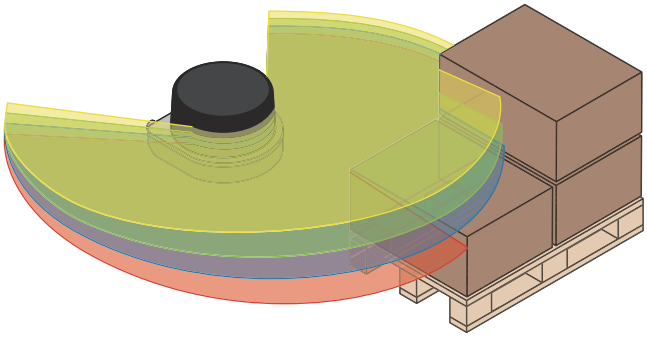


Figure 3: LiDAR sensor with 4 scan layers

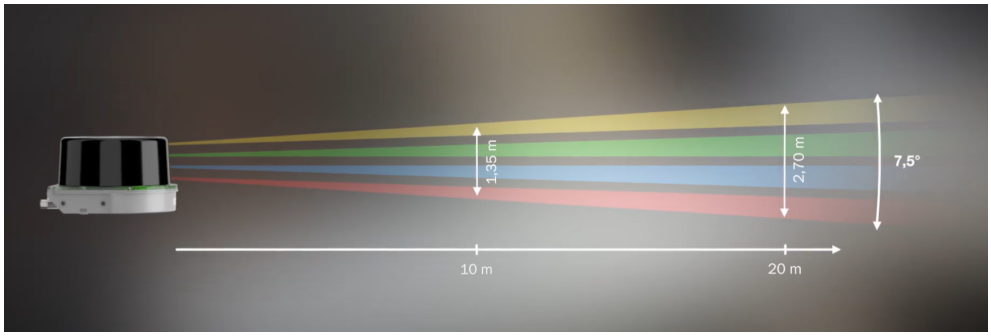
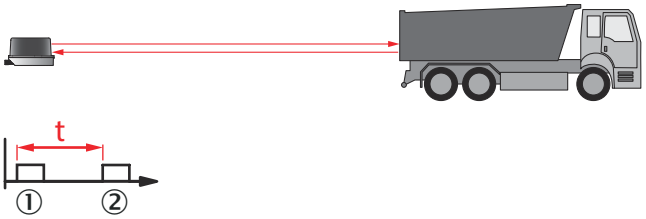


Figure 4: LiDAR sensor with 4 scan layers

3.5.2 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 Send pulse

② Receive pulse

The device uses SICK's own HDDM+ (High Definition Distance Measurement plus) technology. With this measurement process, a measured value is formed statistical evaluation of multiple single pulses. The multi-pulse feature of HDDM+ evaluates up to 165000 echoes per second. Every single HDDM+ measured value output therefore provides even more information because it is not just composed of a single time-of-flight measurement, but includes evaluated information from numerous pulses. In this process, the digitized echoes are compiled into data packages which overlap during evaluation. This ensures a significantly more stable time and distance measurement.

3.5.3 Direction measurement

The laser beams are emitted using internally rotating sender-receiver units (SRUs) and scan the surroundings orbitally. The received measured values are assigned to the associated angular cut and thus to the direction.

3.5.4 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.

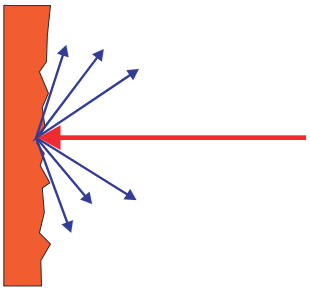


Figure 5: Reflection of light on the surface of the object

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.

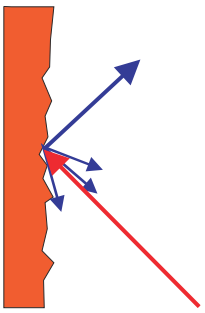


Figure 6: Angle of reflection

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.

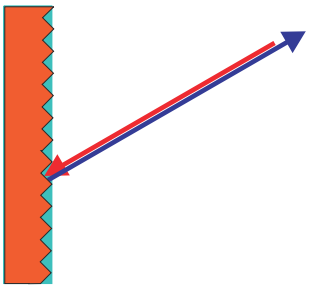


Figure 7: Retroreflection

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.

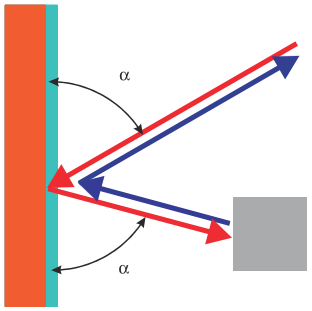


Figure 8: Specular surfaces

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.

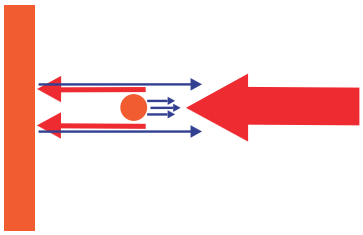


Figure 9: Object smaller than the laser beam diameter

3.5.5 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.

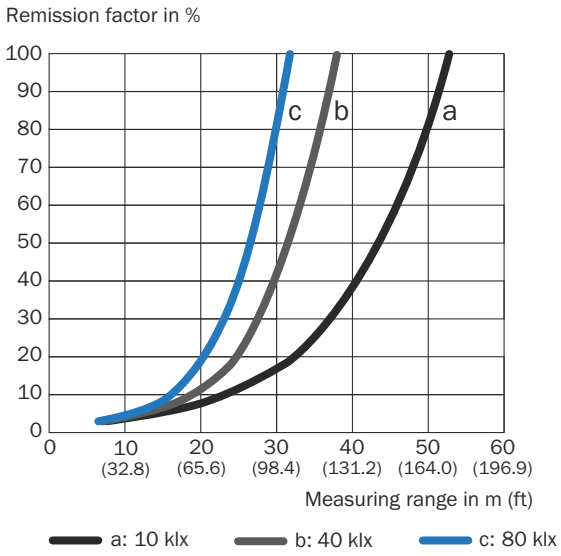


Figure 10: Scanning range as a function of the remission factor for various ambient light influences (no filter activated; no affect from fog, rain or dust)

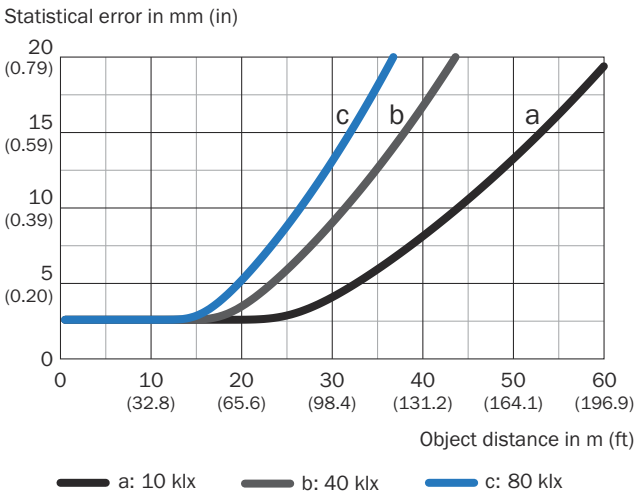


*Figure 11: Statistical error for **white objects** as a function of object distance for various ambient light influences*

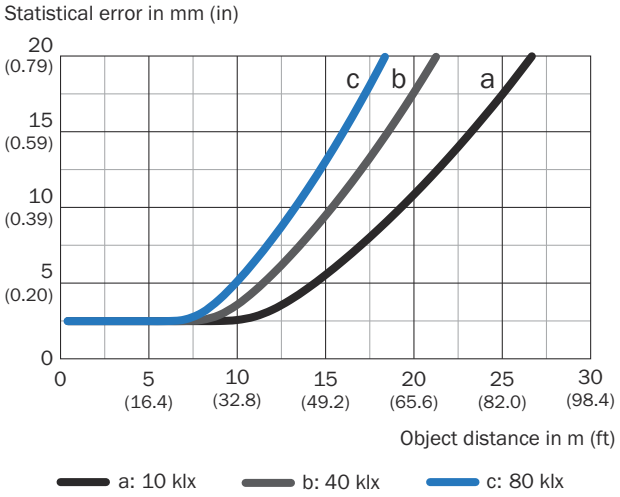


*Figure 12: Statistical error for **black objects** as a function of object distance for various ambient light influences*

3.5.6 Object sizes

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



Figure 13: Beam expansion

① Expanded laser beam

② Optical axis

Values required to calculate the light spot size:

- Light spot size on the device cover: 7 mm (rounded up)
- Divergence of 1 light spot: 0.59 deg (10.4 mrad)
- HDDM+ supplement (1 spot comprises several superimposed individual pulses): 0.50 deg (8.7 mrad)

Formula for calculating the light spot width:

(Light spot divergence [mrad] + HDDM+ supplement [mrad]) x distance [mm] + light spot size on the device cover [mm] = light spot width [mm]

Example calculation of the light spot width at a distance of 16 m (16000 mm):

(10.4 mrad + 8.7 mrad) * 16,000 mm + 7 mm = 312.6 mm

Formula for calculating the height of the light spot:

Light spot divergence [mrad] x Distance [mm] + Light spot height at the n device cover [mm] = Light spot width [mm]

Example calculation of the light spot height at a distance of 16 m (1,600 mm):

10.4 mrad * 16,000 mm + 7 mm = 173.4 mm

Formula for calculating the minimum object size:

(Light spot divergence [mrad] + angular resolution [mrad]) x distance [mm] + light spot height on the device cover [mm] = minimum object size [mm]

Example calculation of the minimum object size at a distance of 16 m (16,000 mm):

Angular resolution: 0.25° → 4.4 mrad

(10.4 + 8.7 + 4.4 mrad) * 16,000 mm + 7 mm = 383 mm

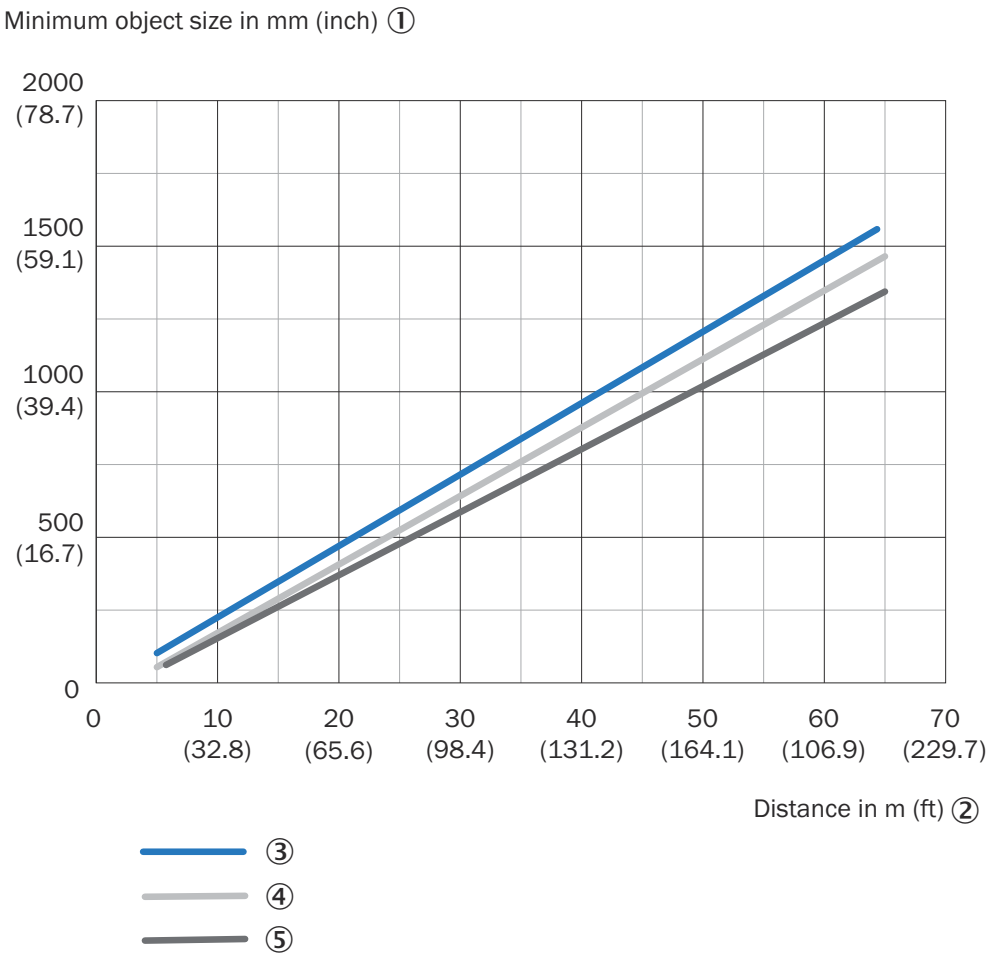


Figure 14: Minimum object size as a function of distance

① Minimum object size in mm (inches)
② Distance in m (ft)
③ Minimum object size
④ Minimum object size (2-fold interlaced mode)
⑤ Minimum object size (4-fold interlaced mode)

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 LiDAR sensor and the object must not be moving.

3.5.7 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 three echo signals for each measuring beam to deliver reliable measurement results, even under adverse ambient conditions.

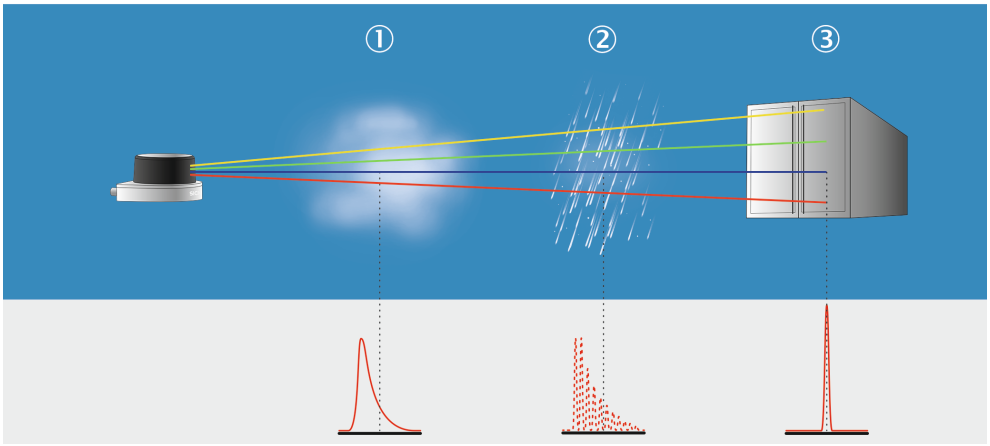


Figure 15: Multi-echo analysis: example industrial application for ports, cranes, and traffic.

① Fog
② Rain
③ Measuring object

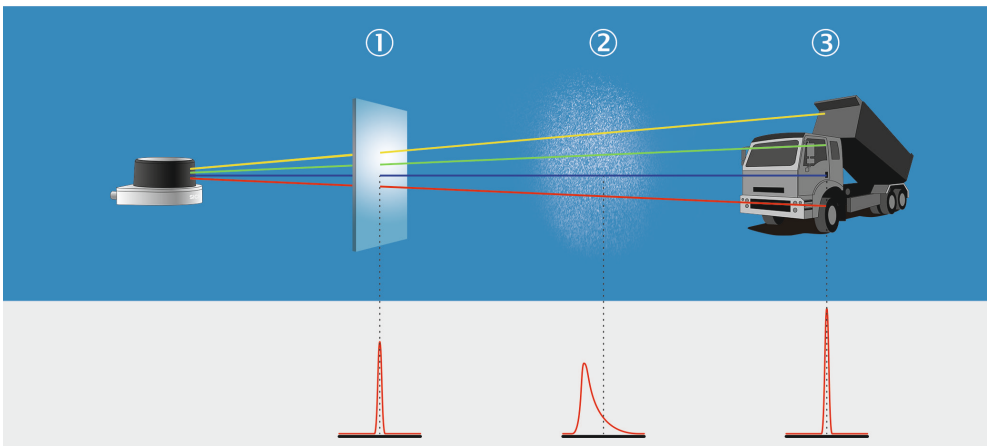


Figure 16: Multi-echo analysis: example industrial application for mining.

① Glass pane
② Dust
③ Measuring object

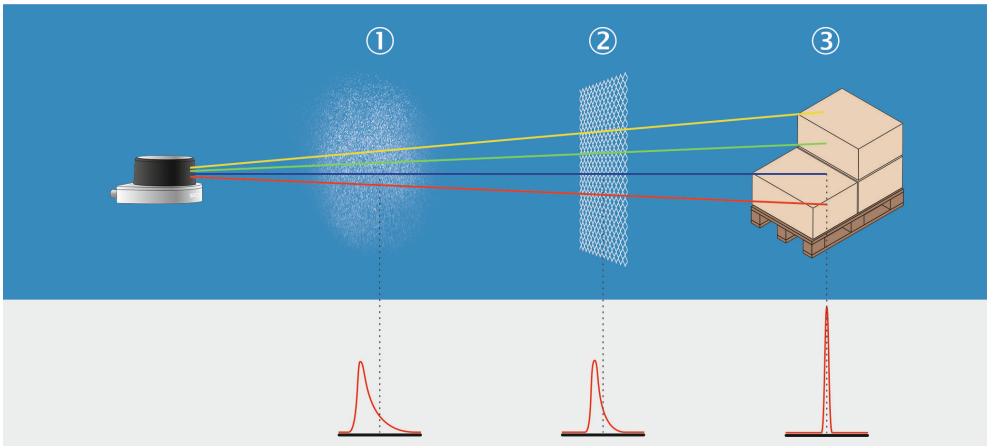


Figure 17: Multi-echo analysis: example industrial application for industrial vehicles.

① Dust
② Fences
③ Measuring object

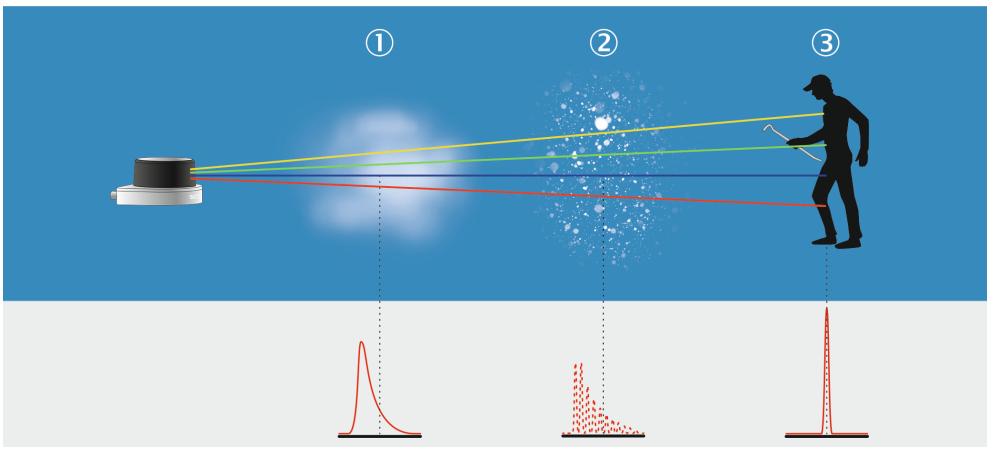


Figure 18: Multi-echo analysis: example industrial application for building management.

① Fog
② Snow
③ Measuring object

3.5.8 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.

You can arbitrarily combine the filters. If several filters are active, then the filters are applied sequentially to the results of the preceding filter. Processing occurs in the following sequence: echo filter, edge filter, particle filter, median filter, ground reference evaluation, average filter. It should be noted, however, that the measurement data output may be delayed due to the multiple calculation steps.

For many applications, it is useful to use a combination of the median filter and the mean filter. The median filter can eliminate the larger outliers, which “smooths” the scan line. The values calculated by the median filter are then averaged over a configurable number of scans by the mean filter.

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

3.5.8.1 Echo filter

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

You can set whether the first, the last, or all three echoes are output.

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

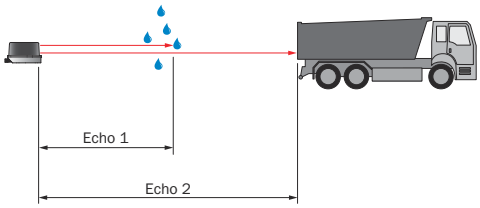


Figure 19: Without the echo filter: The device receives unwanted echoes from ambient conditions such as rain.

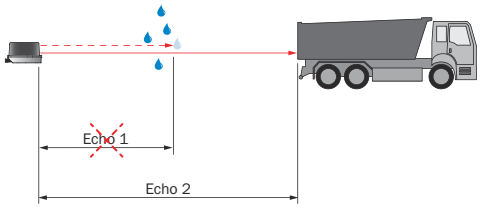


Figure 20: Using the echo filter (setting: last echo): the device screens out unwanted echoes from ambient conditions as per the settings chosen.

3.5.8.2 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.

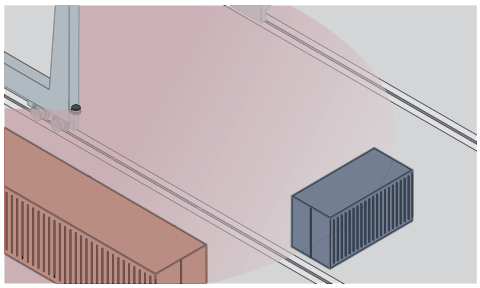


Figure 21: Without the fog filter: objects are difficult to detect through the fog due to reflections.

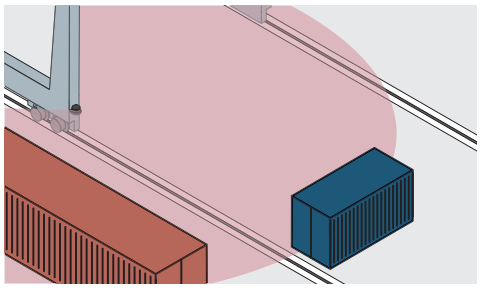


Figure 22: Using the fog filter: objects can be detected reliably because unwanted echoes are screened out.

3.5.8.3 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.

NOTE

If the particle filter is activated, measurement data output or the response time of the field evaluation is delayed by one scan.

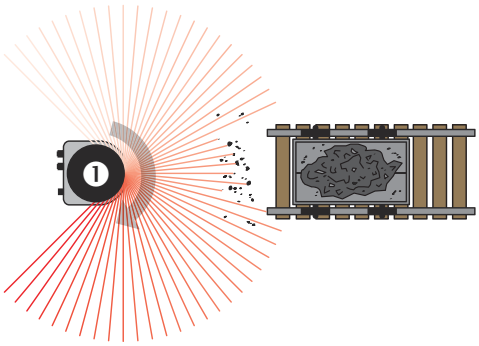


Figure 23: Without the particle filter: Violation of the contour due to dust particles in the vicinity of the object.

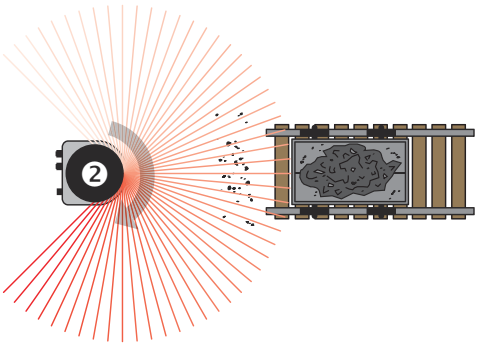


Figure 24: Using the particle filter: The response to dust particles in the detection field is delayed by one scan. Particles can thereby be blanked.

3.5.8.4 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.

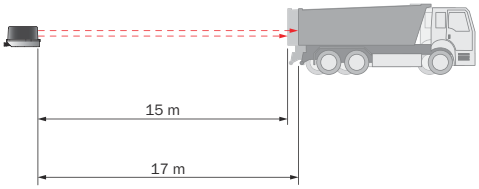


Figure 25: Without the mean filter: The device detects and processes all received signal values.

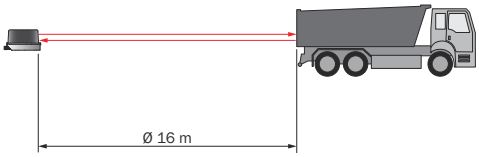


Figure 26: Using the mean filter: The device calculates an average value from several signal values.

3.5.8.5 Median filter

This filter is suitable for excluding individual outliers from the calculation of an average value.

The moving median filter sorts the measured values according to their size and selects the middle value from this sequence. The values taken into consideration are defined by a matrix, the size of which is configurable.

A 3 x 3 matrix, for example, takes into consideration 9 measured values consisting of the distance values of a point and its two neighboring points, along with the distances values of these points determined from the preceding and following scan. The fifth largest measured value is outputted as the distance value.

Median calculation: 850 (lowest value) | 850 | 950 | 1,100 | **1150** (median) | 1,150 | 1,200 | 1,200 | 1,250 (highest value)

Table 3: Example: Median for scan 2 and angle 3 (3 x 3 matrix: scan 1 to 3, angle 2 to 4)

	Angle (distance values)					
Scan	**1**	**2**	**3**	**4**	**5**	**...**
1	0	**0**	**850**	**1100**	1150	...
2	0	**0**	**950**	**1200**	1250	...
3	0	**0**	**850**	**1150**	1200	...
...	...	...	...	...	...	...

Median calculation: 0 (lowest value) | 0 | 0 | 850 | **850** (median) | 950 | 1,100 | 1,150 | 1,200 (highest value)

Table 4: Example: Median for scan 2 and angle 4 (3 x 3 matrix: scan 1 to 3, angle 3 to 5)

	Angle (distance values)					
Scan	**1**	**2**	**3**	**4**	**5**	**...**
1	0	0	**850**	**1100**	**1150**	...
2	0	0	**950**	**1200**	**1250**	...
3	0	0	**850**	**1150**	**1200**	...
...	...	...	...	...	...	...

Median calculation: 850 (lowest value) | 850 | 950 | 1,100 | **1150** (median) | 1,150 | 1,200 | 1,200 | 1,250 (highest value)

3.5.8.6 Ground reference evaluation

NOTE
When the ground reference evaluation function is activated, the mean filter is automatically deactivated.

The ground reference evaluation function enables the LiDAR sensor to protect hazardous areas when used in conjunction with automated guided vehicle systems (AGVS). Any objects lying on the path (pallets or small load carriers), or steps and depressions in the ground, are distinguished and detected, thereby avoiding collisions.

The PC software SOPAS ET is used to configure the device and enter the installation location (orientation of the device), "Operation with SOPAS ET", page 45. The following application parameters must be taken into consideration and adhered to:

Table 5: Application parameters for ground reference evaluation

Condition	Value
Mounting height	0.3 ... 1.3 m
Tilt angle	+11° ... +30° Overhead mounting [1]: -10° ... -30° Details see table 6, page 25
Roll angle	± 0.5°
Yaw angle	Any
Required accuracy for the configuration data inputs	Mounting height: ± 5 cm Tilt angle: ± 0.5°
Ambient conditions	Indoor applications, level floor surface, no slope, min. corridor width 2 m Shiny surfaces can negatively affect the availability of the function

1) The following offset values must be configured for overhead mounting: yaw angle 180°, tilt angle 180°

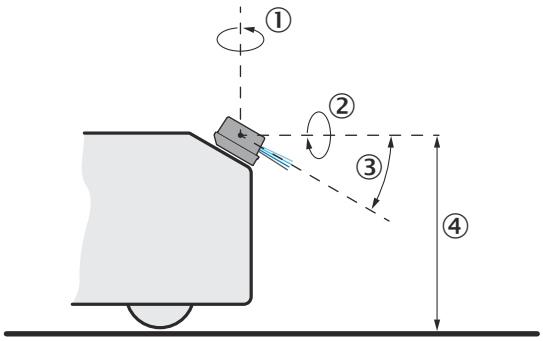


Figure 27: Ground reference evaluation, explanation of application parameters

① Yaw angle
② Roll angle
③ Tilt angle
④ Mounting height

Table 6: Detailed data for the tilt angle

Mounting height [m]	Tilt angle [°]	Tilt angle for overhead mounting [°]
0.3	11.0	-10.0
0.4	12.0	-10.0
0.5	13.0	-10.0
0.6	14.0	-11.0
0.7	14.0	-11.5
0.8	15.0	-12.0
0.9	16.0	-13.0
1.0	16.5	-13.5
1.1	17.5	-15.0
1.2	18.5	-16.0
1.3	19.0	-17.0

Based on the aforementioned application parameters, the maximum measuring range is 5 m (for the highest mounting height and lowest tilt angle as per the last table row).

The SOPAS ET software also provides several other parameters for adjusting the sensitivity of the ground reference evaluation.

3.5.8.7 Edge filter

The edge filter eliminates erroneous or extreme distance values at edges. These arise as a result of laser light that partially hits an object in the front and partially hits an object farther away, or due to too little light remission at the object itself, see "Impact of object surfaces on the measurement", page 14. The filter evaluates the difference in distance between adjacent points.

When the edge filter is activated, the device sets the distance value to 0 at each edge (values in **bold**) as soon as the parameterized neighbor distance has been exceeded in the edge filter area.

Table 7: Example: Measured values with and without edge filter

	Angle (distance values in mm)							
Scan	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**...**
1, not filtered	0	750	1145	1150	1147	1500	0	...

	Angle (distance values in mm)							
Scan	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**...**
1, filtered	0	**0**	1145	1150	1147	**0**	0	...

The edge filter enables points to be entirely suppressed at the outer edges of the object. In this case, the width of an object is detected by up to 2× angular resolution too narrow.

Maximum neighboring distance

The maximum distance difference in millimeters between two measuring points on the edge can be parameterized. A measuring point on the edge that has a greater distance difference than the parameterized value to its adjacent measuring point is taken into account.

Maximum edge filter range (range limitation)

The maximum distance can be set up to 16,000 mm. The measuring points in the parameterized range are taken into account.

3.5.9 Multi-layer technology

The multi-layer technology of the device uses four scan layers at different vertical angles to compensate for pitch angle, for example when the device is attached to a vehicle. This enables the device to reliably detect an object even, for example, when the vehicle accelerates or brakes.

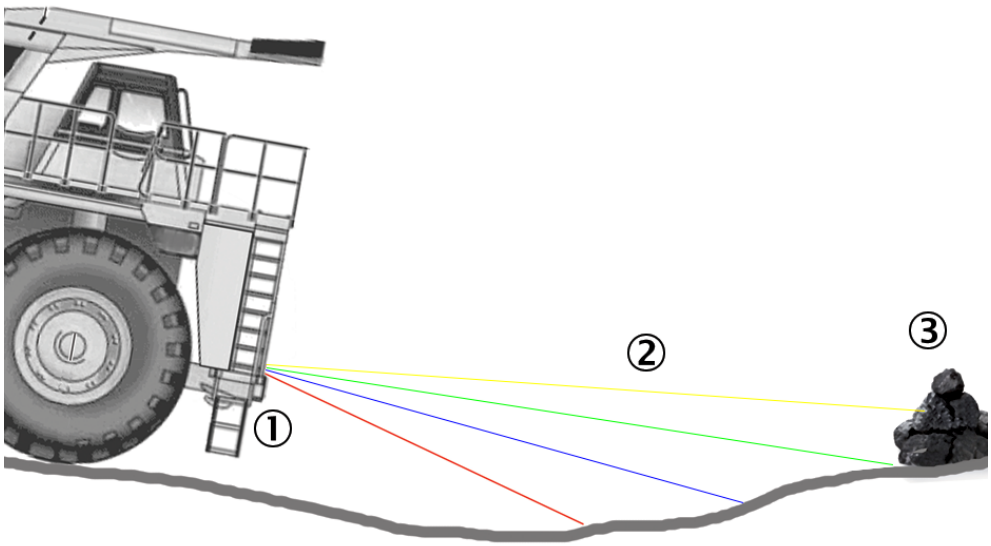


Figure 28: Multi-layer technology

① LiDAR sensor
② Scan layers
③ Object

3.5.10 Scan data output

The device has 4 modules, each of which scans one plane.

The scan data is always output on a per module basis (scan layer). The data for a specific scan layer is output every 20 ms. The device requires 80 ms to output the data for all four scan layers.

In the data message, the individual scans and the associated scan layers are identified by their time stamp and layer address.

Data message:

```
<STX>sSN LMDscandata 1 0 1053B01 0 0 8531 8531 299855DA 2A6073CC 0 0 0 0 0 1388 B4 ...
<STX>sSN LMDscandata 1 0 1053B01 0 0 8532 8532 2998A3BB 2A60C6C8 0 0 0 0 FF06 1388 B4 ...
<STX>sSN LMDscandata 1 0 1053B01 0 0 8533 8533 2998F1CF 2A61100C 0 0 0 0 FA 1388 B4 ...
<STX>sSN LMDscandata 1 0 1053B01 0 0 8534 8534 29993FE6 2A615E0B 0 0 0 0 FE0C 1388 B4 ...
```

Time stamp:

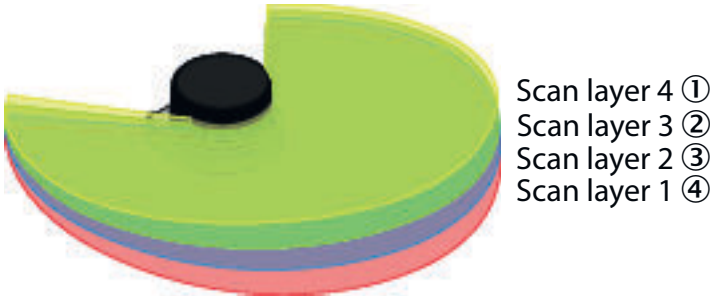
- Line 1: **299855DA** = 697849306 = 697849.306 ms
- Line 2: **2998A3BB** = 697869243 = 697869.243 ms
- Difference = 20 ms (corresponds to a scan data output at 50 Hz)

Location address (as per DIN 70000):

- Line 1: **0**= 0 (corresponds to 0°) = scan layer 2
- Line 2: **FF06**= -250 (corresponds to -2.5°) = scan layer 3
- Line 3: **FA**= 250 (corresponds to 2.5°) = scan layer 1
- Line 4: **FE0C**= -500 (corresponds to -5°) = scan layer 4
- The scans and the output of scan data occur in the sequence: scan layer 2, scan layer 3, scan layer 1, scan layer 4.

Table 8: Address assignment method

Scan layer	DIN 70000	Physical
1	+2.5°	-2.5°
2	0°	0°
3	-2.5°	+2.5°
4	-5°	+5°



① Scan layer 4
② Scan layer 3
③ Scan layer 2
④ Scan layer 1

3.5.11 Interlaced mode

With Interlaced mode, scan data can be acquired with a fine angular resolution even at high scanning frequencies. Recurring sequences of scans are generated.

- Standard: Angular resolution 0.25°, response time: 80 ms
- Dual evaluation: Angular resolution 0.125°, response time: 160 ms
- Quadruple evaluation: Angular resolution 0.0625°, response time: 320 ms

Raw data via LMDscandata telegram

The first scan of a sequence has one scan point more than the other scans of a sequence. This allows the customer to create a scan from a nested sequence. The device sends a telegram after every single scan (rotation). The individual scans of a sequence are not merged into one telegram. The merging of the telegram must be implemented externally.

Field evaluation

For the field evaluation, the device internally merges the interlaced sequence to an internal telegram. This data telegram is then used internally to evaluate the fields. It is not available as an output for the customer.

SOPAS ET

The **field evaluation monitor** shows the merged data of a sequence. The **ScanView** displays the individual scans, similar to the LMDscandata telegrams sent to the customer controller. Normally, the jitter of the angular position of the scan points is detectable.

Parameterization via SOPAS ET

The interlaced mode must be activated via SOPAS ET in the default settings and added as an additional mode in the drop-down menu.

Parameterization via telegram

The interlaced mode can be activated with the **mLMPsetscancfg** telegram. The scanning frequency and angular resolution can be set as parameters.

If the device can only adjust to the desired combination of scanning frequency and angular resolution in interlaced mode, then interlaced mode is activated. If this combination is also possible without interlaced mode, it is deactivated.

Technical limits

The interlaced mode is implemented so that three modules of the device are in the scan field when the angle switches for the next interlaced scan. This causes some of the measuring points in the switching range to cause uneven angular steps. This jitter varies depending on the set interlaced mode. Depending on the selected interlaced mode, the jittering of the individual measuring points only affects a defined angular range and not the entire scan field. The actual angle is transmitted by telegram to show the correct position of the measuring points.

The 0° position is located directly in the blind zone of the scan field, shown in yellow in figure 29. For this reason, synchronizing the interlaced mode to one of the four scan layers has no effect.

For the MRS1000, this is the horizontal scan layer.

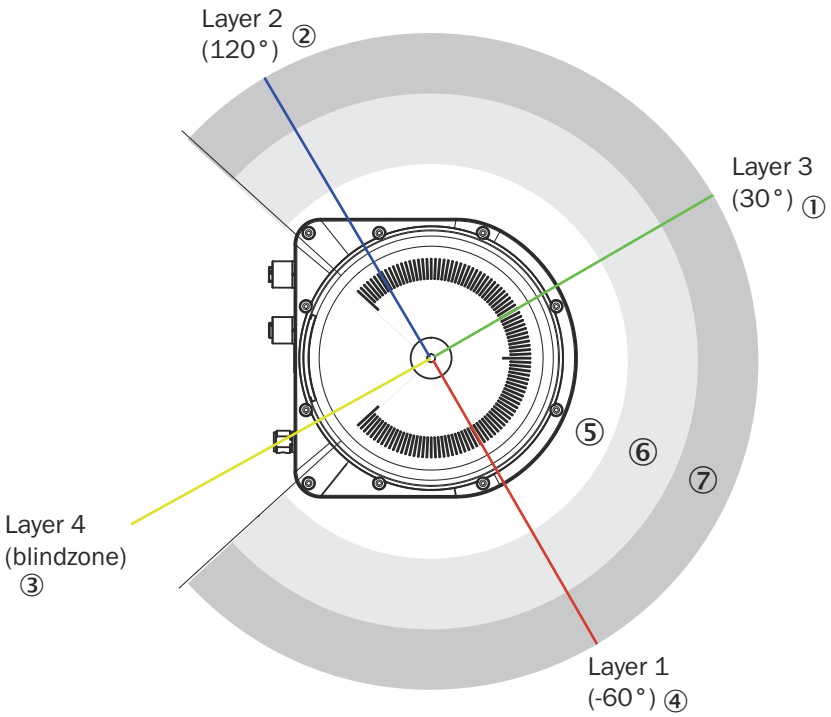


Figure 29: Position of the scan layers when switching to interlaced mode

① Scan layer 3 (30°)
② Scan layer 2 (120°)
③ Scan layer 4 (blind zone)
④ Scan layer 1 (-60°)
⑤ Near range
⑥ Detection zone
⑦ Measuring range

Affected angular range and angular steps:

- With interlaced mode 2x: ± 0.125°

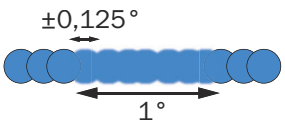


Figure 30: Jittering of the individual measuring points in the affected angular range

- With interlaced mode 4x: ± 0.0625°

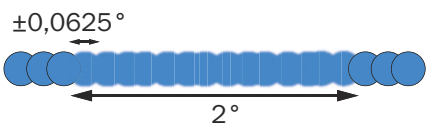


Figure 31: Jittering of the individual measuring points in the affected angular range

Formula for calculating the minimum object size in interlaced mode

For calculation of the minimum object size see "Object sizes", page 17.

With interlaced mode 2x: (10.4 + 8.7 + 4.4 mrad / 2) * 16,000 mm + 7 mm = 348 mm

With interlaced mode 4x: (10.4 + 8.7 + 4.4 mrad / 4) * 16,000 mm + 7 mm = 330 mm

3.5.12 RSSI values

RSSI (Received Signal Strength Indicator) is the measure of the signal strength that the device receives. This value is calculated for every measurement. The device therefore provides, for every echo signal, an associated RSSI value for the signal strength.

The RSSI values have a resolution of 8 bits with whole-number values between 1 and 255, where 1 stands for the weakest signal and 254 for the strongest signal (e.g., with one reflector). A value of 255 means “dazzled”.

The value 0 (zero) means that the received energy was too low to produce a valid measured value. A valid distance measurement has an RSSI value of at least 1.

If the RSSI value is 0, then no distance measurement is possible. There can be two reasons for this:

- The target object lies outside the sensing range.
- The target object has an extremely low remission.

Please note that white paper can have very similar values as a reflector at a short distance.

The RSSI values are sensor-specific, relative values that can vary slightly between different devices and during the service life of the device.

3.5.13 Contamination indication

The device has an front screen to protect it. This front screen can get dirty. 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 contamination is constantly measured by a separate system during operation. A contamination warning is output first for the different degrees of contamination. If the front screen is not cleaned and contamination increases, then a contamination error is output.

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

- **Deactivated**: No indication of contamination.
- **High availability**: Contamination warning and contamination errors are only output when the front screen is uniformly contaminated. This means that all active sectors (e.g., 6 by default) must have reached the limit value for a warning or error.
- **Available**: Contamination warning and contamination errors are output when the front screen is partially contaminated. With this strategy, the front screen is divided into two monitored areas (by default: 3 sectors on the right half and 3 sectors on the left half). Only when all 3 sectors of a monitored area have reached the limit value for warning or error does the contamination warning or contamination error become active.
- **Semi-sensitive**: Contamination warning and contamination errors are output when the front screen is partially and only slightly contaminated. The sensor evaluates all sectors and activates the contamination warning or contamination error using the following strategy:

  - No warning/error: if the sum of sectors with reached limit value for error = 0 and the sum of sectors with reached limit value for warning ≤ 1
  - Warning: if the sum of sectors with reached limit value for error = 1 and the sum of sectors with reached limit value for warning ≥ 2
  - Error: if the sum of sectors with reached limit value for error ≥ 2
- **Sensitive**: Contamination warning and contamination errors are output even when there is just a little or local contamination. When a sector has reached the limit value for warning or error, the contamination warning or contamination error becomes active.

NOTE

- The cleaner the application environment is, the lower you can set the contamination indication sensitivity. If a high precision of the measured values is required, the contamination indication must be set to the most sensitive level.
- If a weather protection hood is mounted on the device, the accessory combination used must be communicated to the device in SOPAS ET.

Contamination warnings and contamination errors are displayed on the status indicators of the device see "Status indicators", page 11.

3.5.14 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 56.

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.

3.5.14.1 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.

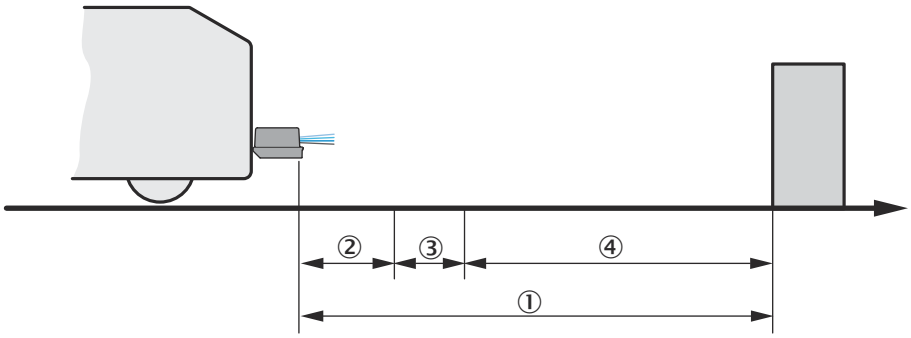


Figure 32: Stopping distance

① S_A

② S_{AnF}

③ S_{AnS}

④ S_{Br}

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 56.

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).

3.5.14.2 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

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.
- To avoid ingress of dust and water, only remove the protective elements, e.g. protective caps of the electrical connections just before attaching the connecting cable.

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.

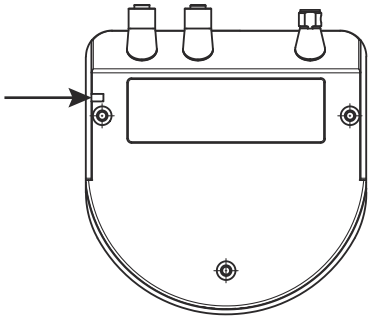
4.4 Storage

- Electrical connections are provided with a protective cap.
- Do not store outdoors.
- Store in a place protected from moisture and dust.
- Recommendation: Use the original packaging.
- To allow any residual dampness to evaporate, do not package in airtight containers.
- Do not expose to any aggressive substances.
- Protect from sunlight.
- Avoid mechanical shocks.
- Storage temperature: see "Technical data", page 56.
- Relative humidity: see "Technical data", page 56.
- 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.
- The device can be mounted in any position.
- When mounting the device upside down, make sure that the ventilation duct of the ventilation element remains free of water and contamination.



- 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 63.
- 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 63.
- 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.
- Do not place magnetic objects on the optics cover, since they can disturb the angular measurement.
- 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 affix any labels or stickers to the optics cover.
- Do not subject the device to excessive shock or vibrations. In systems subjected to heavy vibrations, secure the fixing screws with screw-locking devices.
- When mounting the device up high, and in particular to secure the device during mounting, a retaining cable for fastening is available as an accessory.

5.2 Mounting device

1. Mount the device in a suitably prepared bracket using the fixing holes provided (see "Dimensional drawing", page 60). Mounting brackets are available as accessories, see "Accessories", page 63.
2. Make the electrical connection. Attach and tighten a voltage-free cable, see "Connecting the device electrically", page 44.
3. Align the vertical center line of the field of view of the device with the center of the area to be monitored. The marking on the upper side of the optics cover serves as a bearing alignment aid.

NOTE
By enabling alignment mode, it is possible to activate a single one of the four scan layers. This makes it easier to align the device using this selected reference plane.

4. Switch on the supply voltage.
✓ After successful initialization, the two status LEDs light up green. The device is ready for use.
5. Perform a fine adjustment using a test target and, if necessary, use the alignment aid.

5.3 Mounting multiple devices

NOTICE RISK OF INTERFERENCE FROM OTHER DEVICES!
Radiation sources with a wavelength of 850 nm can cause interference if they affect the device directly.

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.

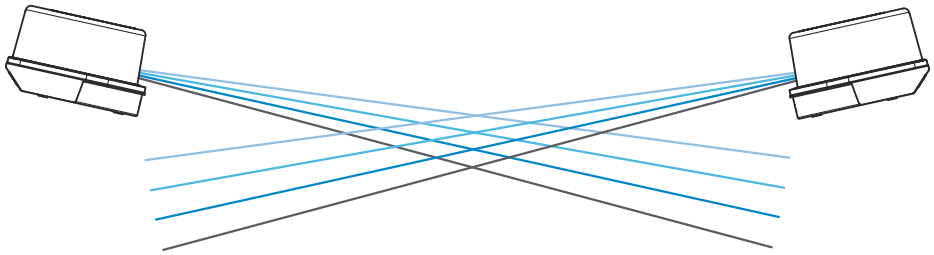


Figure 33: Arrangement for 2 devices

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.

All electrical connections of the device are configured as M12 round connectors.

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.

Observe the following notes to ensure safe and trouble-free operation:

- Connect the connecting cables in a de-energized state. Do not switch on the supply voltage until installation is complete and all connecting cables have been connected to the device and control.
- Wire cross-sections in the supply cable from the customer's power system should be designed in accordance with the applicable standards. Protect the device with an external slow-blow fuse of 5 A at the beginning of the supply cable, viewed from the voltage supply.
- The specified enclosure rating of the device when mounted is reached only if suitable mating connectors or protective caps are used.
- Do not open the screwed housing of the device, since the warranty will then become void.
- Turn the rotatable electrical connections max 270° from end position to end position.
- Prior to connecting the I/O line, check the device configuration for the inputs/outputs.
- Avoid tensile loads to the connecting cables.
- Maximum cable lengths for the voltage supply depending on the power supply voltage used (conditions: cable cross-section 0.34 mm^2, 20 °C cable temperature, 10 V applied to the device, max. power consumption as per type label): 6.5 m at 12 V; 46.5 m at 24 V; 66.5 m at 30 V.
- The power supply unit must be able to bridge a power failure of 20 ms and provide safe isolation (SELV/PELV). Suitable power supply units are available as accessories from SICK.

6.2 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

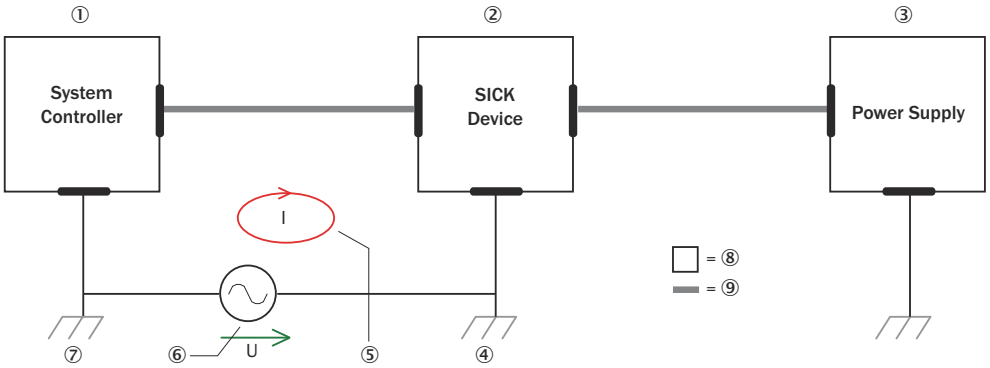


Figure 34: Example: Occurrence of equipotential bonding currents in the system configuration

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.

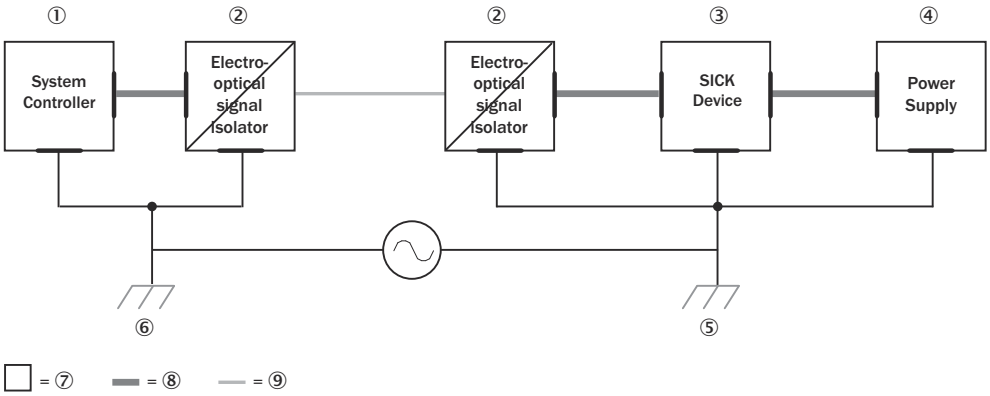


Figure 35: Example: Prevention of equipotential bonding currents in the system configuration by the use of electro-optical signal isolators

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.

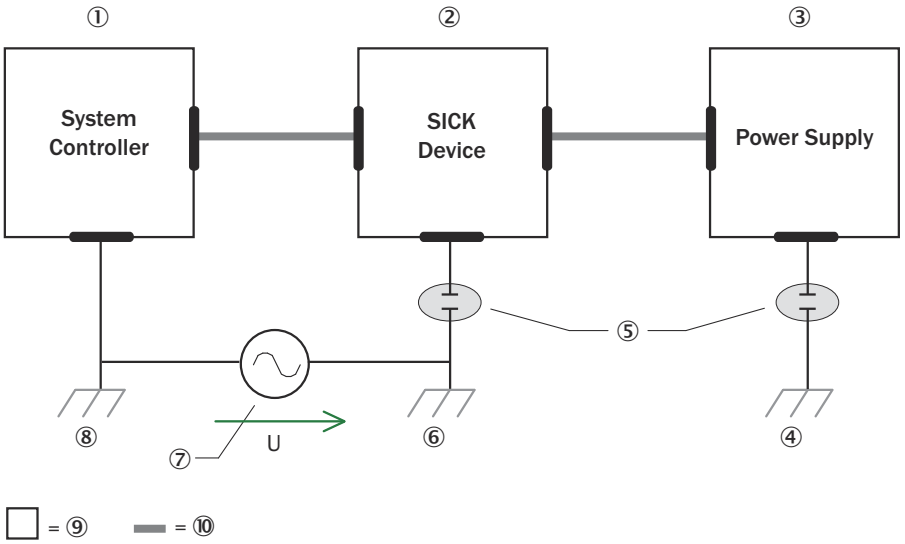


Figure 36: Example: Prevention of equipotential bonding currents in the system configuration by the insulated mounting of the device

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.

6.3 Connection diagram

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).

PWR connection

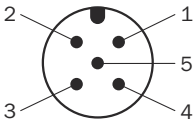


Figure 37: Male connector M12, 5-pin, A-coded

Table 9: Pin assignment for PWR connection

Pin	Identification	Description	Wire color, part number 2095733
1	Vs	Supply voltage: +10 ... +30 V DC	Brown
2	- –	Reserved	White
3	GND	Supply voltage: 0 V	Blue
4	IN8 / OUT8	Digital input 8 / digital output 8	Black
5	–	Reserved	Gray

1) Information only valid when using the specified open-ended connecting cable which is available as an accessory

Ethernet connection

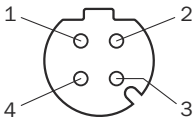


Figure 38: M12 female connector, 4-pin, D-coded

Table 10: Pin assignment for Ethernet connection

Pin	Identification	Description
1	TX+	Sender+
2	RX+	Receiver+
3	TX-	Sender-
4	RX-	Receiver-

Connection I/O

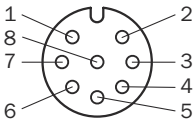


Figure 39: Female connector, M12, 8-pin, A-coded

Table 11: Pin assignment for I/O connection

Pin	Identification	Description	Wire color, part number 2134055 1)
1	IN1 / OUT1	Digital input 1 / Digital output 1	White
2	IN2 / OUT2	Digital input 2 / digital output 2	Brown
3	IN3 / OUT3	Digital input 3 / digital output 3	Green
4	IN4 / OUT4	Digital input 4 / digital output 4	Yellow
5	IN5 / OUT5	Digital input 5 / digital output 5	Gray
6	IN6 / OUT6	Digital input 6 / digital output 6	Pink
7	GND INx / OUTx	Ground for all digital inputs / outputs	Blue

Pin	Identification	Description	Wire color, part number 2134055 [1]
8	IN7 / OUT7	Digital input 7 / digital output 7	Red

1) Information only valid when using the specified open-ended connecting cable which is available as an accessory

6.4 Connection options

The following illustrations show several wiring examples based on the different types of inputs and outputs used.

Inputs:

- IN1-3: Field set switching
- IN4: EasyTeach teach-in

Outputs (maximum output current, "Technical data", page 56):

- OUT5: EasyTeach cut-off path (evaluation)
- OUT6: Cut-off path for subsequent evaluations
- OUT7: EasyTeach confirmed indicator (signals teach-in)
- OUT8: Device ready indicator

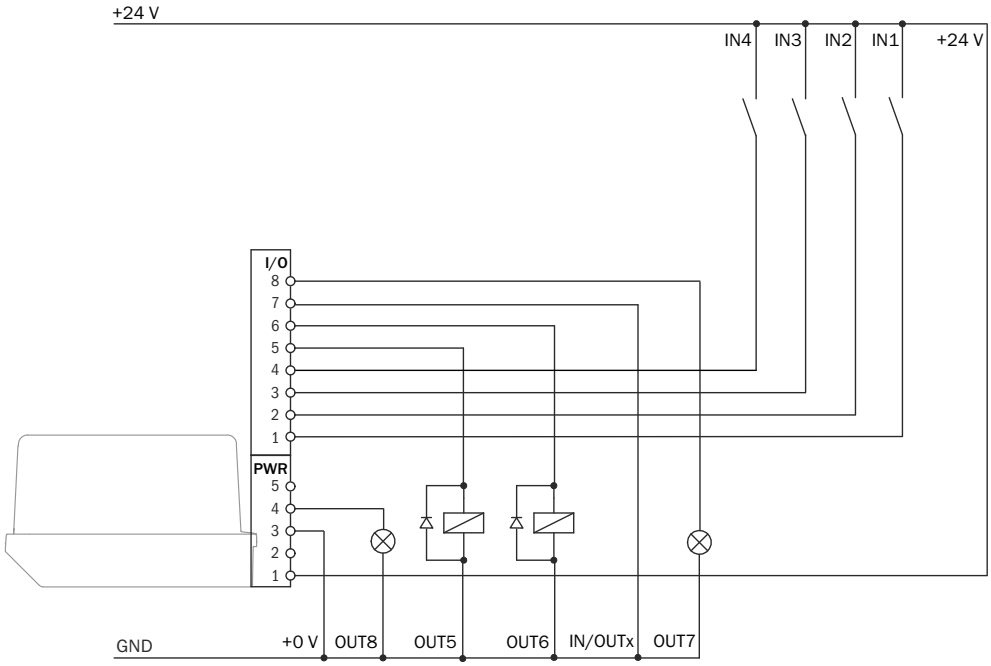


Figure 40: Connection example with PNP outputs and active high inputs (default)

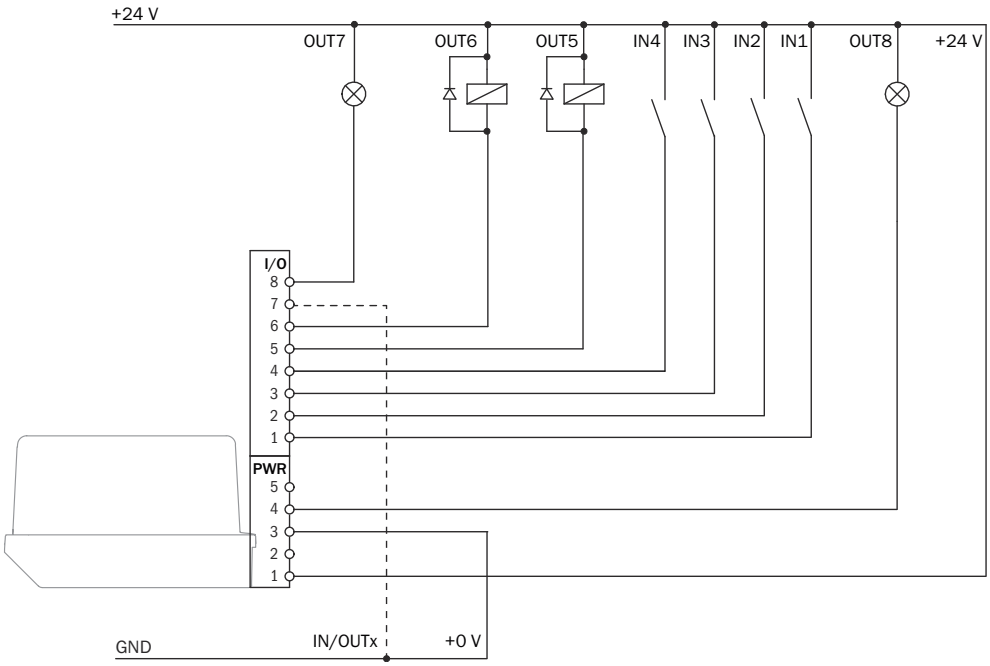


Figure 41: Connection example with NPN outputs and active high inputs

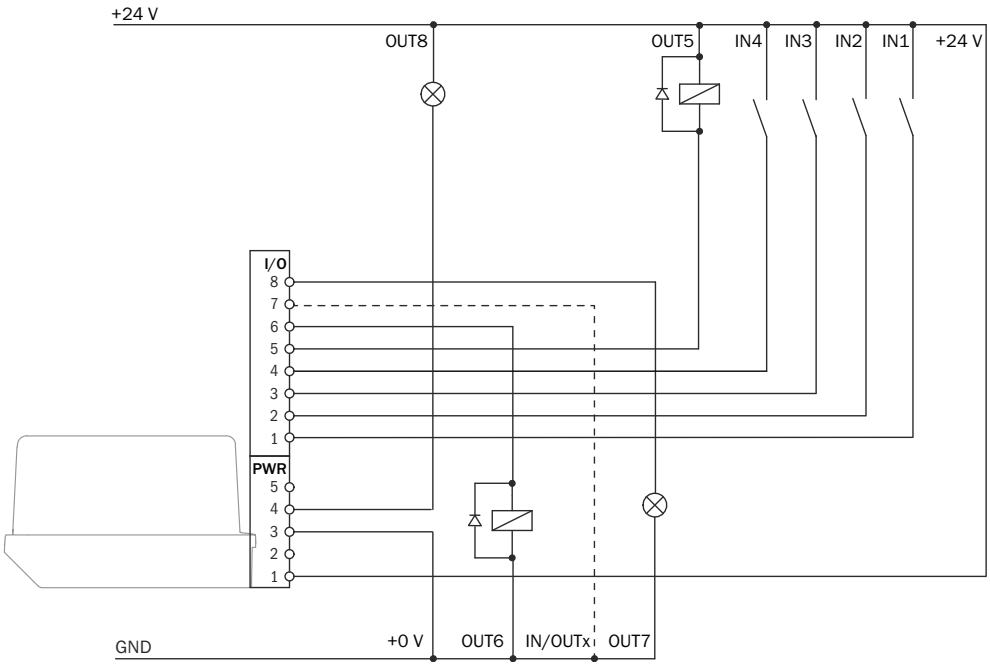


Figure 42: Connection example with push-pull outputs and active high inputs

6.5 Connecting the device electrically

NOTICE

Observe the wiring instructions (see "Wiring instructions", page 37) and prerequisites for safe operation (see "Prerequisites for safe operation of the device", page 38).

1. Ensure the voltage supply is not connected.
2. Connect the device according to the connection diagram, see "Connection diagram", page 41.
   Examples for the connection options, see "Connection options", page 43.

7 Operation

7.1 Operation with SOPAS ET

Version 4.1 and higher of the SOPAS Engineering Tool (SOPAS ET) software can be used to parameterize the device and for service and diagnostic purposes.

To configure the device, you will require a computer with SOPAS ET installed and a free Ethernet connection. Alternatively, the connection can be established via a USB connection using an Ethernet USB adapter.

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.

1. Connect the communication interface (Ethernet, 4-pin M12 female connector) of the device to the computer.
2. Switch on and start the computer.
3. Supply the device with voltage (5-pin M12 male connector, supply voltage 10 ... 30 V DC).

✓ After successful initialization, the two status LEDs light up green. The device is ready for use.

NOTE
To use SOPAS ET with the device, you need a device description file (SDD, SOPAS Device Description) for this device. You can install this within SOPAS ET using the device catalog. The device description file is saved on the device and can be installed there. Alternatively, installation is possible from the SICK website (Internet connection required).

Following installation of the device description file, the device can be selected from the device catalog and added to a project.

A connection to the device is established via the communication interface. The connection must be activated for data transmission (**online**).

Certain functions (e.g., Edit parameters) require you to be logged in to the device (**Device** > **Log In** menu, **User Level**: Authorized customer, **Password** (factory default): client).

Table 12: Passwords

User levels	Password
Maintenance	main
Authorized client	client
Service	service level

NOTE
Change the passwords during initial commissioning to protect your device.
A higher user level can change the password of a lower user level.

The factory-assigned passwords can be changed in the device window (**Device** > **Change password** menu).

Information about the device is displayed in the device window and the device can also be configured here (**Device**> **Open** menu).

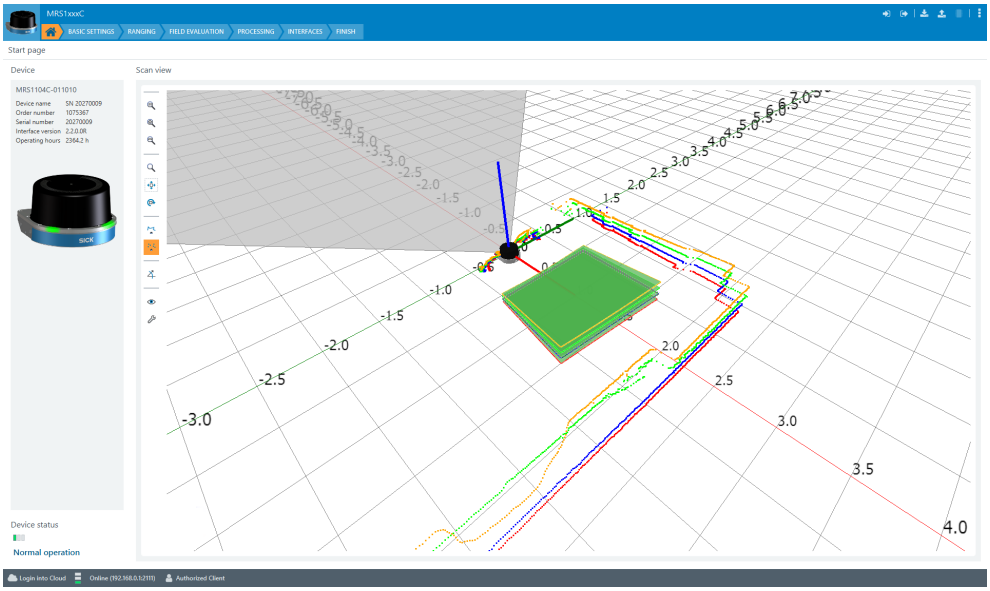
7.1.1 Description of the device window

Overview

The device window features various views supporting a selection of functions.

Start

Start device configuration with display of information about the device (left-hand side) and display of the current scan (right-hand side).



The scan view can be customized using various tools (to the left of the scan view), for example:

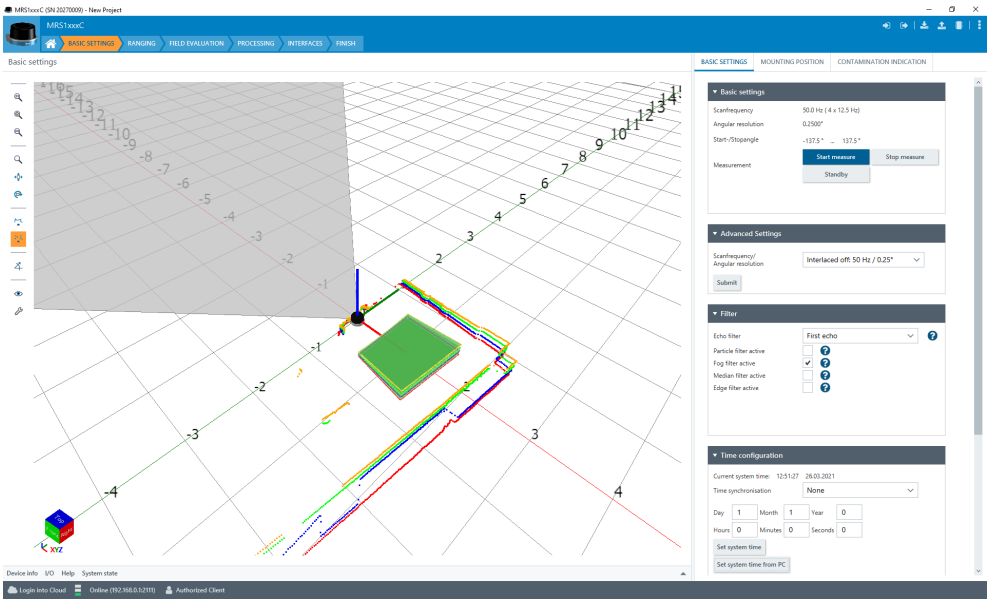
: Resets scan display to default view.

/ : Displays measured values as points or lines.

: Freely rotates scan display.

Defaults

Scan display (left-hand side), entry of the device parameters as well as display of the IMU (Inertial Measuring Unit) data, entry of the mounting position as well as parameterization of the contamination indication (right-hand side):



1. **Setup mode**: Optionally activate one of the four scan layers used for alignment.

Standby via input: The device can be put into standby mode by switching a digital input. This function can be activated in the basic settings. The input to be evaluated can be selected there.

IMU data: The device is equipped with an inertial measuring unit (IMU). This can be used to output the actual inclination of the device. The IMU can output accelerations in X, Y and Z as well as the position angle in yaw, pitch and roll. These data can also be used to display and evaluate the mounting position.

Temperature configuration: A definable temperature range makes it possible to monitor the internal device temperature. If the temperature exceeds or falls below this range, an output is activated. The **Temperature alert** source can be selected for the output on the **Interfaces** tab and evaluated.

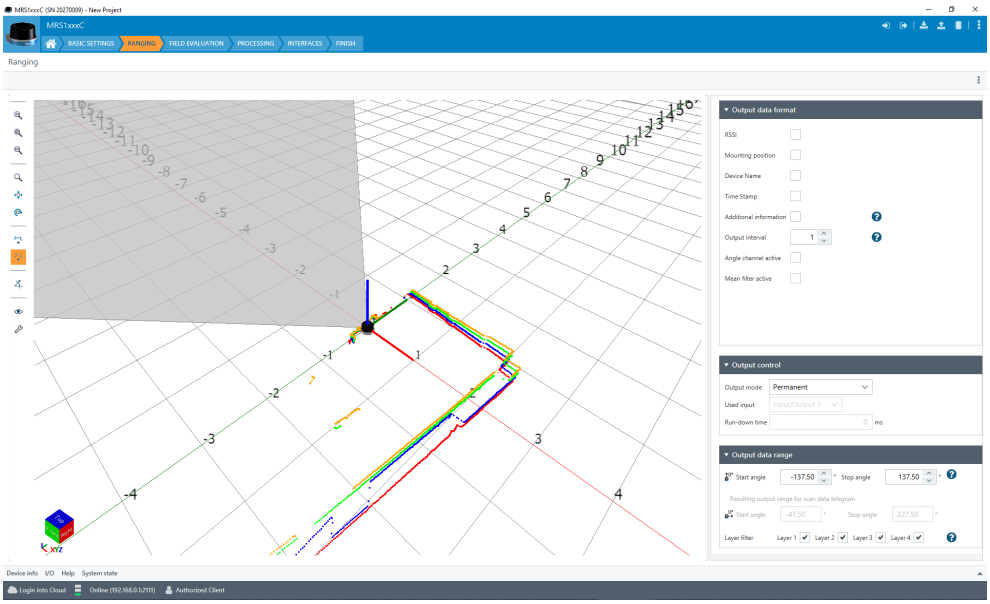
IMU sabotage detection: Reliably detects when the device deviates from a previously defined or taught in position. The reference position can be set manually or applied automatically via the **Teach-in position** function.

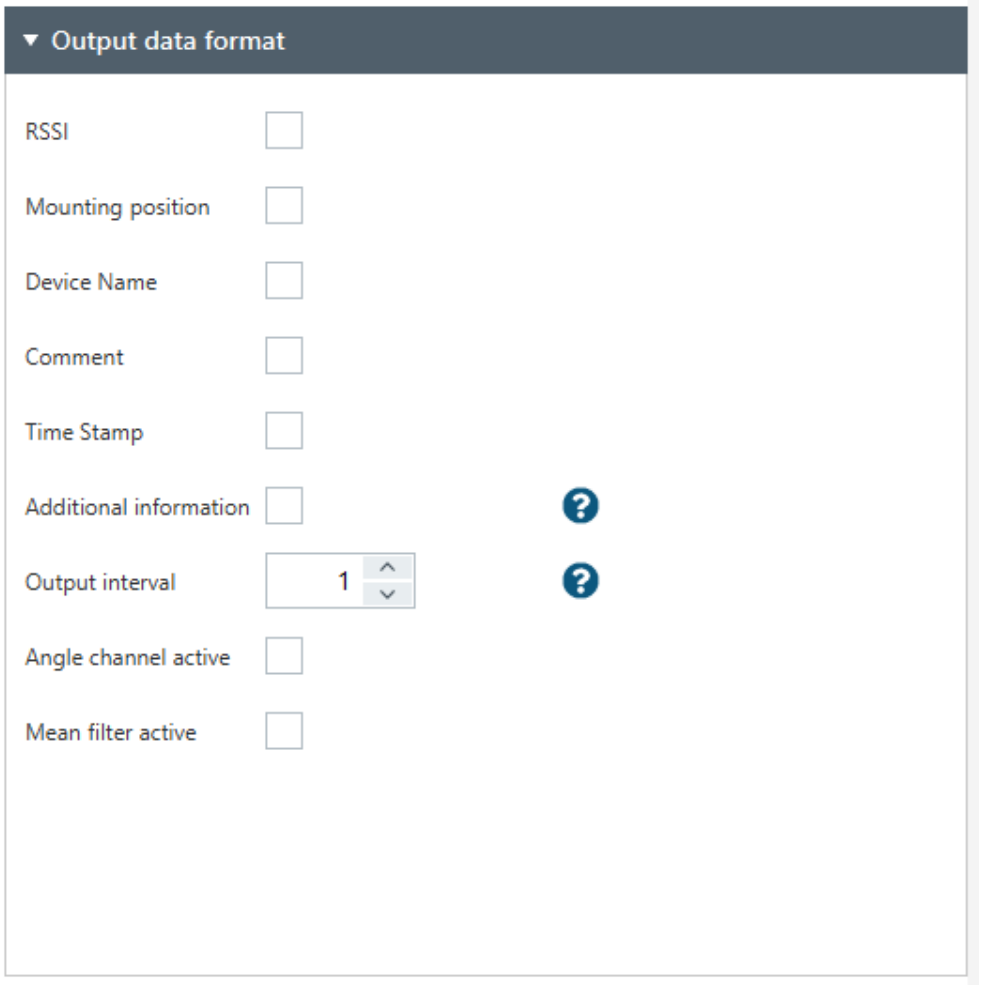
If the device goes outside the defined area including the set tolerance, an output is activated. The **Sabotage detection** source can be selected for this output on the **Interfaces** tab and evaluated appropriately.

To avoid false alarms, it is recommended to select a tolerance that is larger than the maximum expected movement within the application.

Distance measurement

Input fields for output data format, output control, and output range (right-hand side).





Angle channel active: The motor of the device rotates at 12.5 Hz and achieves a scan rate of 50 Hz due to the 4 modules. Assuming that the motor rotates at a constant and uniform speed at all times, a measured value is generated every 20 ms at an angular resolution of 0.25°. The motor speed is subject to minor fluctuations, however,

for hardware limitation reasons. Due to external influences (e.g., strong acceleration or abrupt deceleration of the device), these fluctuations can amount to several tenths of a degree which can result in deviations in the centimeter range when measuring objects. The angle channel enables these fluctuations to be corrected to obtain the true angular position. When activated, an additional angle correction value is output in the measurement data telegram for each measurement point. Each value describes the deviation from the implicitly assumed angular position in a range of ± 3.2768° and is accurate to about ± 0.01°. Since the channel can only output positive numbers, the value 32768 (8000h) corresponds to a deviation of exactly 0°. At larger values, from 32,769 to 65,535 (8001h ... FFFFh), the motor ran faster (objects appear narrower). At smaller values, from 0 to 32,767 (0h ... 7999h), the motor ran slower (objects appear wider).

Supplementary information: Ambient light indication: The supplementary information bits can output an ambient light indication using the `LMDscandata` telegram (AINF 1). Bits 3 to 5 in the 8-bit channel specify the intensity of the ambient light.

Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1
0	0	0	0	0	0	0	0

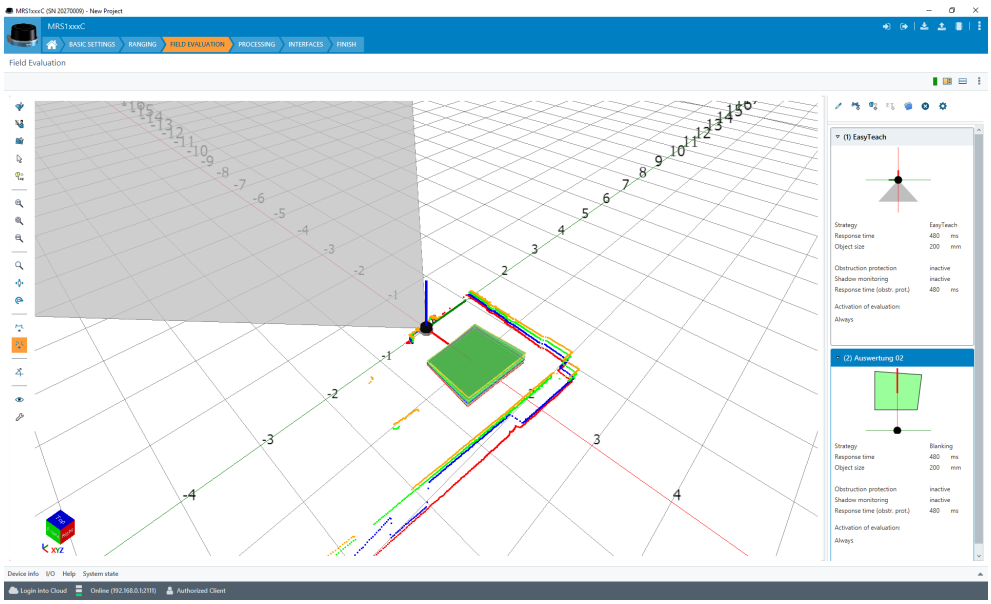
Example: `AINF 1: 000`**`101`**`00`. The ambient light is determined for the 0° layer. The higher the transmitted value, the stronger the ambient light. The three bits highlighted in bold can range from '000' (very low) to '111' (very strong).
A threshold for switching a digital output can also be set. If this value is reached or exceeded in the scan, the output (ambient light) is switched.

Options

- **Single value:** The output is switched as soon as a single measurement value reaches or exceeds the threshold.
- **Scan average:** The output is switched if the average of all measured values over a complete scan reaches or exceeds the threshold.

Field evaluation

Scan display for adapting the field geometry (left-hand side) and entering detection parameters (right-hand side):



The **Edit field points** tool (mouse pointer in the list of tools to the left of the scan view) can be used to customize the field geometry using the mouse pointer. Alternatively, the field vertices can also be customized by entering the coordinate values in the **Field coordinates** mask.

Table 13: Options for editing the field geometry using the mouse pointer

Function	Procedure
Scale field	Drag square red handle.
Rotate field	Drag round black handle.
Customize field shape	Drag square green handle.
Add field vertices	Double-click in required position.
Remove field vertex	Click to select square green handle and press the **[del]** key.
Delete field	Press the **[del]** pushbutton without first selecting a green handle.
Insert new field (function is only available if a field has not been created)	Select the **Insert field points** tool, click to set field vertices, and double-click the last field vertex to close the field.

NOTE

Fields are defined on the y global plane.

Depending on the mounting location of the device (specified in the Default view), the fields are then projected onto the individual scan planes (layers).

The following options for evaluation switchover are available:

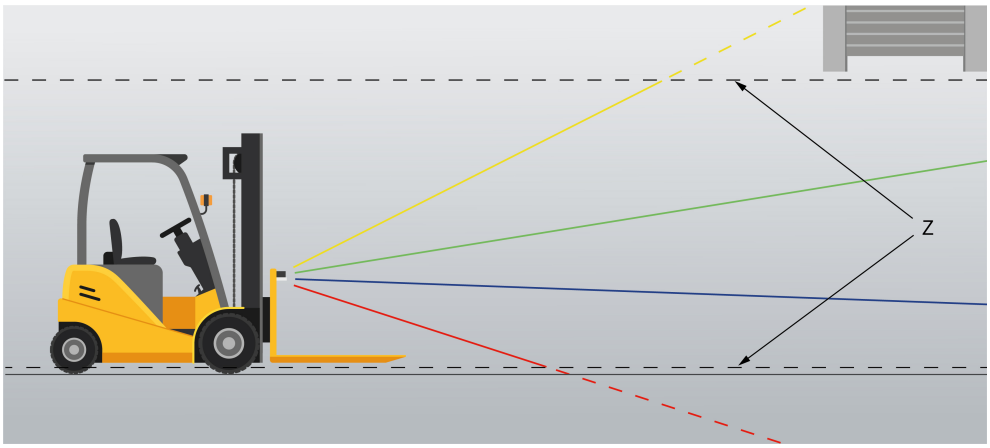
- Always
- Via digital input
- Time
- Telegram for FW. 2.3.0.0 and above
- Deactivated

For switchover via telegram (`ActivateEvaluation` in the Telegram Listing see "Set activate evaluation [sMN ActivateEvaluation]", page 159).

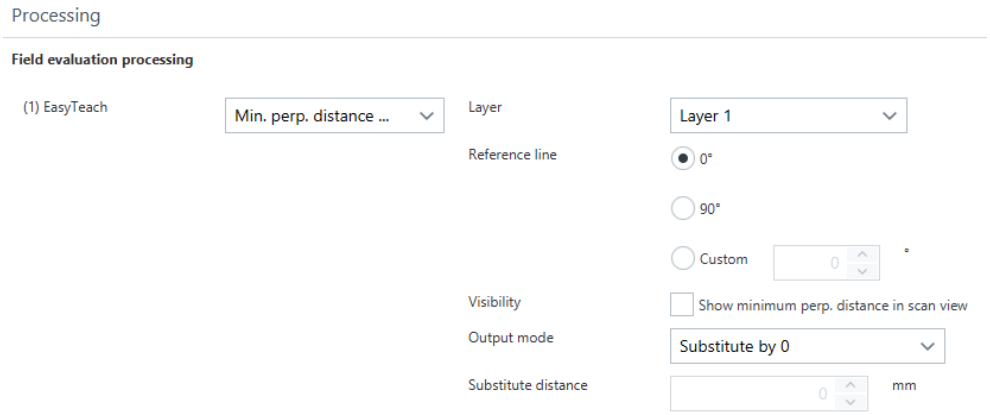
NOTE

Monitoring of case switching (expert function) : The behavior of the sensor when switching detection fields can be customized.

Limitation in Z direction limits the monitored area considered for field evaluation. The input values define the upper and lower limits of a monitoring corridor. Objects that are inside a field but outside the defined corridor will not trigger detection in the field. This feature can be useful in applications where surrounding structures, such as ceilings, can be detected in the configured fields but should not result in a response.



Processing

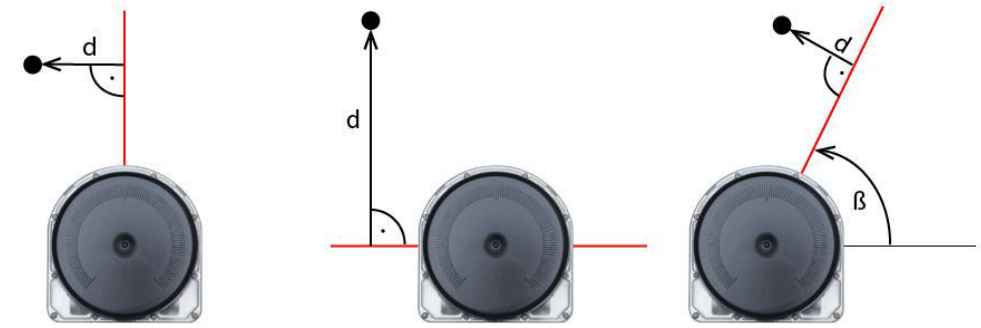


NOTE

To obtain the data, a field needs to be created. The data are only output when an object detection occurs. This is configured on this tab.

Further input fields for field evaluation processing (e.g., minimum perpendicular distance to reference line, direction detection).

The functions **Minimum perpendicular distance** and **Maximum perpendicular distance** can be used to determine the minimum or maximum distance of an object that triggers an object detection.



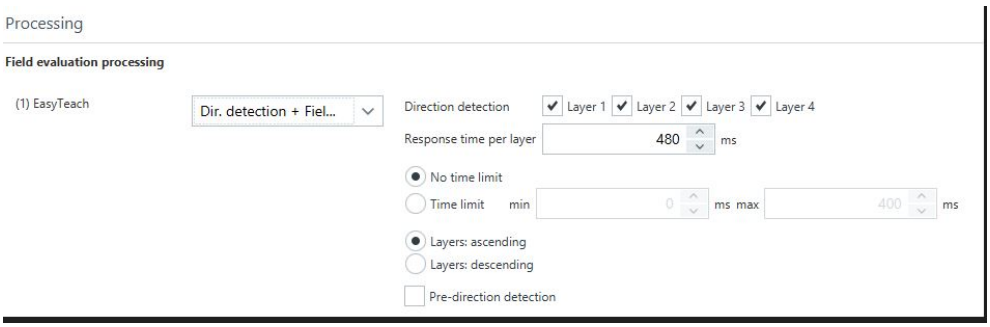
Reference line 0° ① Reference line 90° ② Reference line custom ③

① Reference line 0°

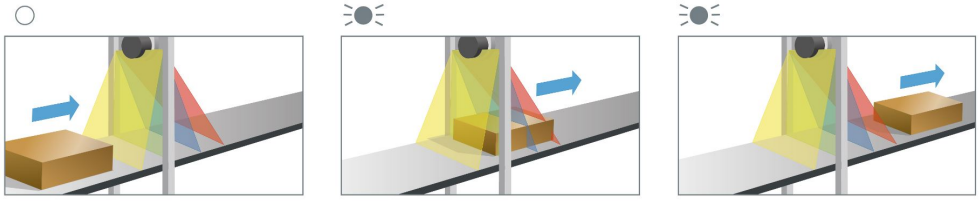
② Reference line 90°

③ User-defined reference line

Available via telegram `LFEperpdistresult` in the Telegram Listing see "Request minimal and maximal perpendicular distance once [sRN LFEperpdistresult]", page 154.

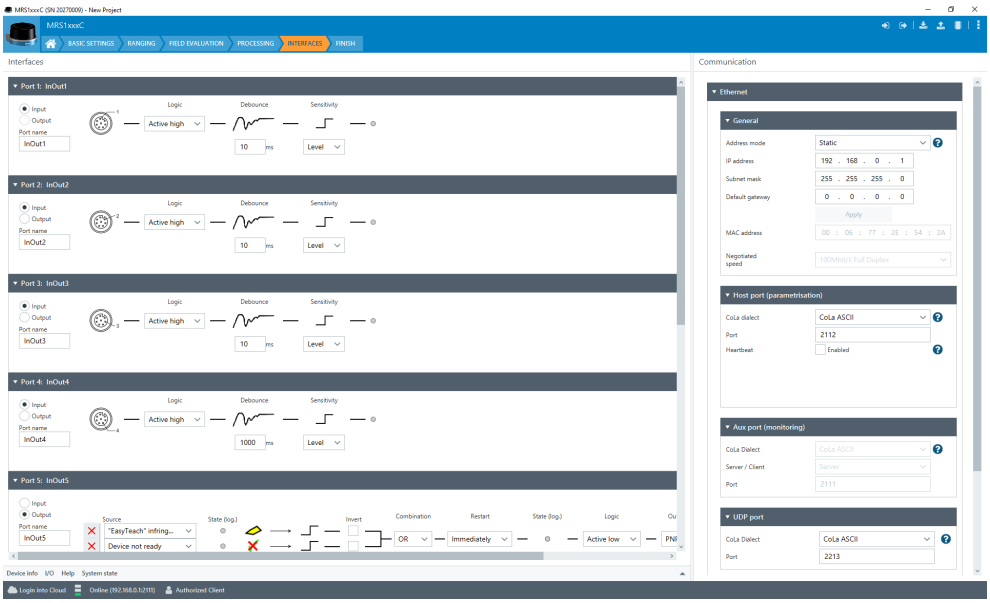


With **Direction detection**, the output is switched as soon as an object has passed the last layer and the measuring range is free again. With **Premature direction detection**, the direction is already output while the object is still in the measuring range.



Interfaces

Configuration of the interfaces (left-hand side) and input of communication parameters (right-hand side).

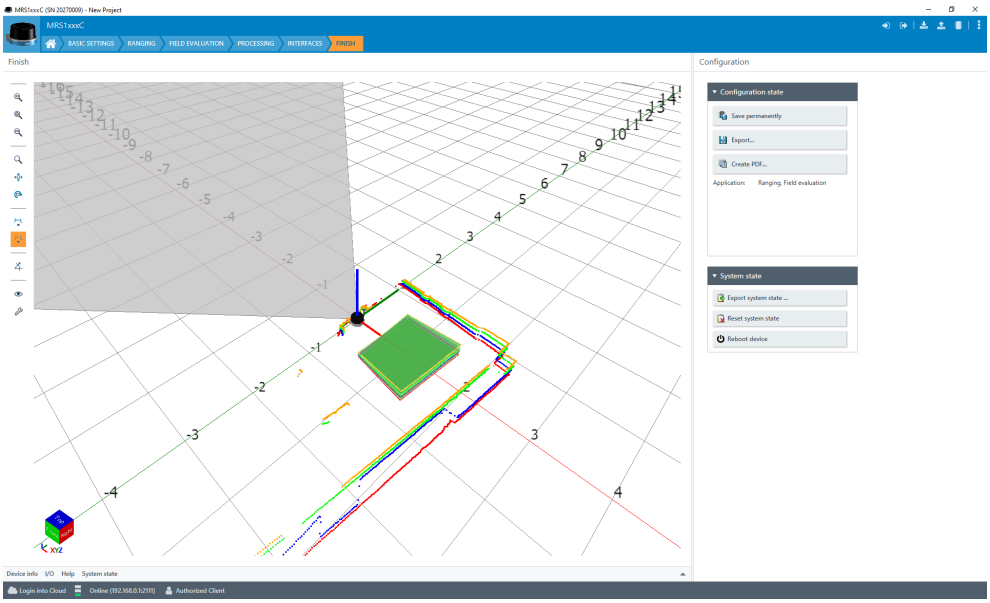


Display: Defines the activation and function of the LEDs

- On: LEDs are always switched on
- Sopas command: LEDs can be controlled using SOPAS commands
- Digital input: LEDs can be activated by means of a digital input
- Off: LEDs are always switched off

Conclusion

Scan display (left-hand side) and buttons for finalizing the device parameterization and for system status/system diagnosis.



NOTE

Changes to parameters that are made in SOPAS ET are not saved automatically in the device. After you have completed the configuration, you must save it in the device permanently by pressing the **Save permanent** button.

NOTE

For examples of using the software and the configuration options in the device window, refer to the annex, see "Configuration with computer", page 65.

7.2 Operation with teach-in (EasyTeach)

For the field evaluation application, it is possible to configure the device using the field contour teach-in (EasyTeach) function rather than configuring it via SOPAS. The teach-in phase can be activated / deactivated via an input circuit or SOPAS.

NOTE

For an example of configuring the device using EasyTeach and for an input circuit, refer to the annex, "Configuration without computer", page 64.

7.3 Opening the web server user interface (SOPASair)

Before opening the user interface, perform the following work steps:

- Connect the device to the computer via Ethernet.
- Set up the voltage supply for the device.
- Ensure that the computer and device are located in the same network.
- Ensure that the computer uses a different IP address than the device, but is in the same IP address range (e.g. 192.168.0.xxx)

Open the user interface:

1. Open web browser (recommendation: Google Chrome).
2. Enter the device IP address into the address line. The standard IP address is: 192.168.0.1

✓ The SOPASair user interface is displayed.

8 Maintenance

8.1 Maintenance plan

During operation, the device works maintenance-free.

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

Table 14: 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 that all unused connections are sealed with protective caps.	Depends on ambient conditions and climate. Recommended: At least every 6 months.	Specialist

8.2 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.

► For MRSxxxxx-9xxxxxS05 (Heavy Duty): If possible, cover the type label during cleaning.

9 Troubleshooting

9.1 General faults, warnings, and errors

Possible errors and corrective actions are described in the table below. In the case of errors that cannot be rectified using the information below, please contact the SICK Service department. 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 telephone processing.

Table 15: Troubleshooting questions and replies

Question / status	Response / remedial actions
Both LEDs flash red.	Device fault: Read out the fault code via the PC software SOPAS ET and remedy the cause of the fault.
LEDs indicate an undefined status.	Check the device status, if necessary contact the SICK Service department.
Measurement data show anomalies.	Optics cover contaminated: Clean the optics cover.

9.2 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.

9.3 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

9.4 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

Measurement principle	HDDM+
Application	MRSxxxxx-0xxxxx: Indoor MRSxxxxx-1xxxxx: Outdoor MRSxxxxx-9xxxxxS05: Outdoor
Light source	Infrared (wavelength 850 nm; max. output power 1.26 W; pulse duration 3.5 ns; average power 4.0 mW)
Laser class	Laser class 1 (EN 60825-1:2014+A11:2021, IEC 60825-1:2014, EN/IEC 60825-1:2007) 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.
Horizontal aperture angle	275°
Vertical aperture angle	7.5° (over 4 measurement levels)
Scan field flatness	Conical error: ± 0.6° Tilt: ± 0.6°
Scanning frequency	50 Hz, 4x 12.5 Hz
Angular resolution	Horizontal: 0.25°, 0.125° (interlaced mode), 0.0625° (interlaced mode) Vertical: 2.5°
Heating	Self-heating
Working range	0.2 m ... 64 m
Scanning range with 10% remission	16 m
Scanning range with 90% remission	30 m
Spot divergence	10.4 mrad x 8.7 mrad
Light spot size at front screen	7 mm rounded
Number of echoes evaluated	3

Working range diagram

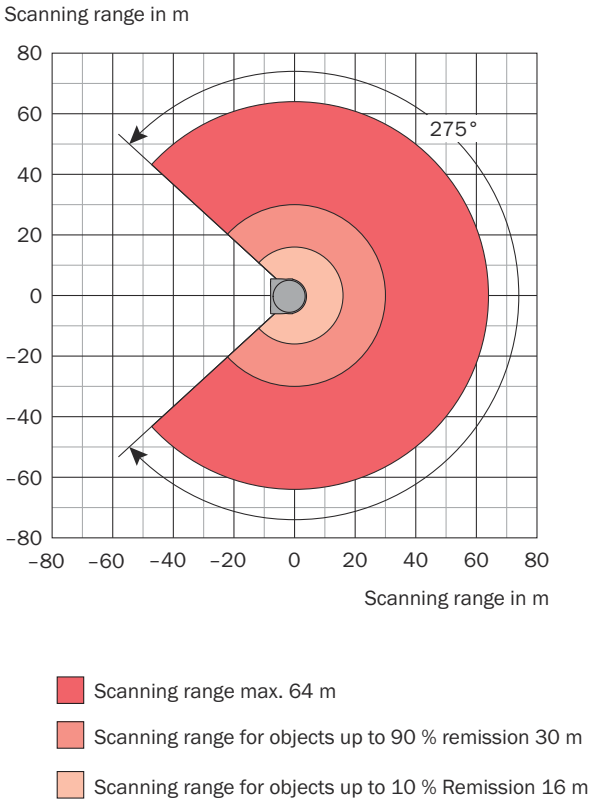


Figure 43: Working area diagram, top view

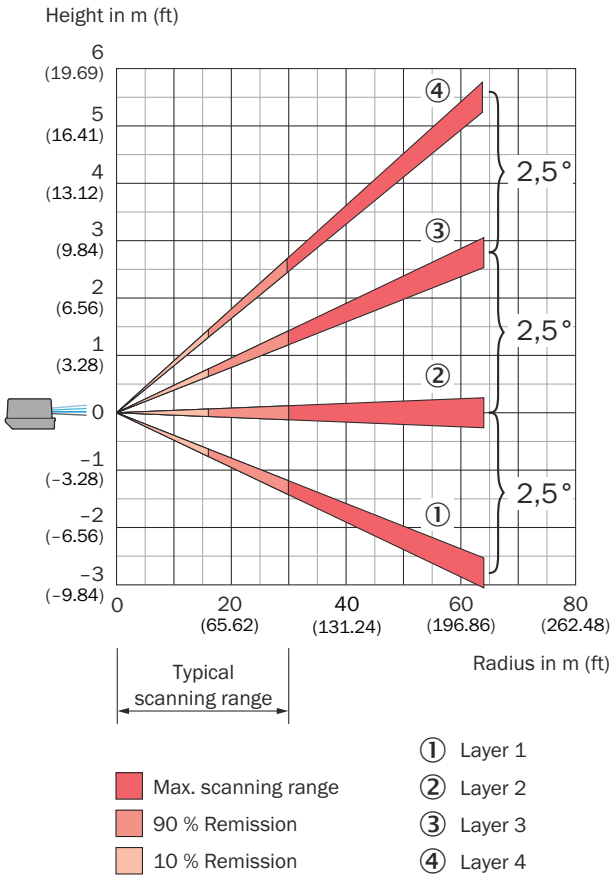


Figure 44: Working area diagram, side view

10.2 Mechanics/electronics

Connection type	3 x M12 round connectors with swivel connector, cast zinc lacquered MRSxxxxx-9xxxxxS05 (Heavy Duty): Stainless steel 1.4404
Supply voltage	10 V DC ... 30 V DC [1)]
Permissible residual ripple	±5%
Power consumption	P_{typ} = 13 W P_{start} = 30 W for 1 s (motor start-up) P_{max} = 37 W (with full specified current at all outputs)
Housing	ALSi12 Optics cover: PC
Housing color	MRSxxxxx-xxxxx0: Light blue (RAL 5012) MRSxxxxx-xxxxx1: Gray (RAL 7042)
Enclosure rating	MRSxxxxx-0xxxxx (indoor): IP65 (IEC 60529:1989+AMD1:1999+AMD2:2013) MRSxxxxx-1xxxxx (outdoor): IP65 / IP67 (IEC 60529:1989+AMD1:1999+AMD2:2013) MRSxxxxx-9xxxxxS05 (Heavy Duty): IP65 / IP67 (IEC 60529:1989+AMD1:1999+AMD2:2013) / IPX9K (ISO 20653)
Protection class	III (IEC 61140:2016-11)
Electrical safety	IEC 61010-1:2010-06
Weight	1.2 kg

Dimensions (L x W x H)	151.9 mm x 150 mm x 92.5 mm
Encoder input frequency	not available
Maximum output current	Max. 100 mA per output [2]

1) The external voltage supply must be able to bridge a brief power failure of 20 ms as specified in EN 60204. The absolute voltage level must not drop below the specified minimum voltage. Suitable power supply units are available as accessories from SICK.
2) E.g. 30 V*100 mA = 3 W

10.3 Dimensional drawing

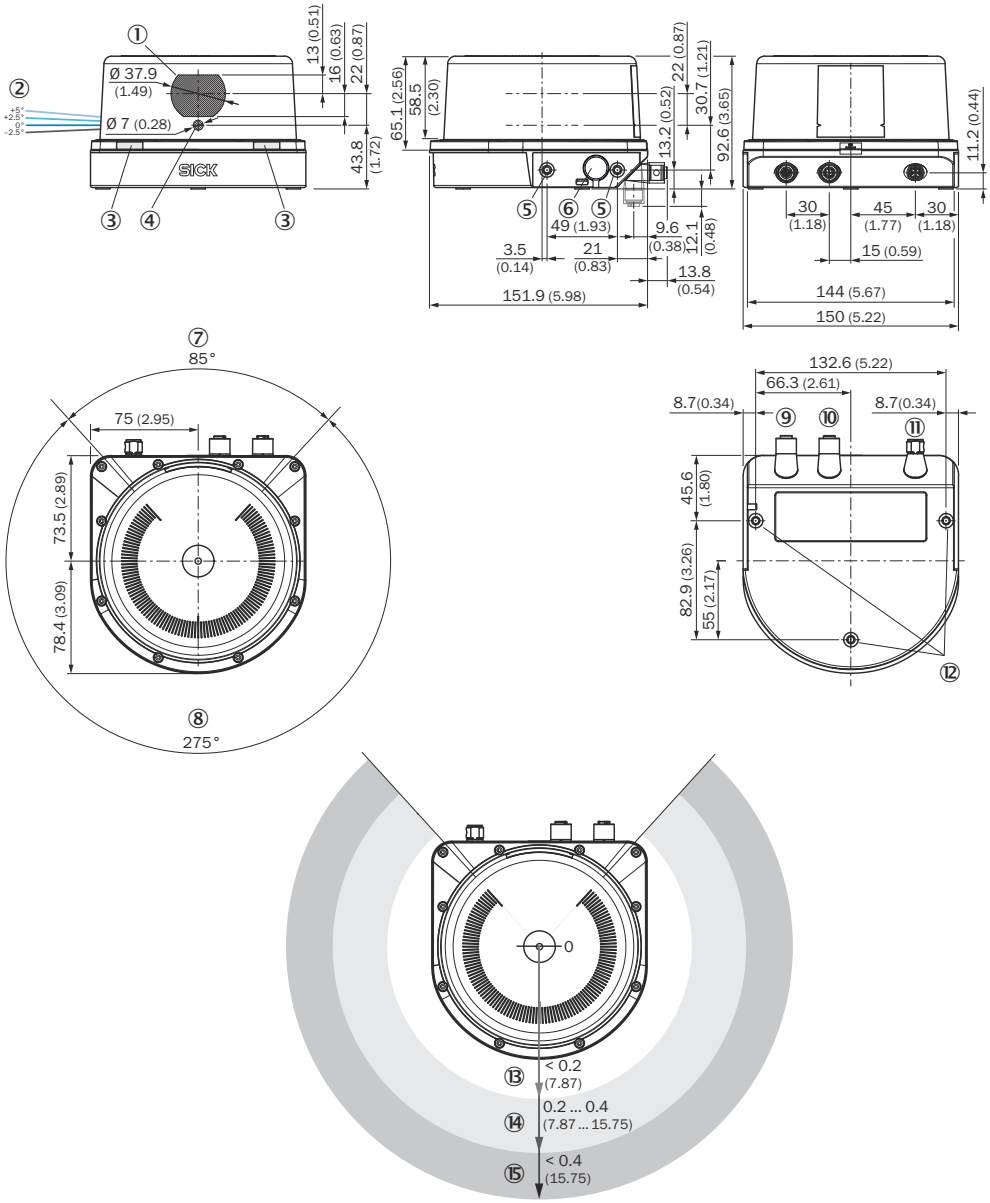


Figure 45: MRS1000 device structure and dimensions, dimensions in mm

① Receiver
② Laser aperture angle scan layers 1 to 4
③ Status LEDs
④ Sender
⑤ M5x7.5 fixing holes
⑥ Ventilation element
⑦ Blind spot
⑧ Field of view
⑨ Ethernet connection
⑩ I/O connection
⑪ PWR connection (supply voltage)
⑫ M5x7.5 fixing holes
⑬ Close range (no detection or measurement possible)

⑭ Detection area
⑮ Measuring range

10.4 Performance

Scan/frame rate	55000 ... 165000 measuring points/second
Response time [1]	Typ. 20 ms (4 scan layers), typically 80 ms (1 scan layer) Min. 7 ms, max. 84 ms + 1 scan
Power-up time	approx. 35 s
Systematic error	± 60 mm Temperature drift: typically ± 0.5 mm/K
Statistical error	≤ 30 mm
Integrated application	Integrated field evaluation with flexible fields on 4 scan players, data output
Number of field sets	Up to 64 fields (16 fields can be configured for each of the 4 scan layers) Up to 128 fields (32 fields can be configured for each of the 4 layers) [2]
Simultaneous analysis cases	Up to 16 evaluations
Filters	Fog filter, particle filter, mean filter, median filter, ground reference evaluation, edge filter, echo filter

1) Depends on the object size and selected filter settings.
2) MRS1104C-111011S02 only (part no.: 1106288)

10.5 Interfaces

Ethernet	✓, TCP/ IP, UDP/ IP Function: HOST, OPC, NTP Measurement data output (distance, RSSI, supplementary data) Data transmission rate: 10/100 Mbit/s
digital inputs/outputs	I/O (8 (multiport))
Digital inputs	Input voltage: V_s- 30 V ... +30 V (V_s: device supply voltage) **Static input current** • Logical “Active HIGH” (PNP compatible): Typ. 0.3 mA, Max. 1 mA • Logical “Active LOW” (NPN compatible): Typ. -0.3 mA, Max. -1 mA **Level for HIGH** • V_s ≥ 18 V: +12.5 V ... +30 V • V_s < 18 V: 0.75*Vs ... +30 V (at 12 V this means: 0.75*12=9 V → 9 V ... 30 V) **Level for LOW** • Vs ≥ 18 V: Vs-30 V ... +10.5 V (at 24 V this means: 24 V - 30 V= -20 V → -20 V ... +10.5 V • V_s < 18 V: V_s-30 V ... 0.4 * V_s Input capacity: 2.5 nF

Digital outputs	**Typ. Output voltage** • LOW: <= 1 V DC • HIGH: >= V_S- 1 V (V_S= device supply voltage) **Max. switching current per pin under continuous load/current limiting (after 5 ms)** • Max. switching current: 100 mA • Current limiting: 200 mA • Internal resistance: 2.3 Ohm ... 4.6 Ohm • Max. power loss: 1.5 W (total for digital inputs/outputs)
Output data	Contamination indication, IMU (secondary sensor data)
IMU (inertial measuring unit)	BOSCH BHI160B with 3-axis gyroscope and 3-axis accelerometer Sampling rate: 100 Hz **Relative position of the IMU to the optical origin:** • X: -17 mm • Y: +23.5 mm • Z: -30 mm
Optical indicators	2 LEDs
Configuration software	SOPAS ET, web server (display)

10.6 Ambient data

Remission factor	2% ... > 1,000% (reflector)
Electromagnetic compatibility (EMC)	EN 61000-6-2:2005 EN 61000-6-3:2007+A1:2011
Vibration resistance	According to IEC 60068-2-6:2007-12: 10 Hz ... 150 Hz: amplitude 0.35 mm to 5 g, 20 cycles
Shock resistance	Single shock according to IEC 60068-2-27:2008-02: 15 g, 11 ms, 6 shocks per axis Continuous shock according to IEC 60068-2-27:2008-02: 10 g, 16 ms, 1000 shocks per axis
Harmful gas resistance	Test for determining the resistance to flowing mixed gas according to DIN EN 60068–2–60:2016–06, Ke test, method 1
Chemical resistance	Salt spray test: DIN EN IEC 60068-2-52 (2018/08), test method 4
Ambient operating temperature	MRSxxxxx-0xxxxx: -10 °C ... +50 °C MRSxxxxx-1xxxxx: -30 °C ... +50 °C MRSxxxxx-9xxxxxS05: -30 °C ... +50 °C
Storage temperature	-40 °C ... +75 °C
Operating and storage air humidity	Max 90% air humidity (non-condensing)
Ambient light immunity	80 klx

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 Licenses

SICK uses open source software which is published by the rights holders under a free license. Among others, the following license types are used: GNU General Public License (GPL version 2, GPL version 3), GNU Lesser General Public License (LGPL), MIT license, zlib license and licenses derived from the BSD license.

This program is provided for general use without warranty of any kind. This warranty disclaimer also extends to the implicit assurance of marketability or suitability of the program for a particular purpose.

More details can be found in the GNU General Public License.

For license texts see www.sick.com/licensetexts.

Printed copies of the license texts are also available on request.

12.3 Configuration without computer

12.3.1 Field evaluation

Configuration is performed by teaching in (EasyTeach) of the surrounding contour in order to automatically generate a field with any shape, including more complex shapes. In the as-delivered configuration, the teach-in phase is activated and deactivated using the wiring for input IN4.

Preparing EasyTeach

- Remove all objects that will not permanently be in the field of view in monitoring mode later on.
- Distance yourself sufficiently from the device during the teach-in phase and do not enter the monitored area, so that you are not detected as part of the field contour.

Teaching in the field contour

- The device creates the field from the surrounding contour with a negative offset of 250 mm.
- The field shape to be formed can be defined by pacing out the limits during the teach-in phase. It is important to make sure that the person can always be unequivocally detected.
- The device stores the shortest value measured during the teach-in phase as a field limit for each angle.
- Note: A parameter upload is required in order to display the newly taught-in field contour in SOPAS.

1. Install the device.
2. Carry out the EasyTeach wiring.

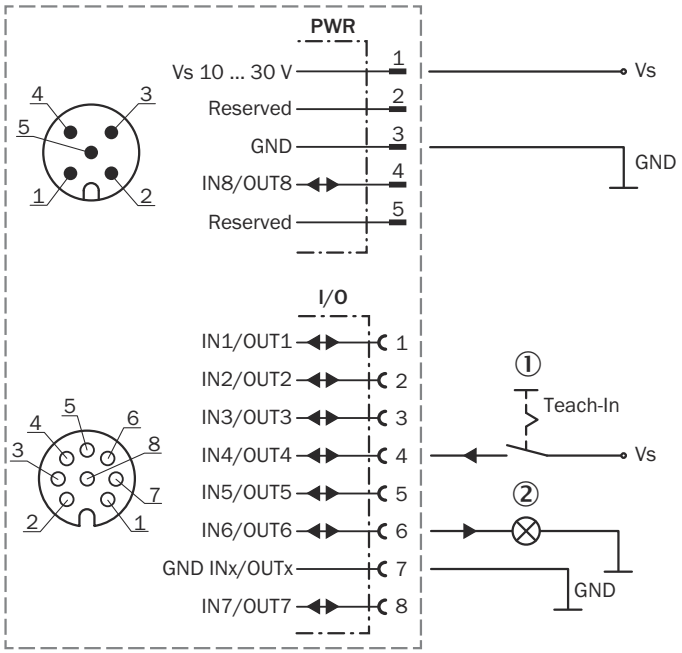


Figure 46: Wiring for EasyTeach (example)

① Teach-in via an external signaling device such as a pushbutton, at input IN4/OUT4 in this example.

② Feedback for object detection; At output IN6/OUT6 in this example

3. Set the EasyTeach input signal to start the teach-in phase (active high).
4. Define the field (e.g., using the existing surrounding contour or by pacing out the field).
5. Remove the EasyTeach input signal to end the teach-in phase (active high).
6. Thoroughly check correct detection of the field borders by performing a functional check in which For this purpose, deliberately induce object detections.

12.4 Configuration with computer

The SOPAS ET configuration software (from Version 4.1) is used as standard to display the surrounding contour (measuring line) recorded by the LiDAR sensor, as well as diagnostic information in the event of an error.

12.4.1 Output of measured values via terminal program

Example commands for output of measured values

If the LiDAR sensor receives one of the two example commands by telegram through the Ethernet interface, it will start the output of measured values through this data interface.

The detailed structure of the output telegram and the process of request and output is described in the Telegram Listing see "Measurement output telegram", page 100.

Note:

For the content of the response telegram for the measured value request, see "Send data permanently [sEN LMDscandata]", page 104. The telegram consists of one part with information on the configuration of the device and the time stamp, one part with measured data, and a status part with information on the device status and the statuses of the inputs/outputs.

After starting the measurement mode, the device needs some time to reach status 1 (“ready”). You should therefore query the status of the device using the **sRN SCdevicestate** telegram.

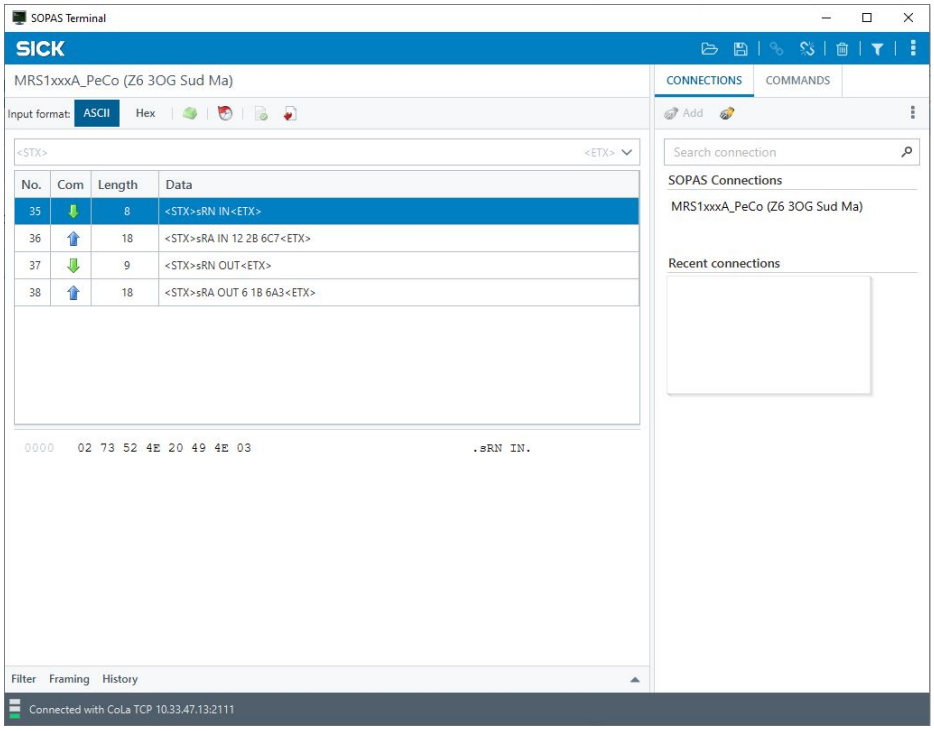
Then request measured data via telegram at the data interface from which you want to receive measured data. This can be achieved in two ways:

- 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 1** telegram – measured data is transmitted until the output of measured values is stopped with the **sEN LMDscandata 0** telegram.

Activating the output of the measured values in SOPAS on a test basis:

1. Start the terminal emulator: **Tools** menu > **Terminal**.
2. In the dialog window in the **Connections** menu, select the **Create new connection** function.
3. In the connection wizard, select the communication interface (**Ethernet**) and connection settings (default IP address: 192.168.0.1) and establish the connection (**Finish** button).
4. Enter in the input line one of the two example telegrams from the annex as they appear (automatically framed by **STX** and **ETX** when sending in the default setting). Pay attention to blank characters in the string.
5. Use the button to transmit the telegram to the LiDAR sensor.

✓ The LiDAR sensor responds by providing the data as a one-off or continuously in the display area of the terminal emulator. The data is output in the following order: Scan layer 2, 3, 1, 4 (see "Features", page 56 “Working range diagram, side view”).



Data output format of the measured values

The data output format per scan is comprised of the configuration information, measured values, radial distance, RSSI, scan layer assignment, device and status information, and time stamp.

In the default settings, the distance is output as a measured value (in mm).

To output the reflectance values in the telegram: Open the device window (**Device**> **Open** menu) and, in the **Distance measurement** view, select the **RSSI** check box under **Output data format**.

12.4.2 Measured value display and configuration via start window

► Open start window: **Device** > **Open** menu.
✓ The **Start** view with information on the **device**, **device status**, and the graphic **scan display** opens.

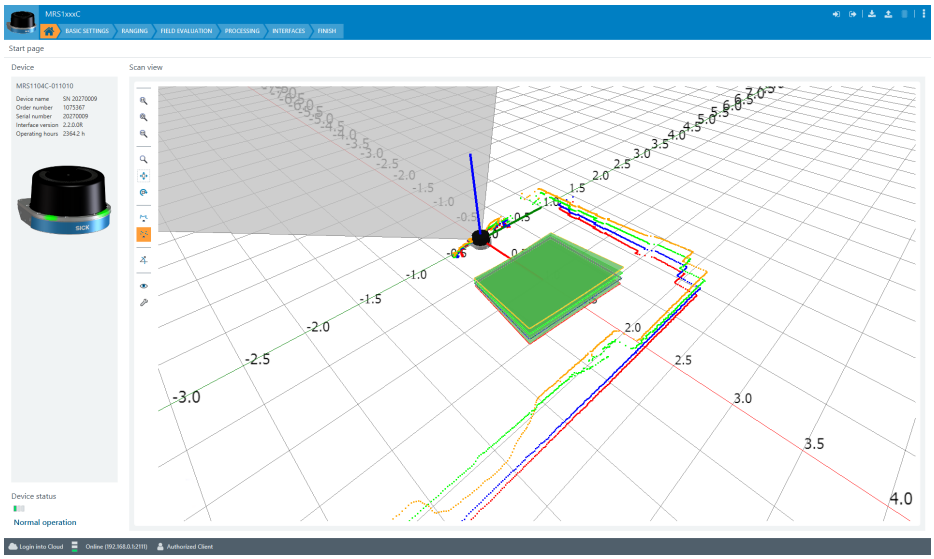


Figure 47: SOPAS ET: Start window

12.4.3 Selecting the application

1. Open the **Application** view.
2. Select **Field evaluation** or **Measurement data output**.

12.4.4 Field evaluation

Creating a new evaluation

► Click the button.

Entering evaluation parameters

► Click the button and enter evaluation parameters.

Inserting field points

1. Click the button.
2. Insert field points by clicking.
3. Double-click to close the field.

Editing field points

The Edit field points tool () can be used to customize the field geometry with the assistance of the mouse pointer. Alternatively, the field vertices can also be customized by entering the coordinate values in the Field coordinates mask.

Function	Procedure
Scale field	Drag square red handle.
Rotate field	Drag round black handle.

Function	Procedure
Customize field shape	Drag square green handle.
Add field vertices	Double-click in required position.
Remove field vertex	Click to select the square green handle and press the [del] key.
Delete field	Press the [del] key without first selecting a green handle.

Important information about fields

It is possible to define up to 64 fields and 16 evaluations.

In the default settings, fields are projected into all levels. If you do not want this to happen, you can deactivate the option in Field properties.

12.4.5 Configuring interfaces

Settings for using the multifunctional I/Os can be made in the **Interfaces** area. In addition, the current status and saved function of the input/output are displayed.

The multifunctional I/Os are switchable and can therefore each be used as either a digital input or output. The factory settings are as follows: multifunctional I/Os 1-4 preset as inputs and multifunctional I/Os 5-8 preset as outputs.

The inputs may switch on, switch off, and switch over analysis cases, for example. The inputs can also be used to activate other functions, such as EasyTeach or measurement data output triggering.

The outputs can be used as digital switching outputs, for example to ground (PNP) (depending on the device type). For each output, it is necessary to define whether it is to be switched by the field evaluation application or by means of SOPAS ET telegrams, or whether it is being used to signal device readiness. If the field evaluation application is being used to switch an output, the sensor may signal evaluation field breaches or contour breaches. In SOPAS ET, you can configure which evaluation affects which output.

12.4.6 Some useful functions

button: Resets scan display to default view.

And button: Displays measured values as points or lines.

button: Freely rotates scan display.

12.4.7 Configuring the device

1. Set the desired configuration data in the **Basic settings** and **Interfaces** views (buttons under the menu bar).
2. Finally, permanently save the configuration in the **Finish** view:
   - Click the **Save permanent** button to save the parameter set in the device.
   - Click the **Export...** button to save the parameter set on the computer.

13 Telegram listing (EN)

Contents

13.1.1 About this document ... 71
13.1.1.1 Information on the telegram listing ... 71
13.1.1.2 Explanation of symbols ... 72
13.1.2 Communication format ... 72
13.1.2.1 Binary telegram (CoLa B) ... 72
13.1.2.2 ASCII telegram (CoLa A) ... 73
13.1.2.3 Variable types ... 73
13.1.2.4 Command basics ... 74
13.1.2.5 Log in: Required user level ... 75
13.1.3 Workflows ... 75
13.1.3.1 Parameterize the scan ... 75
13.1.3.2 Set timestamp/data angle ... 76
13.1.3.3 Common telegrams ... 76
13.1.4 Telegrams ... 77
13.1.4.1 Log in [sMN SetAccessMode] ... 78
13.1.4.2 Basic Settings ... 79
13.1.4.2.1 Set frequency and angular resolution/measurement sectors [sMN mLMPsetscancfg] ... 79
13.1.4.2.2 Read for frequency and angular resolution [sRN LMPscancfg]. 80
13.1.4.2.3 Alignment mode (one layer activation for adjustment) [sWN MMAlignmentMode] ... 81
13.1.4.2.4 Activate standby mode [sMN LMCstandby] ... 82
13.1.4.2.5 Start measurement [sMN LMCstartmeas] ... 83
13.1.4.2.6 Stop measurement [sMN LMCstopmeas] ... 84
13.1.4.2.7 Application selection and switching [sWN SetActiveApplications] ... 85
13.1.4.2.8 Read Application selection and switching [sRN SetActiveApplications] ... 86
13.1.4.2.9 Load factory defaults [sMN mSCloadfacdef] ... 87
13.1.4.2.10 Load application defaults [sMN mSCloadappdef] ... 87
13.1.4.2.11 Change password [sMN SetPassword] ... 88
13.1.4.2.12 Check password [sMN CheckPassword] ... 90
13.1.4.2.13 Set contamination indication settings [sWN ContaminationConfig] ... 91
13.1.4.2.14 Read contamination indication settings [sRN ContaminationConfig] ... 92
13.1.4.2.15 Read contamination indication data [sRN ContaminationData] 94
13.1.4.2.16 Read contamination indication result [sRN ContaminationResult] ... 94
13.1.4.2.17 Set device temperature alarm thresholds [sWN TemperatureAlarmConfig] ... 95
13.1.4.2.18 Read device temperature alarm thresholds [sRN TemperatureAlarmConfig] ... 96
13.1.4.2.19 Save parameters permanently [sMN mEEwriteall] ... 97

13.1.4.2.20 Set to run [sMN Run] ... 98
13.1.4.2.21 Reboot device [sMN mSCreboot] ... 99
13.1.4.3 Measurement output telegram ... 100
13.1.4.3.1 Configure the data content for the scan [sWN LMDscandatacfg] ... 100
13.1.4.3.2 Configure aperture angle of the scandata for output [sWN LMPoutputRange] ... 101
13.1.4.3.3 Read for actual output range [sRN LMPoutputRange] ... 102
13.1.4.3.4 Poll one telegram [sRN LMDscandata] ... 103
13.1.4.3.5 Send data permanently [sEN LMDscandata] ... 104
13.1.4.3.6 Read IMU data [sRN InertialMeasurementUnit] ... 113
13.1.4.4 Time stamp ... 115
13.1.4.4.1 Set time synchronization [sWN TSCRole] ... 115
13.1.4.4.2 Set time stamp [sMN LSPsetdatetime] ... 116
13.1.4.4.3 Set NTP (Network Time Protocol) parameters ... 118
13.1.4.5 Filters ... 122
13.1.4.5.1 Set particle filter [sWN LFPparticle] ... 122
13.1.4.5.2 Set average filter [sWN LFPmeanfilter] ... 123
13.1.4.5.3 Set echo filter [sWN FREchoFilter] ... 124
13.1.4.5.4 Set sensitivity fog filter [sWN MCSenseLevel] ... 124
13.1.4.5.5 Activate Median filter [sWN LFPmedianfilter] ... 125
13.1.4.6 Inputs and Outputs ... 126
13.1.4.6.1 Read Port Configration of all I/Os [sRN PortConfiguration] ... 126
13.1.4.6.2 Read state of the inputs [sRN LIDinputstate] ... 128
13.1.4.6.3 Read state of the outputs [sRN LIDoutputstate] ... 130
13.1.4.6.4 Receive outputstate by event [sEN LIDoutputstate] ... 130
13.1.4.6.5 Set output state [sMN mDOSetOutput] ... 132
13.1.4.6.6 Reset output counter [sMN LIDrstoutpcnt] ... 133
13.1.4.7 Status ... 134
13.1.4.7.1 Read firmware version [sRN DeviceIdent] ... 134
13.1.4.7.2 Read the device state [sRN SCdevicestate] ... 135
13.1.4.7.3 Read device order number [sRN DIornr] ... 136
13.1.4.7.4 Read device type [sRN DItype] ... 137
13.1.4.7.5 Read operating hours [sRN ODoprh] ... 137
13.1.4.7.6 Read power on counter [sRN ODpwrc] ... 138
13.1.4.7.7 Read temperature [sRN OPcurtmpdev] ... 139
13.1.4.7.8 Set device name [sWN LocationName] ... 140
13.1.4.7.9 Read device name [sRN LocationName] ... 141
13.1.4.7.10 FieldEvalLoggingList by Event [sEN FieldEvalLoggingList] ... 141
13.1.4.7.11 FieldEvalLoggingList on Request [sRN FieldEvalLoggingList] ... 144
13.1.4.8 Interfaces ... 145
13.1.4.8.1 Set IP address [sWN EIIpAddr] ... 145
13.1.4.8.2 Read IP address [sRN EIIpAddr] ... 146
13.1.4.8.3 Set Ethernet gateway [sWN EIgate] ... 147
13.1.4.8.4 Read Ethernet gateway [sRN EIgate] ... 148
13.1.4.8.5 Set IP mask [sWN EImask] ... 149

13.1.4.8.6 Read IP mask [sRN EImask] ... 150
13.1.4.8.7 Set Host/ UDP port number [sWN EIHstPort, sWN EIUDPPort]. 151
13.1.4.8.8 Set Host port Command Language (CoLa dialect) [sWN EIHst-Cola] ... 152
13.1.4.9 Application ... 154
13.1.4.9.1 Request minimal and maximal perpendicular distance once [sRN LFEperpdistresult] ... 154
13.1.4.9.2 Request minimal and maximal perpendicular distance continiously on event [sEN LFEperpdistresult] ... 156
13.1.4.9.3 Set activate evaluation [sMN ActivateEvaluation] ... 159
13.1.5 Diagnostics ... 160
13.1.5.1 SOPAS error codes ... 160

13.1.1 About this document

13.1.1.1 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

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):

- V2.3.0.0R

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.

13.1.1.2 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

13.1.2 Communication format

13.1.2.1 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 16: 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

13.1.2.2 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 17: 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)

13.1.2.3 Variable types

Variable type	Length (byte)	Value range	Sign
Bool_1	1	0 or 1	No

Variable type	Length (byte)	Value range	Sign
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
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	
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).

13.1.2.4 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	

Description	Value ASCII	Value Hex	Value Binary
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 146.

13.1.2.5 Log in: Required user level

Task	Required user level
Change sensor parameters	Authorized Client
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 78. 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 98.

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

13.1.3 Workflows

13.1.3.1 Parameterize the scan

Log in: `sMN SetAccessMode` see "Log in [sMN SetAccessMode] ", page 78

Set frequency and resolution: `sMN mLMPsetscancfg` see "Set frequency and angular resolution/measurement sectors [sMN mLMPsetscancfg]", page 79

Configure scandata content: `sWN LMDscandatacfg` see "Configure the data content for the scan [sWN LMDscandatacfg]", page 100

Configure scandata output: `sWN LMPoutputRange` see "Configure aperture angle of the scandata for output [sWN LMPoutputRange]", page 101

Store parameters: `sMN mEEwriteall` see "Save parameters permanently [sMN mEEwriteall]", page 97

Log out: `sMN Run` see "Set to run [sMN Run]", page 98

Request scan:

`sRN LMDscandata` see "Poll one telegram [sRN LMDscandata]", page 103

`sEN LMDscandata` see "Send data permanently [sEN LMDscandata]", page 104

(Device output ...)

13.1.3.2 Set timestamp/data angle

Log in: `sMN SetAccessMode` see "Log in [sMN SetAccessMode] ", page 78

SOPAS command: `sMN LSPsetdatetime` see "Set time stamp [sMN LSPsetdatetime]", page 116

Log out: `sMN Run` see "Set to run [sMN Run]", page 98

13.1.3.3 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 78
"Set frequency and angular resolution/measurement sectors [sMN mLMPsetscancfg]", page 79
"Start measurement [sMN LMCstartmeas]", page 83
"Stop measurement [sMN LMCstopmeas]", page 84
"Load factory defaults [sMN mSCloadfacdef]", page 87
"Load application defaults [sMN mSCloadappdef]", page 87
"Change password [sMN SetPassword]", page 88
"Check password [sMN CheckPassword]", page 90
"Reboot device [sMN mSCreboot]", page 99
"Save parameters permanently [sMN mEEwriteall]", page 97
"Set to run [sMN Run]", page 98
"Configure the data content for the scan [sWN LMDscandatacfg]", page 100
"Configure aperture angle of the scandata for output [sWN LMPoutputRange]", page 101
"Read for actual output range [sRN LMPoutputRange]", page 102
"Poll one telegram [sRN LMDscandata]", page 103
"Send data permanently [sEN LMDscandata]", page 104
"Set particle filter [sWN LFPparticle]", page 122
"Read state of the inputs [sRN LIDinputstate]", page 128
"Read state of the outputs [sRN LIDoutputstate]", page 130
"Receive outputstate by event [sEN LIDoutputstate]", page 130
"Set output state [sMN mDOSetOutput]", page 132
"Read firmware version [sRN DeviceIdent]", page 134
"Read the device state [sRN SCdevicestate]", page 135
"Read device order number [sRN DIornr]", page 136
"Read device type [sRN DItype]", page 137
"Read operating hours [sRN ODoprh]", page 137
"Read power on counter [sRN ODpwrc]", page 138
"Set device name [sWN LocationName]", page 140
"Read device name [sRN LocationName]", page 141
"Reset output counter [sMN LIDrstoutpcnt]", page 133
"Set IP address [sWN EIIpAddr]", page 145
"Read IP address [sRN EIIpAddr]", page 146
"Set Ethernet gateway [sWN EIgate]", page 147
"Read Ethernet gateway [sRN EIgate]", page 148

"Set IP mask [sWN EImask]", page 149
"Read IP mask [sRN EImask]", page 150
"Set Host/ UDP port number [sWN EIHstPort, sWN EIUDPPort]", page 151
"Set Host port Command Language (CoLa dialect) [sWN EIHstCola]", page 152

13.1.4 Telegrams

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

Table 18: 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 19: 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 20: 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 21: 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"

13.1.4.1 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 88).

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 97) must be send before log out.

Table 22: 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		SetAccessMode	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 23: 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 24: 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		SetAccessMode	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 25: 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

13.1.4.2 Basic Settings

13.1.4.2.1 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 98).

Coordination system:

(-47,5° to 227,5°) front = 90° (0° in SOPAS ET software)

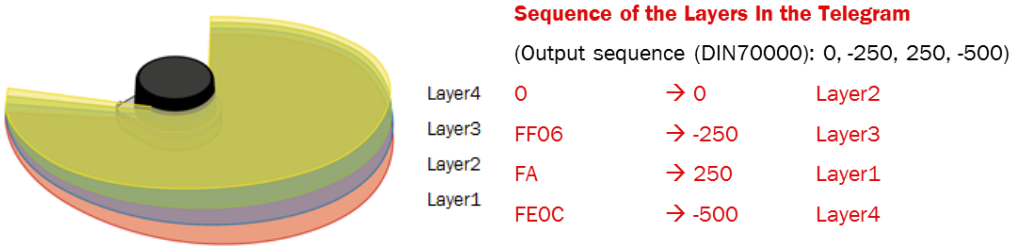


Table 26: Telegram structure: sMN mLMPsetscancfg

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	12,5 Hz: 25 Hz: 50 Hz:	+1250d (4E2h) +2500d (9C4h) +5000d (1388h)	00 00 04 E2 00 00 09 C4 00 00 13 88
Reserved	-	Int_16	2	Always:	+1 (0001h)	00 01
Angular resolution	[1/10000°]	Uint_32	4	0.0625°: 0.125°: 0.25°:	+0625d (271h) +1250d (4E2h) +2500d (9C4h)	00 00 02 71 00 00 04 E2 00 00 09 C4
Start angle	[1/10000°]	Int_32	4	-47.5°	-475000d (FFF8C088h)	FF F8 C0 88
Stop angle	[1/10000°]	Int_32	4	+227,5°	+2275000d (22B6B8h)	00 22 B6 B8

Table 27: 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	50 Hz: 25 Hz: 50 Hz:	4E2h 9C4h 1388h	00 00 04 E2 00 00 09 C4 00 00 13 88
Reserved	-	Int_16	2	Always:	1h	00 01
Angular resolution	[1/10000°]	Uint_32	4	0.0625°: 0.125°: 0.25°:	271h 4E2h 9C4h	00 00 02 71 00 00 04 E2 00 00 09 C4
Start angle	[1/10000°]	Int_32	4	-47.5°	FFF8C088h	FF F8 C0 88
Stop angle	[1/10000°]	Int_32	4	+227,5°	22B6B8h	00 22 B6 B8

Table 28: 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

13.1.4.2.2 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 29: 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 30: 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 31: 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	12.5 Hz: 25 Hz: 50 Hz:	9C4h 1388h	00 00 04 E2 00 00 09 C4 00 00 13 88
Reserved	-	Int_16	2	Always:	1	00 01
Angular resolution	[1/10000°]	Uint_32	4	0.0625°: 0.125°: 0.25°:	271h 4E2h 9C4h	00 00 02 71 00 00 04 E2 00 00 09 C4
Start angle	[1/10000°]	Int_32	4	-47.5°	FFF8C088h	FF F8 C0 88
Stop angle	[1/10000°]	Int_32	4	+227.5°	22B6B8h	00 22 B6 B8

Table 32: 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

13.1.4.2.3 Alignment mode (one layer activation for adjustment) [sWN MMAlignmentMode]

Table 33: Telegram structure: sWN MMAlignmentMode

Telegram structure: sWN MMAlignmentMode (Service) (sMN SetAccessMode 04 81BE23AA)						
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 to alignment mode	String	15		MMAlignment-Mode	4D 4D 41 6C 69 67 6E 6D 65 6E 74 4D 6F 64 65

Telegram structure: sWN MMAlignmentMode (Service) (sMN SetAccessMode 04 81BE23AA)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Layer activation				All layer: Red layer -2.5°: Blue layer 0°: Green layer +2.5°: Yellow layer +5°:	0 1 2 3 4	00 01 02 03 04

Table 34: Example: sWN MMAlignmentMode

CoLa A	<STX>sWN{SPC}MMAlignmentMode{SPC}2<ETX>	
	<STX>sWN MMAlignmentMode 2<ETX>	
	sWN MMAlignmentMode 2	
	02 73 57 4E 20 4D 4D 41 6C 69 67 6E 6D 65 6E 74 4D 6F 64 65 20 32 03	
CoLa B	02 02 02 02 00 00 00 15 73 57 4E 20 4D 4D 41 6C 69 67 6E 6D 65 6E 74 4D 6F 64 65 20 02 14	
	73 57 4E 20 4D 4D 41 6C 69 67 6E 6D 65 6E 74 4D 6F 64 65 20 02	

Table 35: Telegram structure: sWA MMAlignmentMode

Telegram structure: sWA MMAlignmentMode						
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 to alignment mode	String	15		MMAlignment-Mode	4D 4D 41 6C 69 67 6E 6D 65 6E 74 4D 6F 64 65

Table 36: Example: sWA MMAlignmentMode

CoLa A	<STX>sWA{SPC}MMAlignmentMode<ETX>
	02 73 57 41 20 4D 4D 41 6C 69 67 6E 6D 65 6E 74 4D 6F 64 65 03
CoLa B	02 02 02 02 00 00 00 14 73 57 41 20 4D 4D 41 6C 69 67 6E 6D 65 6E 74 4D 6F 64 65 20 19

13.1.4.2.4 Activate standby mode [sMN LMCstandby]

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

Table 37: 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 38: 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 39: 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:	0	00

Table 40: 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

13.1.4.2.5 Start measurement [sMN LMCstartmeas]

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

Table 41: 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 42: 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 43: 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 44: 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

13.1.4.2.6 Stop measurement [sMN LMCstopmeas]

Shut off the laser and stop the motor.

Table 45: 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 46: 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 47: 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

Telegram structure: sAN LMCstopmeas						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Status code	Accepted when value is 0	Enum_8	1	No error: Not allowed:	0 1	00 01

Table 48: 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

13.1.4.2.7 Application selection and switching [sWN SetActiveApplications]

Selection between the field application and the ranging application of the device.

(Since FW V2.x.x both applications are parallely selected/active as default).

Table 49: Telegram structure: sWN SetActiveApplications

Telegram structure: sWN SetActiveApplications (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	Selects all currently active applications of the scanner	String	13		SetActiveAppli-cations	53 65 74 41 63 74 69 76 65 41 70 70 6C 69 63 61 74 69 6F 6E 73
Array lenght					0 ... 2	00 ... 02
Identifier	Application	String		Field Application: Ranging:	FEVL RANG	46 45 56 4C 52 41 4E 47
Active		Bool_1		False: True:	0 1	00 01

Table 50: Example: Activate the field application: : sWN SetActiveApplications 1 FEVL 1

CoLa A	<STX>sWN{SPC}SetActiveApplications{SPC}1{SPC}FEVL{SPC}1<ETX>	
	<STX>sWN SetActiveApplications 1 FEVL 1<ETX>	
	sWN SetActiveApplications 1 FEVL 1	
	02 73 57 4E 20 53 65 74 41 63 74 69 76 65 41 70 70 6C 69 63 61 74 69 6F 6E 73 20 31 20 46 45 56 4C 20 31 03	
CoLa B	02 02 02 02 00 00 00 22 73 57 4E 20 53 65 74 41 63 74 69 76 65 41 70 70 6C 69 63 61 74 69 6F 6E 73 20 31 20 46 45 56 4C 20 01 34	
	73 57 4E 20 53 65 74 41 63 74 69 76 65 41 70 70 6C 69 63 61 74 69 6F 6E 73 20 31 20 46 45 56 4C 20 31	

Table 51: Telegram structure: sWA SetActiveApplications

Telegram structure: sWA SetActiveApplications						
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	Selects all currently active applications of the scanner	String	13		SetActiveAppli-cations	53 65 74 41 63 74 69 76 65 41 70 70 6C 69 63 61 74 69 6F 6E 73

Table 52: Example: sWA SetActiveApplications correct and accepted

CoLa A	<STX>sWA{SPC}SetActiveApplications<ETX>
	02 73 57 41 20 53 65 74 41 63 74 69 76 65 41 70 70 6C 69 63 61 74 69 6F 6E 73 03
CoLa B	02 02 02 02 00 00 00 19 73 57 41 20 53 65 74 41 63 74 69 76 65 41 70 70 6C 69 63 61 74 69 6F 6E 73 02

13.1.4.2.8 Read Application selection and switching [sRN SetActiveApplications]

Table 53: Telegram structure: sRN SetActiveApplications

Telegram structure: sRN SetActiveApplications						
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		SetActiveAppli-cations	53 65 74 41 63 74 69 76 65 41 70 70 6C 69 63 61 74 69 6F 6E 73

Table 54: Example for MRS1000: sRN SetActiveApplications

CoLa A	<STX>sRN{SPC}SetActiveApplications<ETX>	
	<STX>sRN SetActiveApplications<ETX>	
	sRN SetActiveApplications	
	02 73 52 4E 20 53 65 74 41 63 74 69 76 65 41 70 70 6C 69 63 61 74 69 6F 6E 73 03	
CoLa B	02 02 02 02 00 00 00 19 73 52 4E 20 53 65 74 41 63 74 69 76 65 41 70 70 6C 69 63 61 74 69 6F 6E 73 08	
	73 52 4E 20 53 65 74 41 63 74 69 76 65 41 70 70 6C 69 63 61 74 69 6F 6E 73	

Telegram structure: sRA SetActiveApplications						
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		SetActiveAppli-cations	53 65 74 41 63 74 69 76 65 41 70 70 6C 69 63 61 74 69 6F 6E 73

13.1.4.2.9 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.

Table 55: 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
Command	Load factory defaults	String	13		mSCloadfacdef	6D 53 43 6C 6F 61 64 66 61 63 64 65 66

Table 56: 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 57: 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 58: 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

13.1.4.2.10 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 59: 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 60: 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 61: 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 62: 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

13.1.4.2.11 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 63: 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

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	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>	

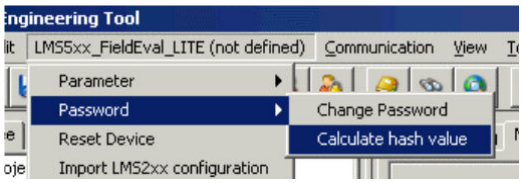
Set password for 'Authorized client' to “testtest” (hash value = 1920E4C9).

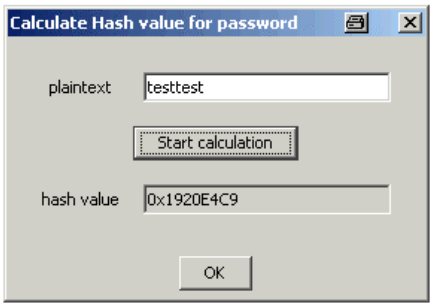
Table 64: Example: sMN SetPassword

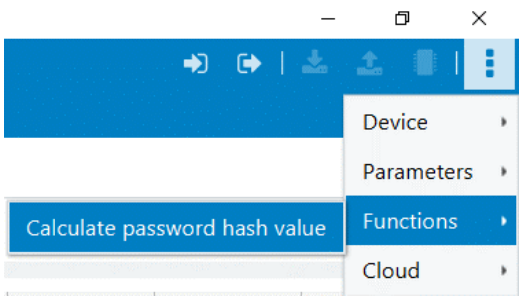
CoLa A	<STX>sMN{SPC}SetPassword{SPC}3{SPC}1920E4C9<ETX>
	<STX>sMN SetPassword 3 1920E4C9<ETX>
	sMN SetPassword 3 1920E4C9
	02 73 4D 4E 20 53 65 74 50 61 73 73 77 6F 72 64 20 33 20 31 39 32 30 45 34 43 39 03
CoLa B	02 02 02 02 00 00 00 15 73 4D 4E 20 53 65 74 50 61 73 73 77 6F 72 64 20 03 19 20 E4 C9 1A
	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.**
- ► Alternatively select > **Functions** > **Calculate password hash value**







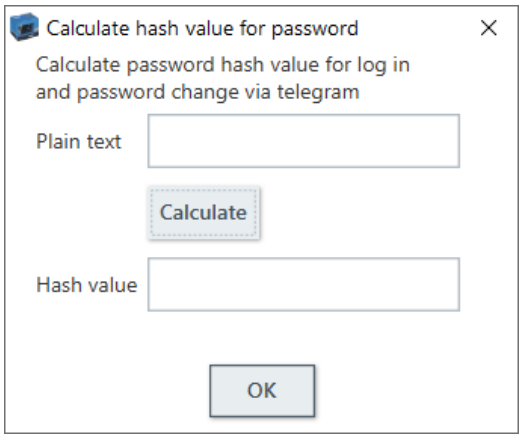


Table 65: Telegram structure: sAN SetPassword

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 66: 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

13.1.4.2.12 Check password [sMN CheckPassword]

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

Table 67: 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 68: 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 69: 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 70: 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

13.1.4.2.13 Set contamination indication settings [sWN ContaminationConfig]

Define the exact way how and when the device shall signal a potential contamination of the optics cover so that it may be cleaned preventively (see "Contamination indication", page 30).

Table 71: Telegram structure: sWN ContaminationConfig

Telegram structure: sWN ContaminationConfig (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	19		Contamination-Config	43 6F 6E 74 61 6D 69 6E 61 74 69 6F 6E 43 6F 6E 66 69 67
Strategy	Strategy code	Enum_8	1	Inactive: High available: Available: Semi-sensitive: Sensitive:	0 1 2 3 4	00 01 02 03 04
Response time	Error / warning reaction time in sec	Uint_16	2		+1d ... +60d (01h ... 3Ch)	00 01 ... 00 3C
Threshold warning	Sensitivity	Enum_8	1	HIGH: MID: LOW:	0 1 2	00 01 02
Cover	Weather protection hood constalation	Enum_8	1	No weather protection hood: Weather protection hood 210 no Flaps: Weather protection hood 210 Flap left: Weather protection hood 210 Flap right: Weather protection hood 210 both Flaps: Weather protection hood 275:	0 1 2 3 4 5	00 01 02 03 04 05

Telegram structure: sWN ContaminationConfig (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Enable Warning	Warning monitoring	Bool_1	1	Off (False): On (True):	0 1	00 01
Enable Error	Error monitoring	Bool_1	1	Off (False): On (True):	0 1	00 01

Table 72: Example: sWN ContaminationConfig strategy high available, response time 3 sec, sensitivity low, no weather protection hood, enable warning, enable error

CoLa A	<STX>sWN{SPC}ContaminationConfig{SPC}1{SPC}3{SPC}2{SPC}0{SPC}1{SPC}1<ETX>	
	<STX>sWN ContaminationConfig 1 3 2 0 1 1<ETX>	
	sWN ContaminationConfig 1 3 2 0 1 1	
	02 73 57 4E 20 43 6F 6E 74 61 6D 69 6E 61 74 69 6F 6E 43 6F 6E 66 69 67 20 31 20 33 20 32 20 30 20 31 20 31 03	
CoLa B	02 02 02 02 00 00 00 1F 73 57 4E 20 43 6F 6E 74 61 6D 69 6E 61 74 69 6F 6E 43 6F 6E 66 69 67 20 01 00 03 02 00 01 01 00	
	73 57 4E 20 43 6F 6E 74 61 6D 69 6E 61 74 69 6F 6E 43 6F 6E 66 69 67 20 01 00 03 02 00 01 01	

Table 73: Telegram structure: sWA ContaminationConfig

Telegram structure: sWA ContaminationConfig						
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	19		Contamination-Config	43 6F 6E 74 61 6D 69 6E 61 74 69 6F 6E 43 6F 6E 66 69 67

Table 74: Example: sWA ContaminationConfig

CoLa A	<STX>sWA{SPC}ContaminationConfig<ETX>
	02 73 57 41 20 43 6F 6E 74 61 6D 69 6E 61 74 69 6F 6E 43 6F 6E 66 69 67 03
CoLa B	02 02 02 02 00 00 00 18 73 57 41 20 43 6F 6E 74 61 6D 69 6E 61 74 69 6F 6E 43 6F 6E 66 69 67 20

13.1.4.2.14 Read contamination indication settings [sRN ContaminationConfig]

Read which contamination measurement strategy is currently active.

Table 75: Telegram structure: sRN ContaminationConfig

Telegram structure: sRN ContaminationConfig						
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	19		Contamination-Config	43 6F 6E 74 61 6D 69 6E 61 74 69 6F 6E 43 6F 6E 66 69 67

Table 76: Example: sRN ContaminationConfig

CoLa A	<STX>sRN{SPC}ContaminationConfig<ETX>
	<STX>sRN ContaminationConfig<ETX>
	sRN ContaminationConfig
	02 73 52 4E 20 43 6F 6E 74 61 6D 69 6E 61 74 69 6F 6E 43 6F 6E 66 69 67 03
CoLa B	02 02 02 02 00 00 00 17 73 52 4E 20 43 6F 6E 74 61 6D 69 6E 61 74 69 6F 6E 43 6F 6E 66 69 67 25
	73 52 4E 20 43 6F 6E 74 61 6D 69 6E 61 74 69 6F 6E 43 6F 6E 66 69 67 25

Table 77: Telegram structure: sRA ContaminationConfig

Telegram structure: sRA ContaminationConfig						
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	19		Contamination-Config	43 6F 6E 74 61 6D 69 6E 61 74 69 6F 6E 43 6F 6E 66 69 67
Strategy	Strategy code	Enum_8	1	Inactive: High available: Available: Semi-sensitive: Sensitive:	0 1 2 3 4	00 01 02 03 04
Response time	Error / warning reaction time in sec	Uint_16	2		01h ... 3Ch	00 01 ... 00 3C
Threshold warning	Sensitivity	Enum_8	1	HIGH: MID: LOW:	0 1 2	00 01 02
Cover	Weather protection hood constalation	Enum_8	1	No weather protection hood: Weather protection hood 210 no Flaps: Weather protection hood 210 Flap left: Weather protection hood 210 Flap right: Weather protection hood 210 both Flaps: Weather protection hood 275:	0 1 2 3 4 5	00 01 02 03 04 05
Enable Warning	Warning monitoring	Bool_1	1	Off (False): On (True):	0 1	00 01
Enable Error	Error monitoring	Bool_1	1	Off (False): On (True):	0 1	00 01

Table 78: Example: sRA ContaminationConfig strategy high available, response time 3 sec, sensitivity low, no weather protection hood, enable warning, enable error

CoLa A	<STX>sRA{SPC}ContaminationConfig{SPC}1{SPC}3{SPC}2{SPC}0{SPC}1{SPC}1<ETX>
	02 73 52 4E 20 43 6F 6E 74 61 6D 69 6E 61 74 69 6F 6E 43 6F 6E 66 69 67 03
CoLa B	02 02 02 02 00 00 00 1F 73 52 41 20 43 6F 6E 74 61 6D 69 6E 61 74 69 6F 6E 43 6F 6E 66 69 67 20 01 00 03 02 00 01 01 0A

13.1.4.2.15 Read contamination indication data [sRN ContaminationData]

Get the current contamination indication data.

Table 79: Telegram structure: sRN ContaminationData

Telegram structure: sRN ContaminationData						
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	17		Contamination-Data	43 6F 6E 74 61 6D 69 6E 61 74 69 6F 6E 44 61 74 61

Table 80: Example: sRN ContaminationData

CoLa A	<STX>sRN{SPC}ContaminationData<ETX>	
	<STX>sRN ContaminationData<ETX>	
	sRN ContaminationData	
	02 73 52 4E 20 43 6F 6E 74 61 6D 69 6E 61 74 69 6F 6E 44 61 74 61 03	
CoLa B	02 02 02 02 00 00 00 15 73 52 4E 20 43 6F 6E 74 61 6D 69 6E 61 74 69 6F 6E 44 61 74 61 3F	
	73 52 4E 20 43 6F 6E 74 61 6D 69 6E 61 74 69 6F 6E 44 61 74 61	

Table 81: Telegram structure: sRA ContaminationData

Telegram structure: sRA ContaminationData						
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	17		Contamination-Data	43 6F 6E 74 61 6D 69 6E 61 74 69 6F 6E 44 61 74 61
Contamination data for different channels	Status of contamination in order of the different channels. One entry for each channel.	Array	6	Order of 6 channels: (1 / 2 / 3 / 4 / 5 / 6) CM channel: CM NONE: CM WARN: CM ERROR:	 x 0 1 2	 x 00 01 02

Table 82: Example: sRA ContaminationData

CoLa A	<STX>sRA{SPC}ContaminationData{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0<ETX>
	02 73 52 41 20 43 6F 6E 74 61 6D 69 6E 61 74 69 6F 6E 44 61 74 61 20 30 20 30 20 30 20 30 20 30 20 30 03
CoLa B	02 02 02 02 00 00 00 1C 73 52 41 20 43 6F 6E 74 61 6D 69 6E 61 74 69 6F 6E 44 61 74 20 00 00 00 00 00 00 10

13.1.4.2.16 Read contamination indication result [sRN ContaminationResult]

This telegram is intended to read the contamination result. The telegram returns two Boolean. One for warning and one for error. Depending on the strategy of the contamination indication the state changes if one or all sectors are in the specific status (warning or error).

Table 83: Telegram structure: sRN ContaminationResult

Telegram structure: sRN ContaminationResult						
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	19		Contamination-Result	43 6F 6E 74 61 6D 69 6E 61 74 69 6F 6E 52 65 73 75 6C 74

Table 84: Example: sRN ContaminationResult

CoLa A	<STX>sRN{SPC}ContaminationResult<ETX>	
	<STX>sRN ContaminationResult<ETX>	
	sRN ContaminationResult	
	02 73 52 4E 20 43 6F 6E 74 61 6D 69 6E 61 74 69 6F 6E 52 65 73 75 6C 74 03	
CoLa B	02 02 02 02 00 00 00 17 73 52 4E 20 43 6F 6E 74 61 6D 69 6E 61 74 69 6F 6E 52 65 73 75 6C 74 26	
	73 52 4E 20 43 6F 6E 74 61 6D 69 6E 61 74 69 6F 6E 52 65 73 75 6C 74	

Table 85: Telegram structure: sRA ContaminationResult

Telegram structure: sRA ContaminationResult						
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	19		Contamination-Result	43 6F 6E 74 61 6D 69 6E 61 74 69 6F 6E 52 65 73 75 6C 74
Contamination Result	Result of contamination indication (Order of results: Warning / Error)	Enum_8	1	Warning Inactive: Active:	 0 1	 00 01
				Error Inactive: Active:	 0 1	 00 01

Table 86: Example: sRA ContaminationResult warning inactive error inactive

CoLa A	<STX>sRA{SPC}ContaminationResult{SPC}0{SPC}0<ETX>
	02 73 52 41 20 43 6F 6E 74 61 6D 69 6E 61 74 69 6F 6E 52 65 73 75 6C 74 20 30 20 30 03
CoLa B	02 02 02 02 00 00 00 1A 73 52 41 20 43 6F 6E 74 61 6D 69 6E 61 74 69 6F 6E 52 65 73 75 6C 74 20 00 00 09

13.1.4.2.17 Set device temperature alarm thresholds [sWN TemperatureAlarmConfig]

This telegram is intended to set alarm thresholds of the device temperatures. The threshold indicates when an alarm state will be set.

Table 87: Telegram structure: sWN TemperatureAlarmConfig

Telegram structure: sWN TemperatureAlarmConfig						
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	22		TemperatureA-larmConfig	54 65 6D 70 65 72 61 74 75 72 65 41 6C 61 72 6D 43 6F 6E 66 69 67
Upper threshold	Set upper threshold value	Int_8	1		-128d ... +127d (80h ... 7Fh)	80 ... 7F
Lower threshold	Set lower threshold value	Int_8	1		-128d ... +127d (80h ... 7Fh)	80 ... 7F

Table 88: Example: sWN TemperatureAlarmConfig - 50 °C (upper threshold) and 0 °C (lower threshold)

CoLa A	<STX>sWN{SPC}TemperatureAlarmConfig{SPC}32{SPC}0<ETX>	
	<STX>sWN TemperatureAlarmConfig 32 0<ETX>	
	sWN TemperatureAlarmConfig 32 0	
	02 73 57 4E 20 54 65 6D 70 65 72 61 74 75 72 65 41 6C 61 72 6D 43 6F 6E 66 69 67 20 33 32 20 30 03	
CoLa B	02 02 02 02 00 00 00 1D 73 57 4E 20 54 65 6D 70 65 72 61 74 75 72 65 41 6C 61 72 6D 43 6F 6E 66 69 67 20 32 00 6D	
	73 57 4E 20 54 65 6D 70 65 72 61 74 75 72 65 41 6C 61 72 6D 43 6F 6E 66 69 67 20 32 00	

Table 89: Telegram structure: sWA TemperatureAlarmConfig

Telegram structure: sWA TemperatureAlarmConfig						
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	22		TemperatureA-larmConfig	54 65 6D 70 65 72 61 74 75 72 65 41 6C 61 72 6D 43 6F 6E 66 69 67

Table 90: Example: sWA TemperatureAlarmConfig

CoLa A	<STX>sWA{SPC}TemperatureAlarmConfig<ETX>
	02 73 57 41 20 54 65 6D 70 65 72 61 74 75 72 65 41 6C 61 72 6D 43 6F 6E 66 69 67 03
CoLa B	02 02 02 02 00 00 00 1B 73 57 41 20 54 65 6D 70 65 72 61 74 75 72 65 41 6C 61 72 6D 43 6F 6E 66 69 67 20 50

13.1.4.2.18 Read device temperature alarm thresholds [sRN TemperatureAlarmConfig]

This telegram is intended to read the alarm thresholds of the device temperatures. The threshold indicates when an alarm state will be set.

Table 91: Telegram structure: sRN TemperatureAlarmConfig

Telegram structure: sRN TemperatureAlarmConfig						
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	22		TemperatureA-larmConfig	54 65 6D 70 65 72 61 74 75 72 65 41 6C 61 72 6D 43 6F 6E 66 69 67

Table 92: Example: sRN TemperatureAlarmConfig

CoLa A	<STX>sRN{SPC}TemperatureAlarmConfig<ETX>	
	<STX>sRN TemperatureAlarmConfig<ETX>	
	sRN TemperatureAlarmConfig	
	02 73 52 4E 20 54 65 6D 70 65 72 61 74 75 72 65 41 6C 61 72 6D 43 6F 6E 66 69 67 03	
CoLa B	02 02 02 02 00 00 00 17 73 52 4E 20 54 65 6D 70 65 72 61 74 75 72 65 41 6C 61 72 6D 43 6F 6E 66 69 67 7A	
	73 52 4E 20 54 65 6D 70 65 72 61 74 75 72 65 41 6C 61 72 6D 43 6F 6E 66 69 67	

Table 93: Telegram structure: sRA TemperatureAlarmConfig

Telegram structure: sRA TemperatureAlarmConfig						
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	22		TemperatureA-larmConfig	54 65 6D 70 65 72 61 74 75 72 65 41 6C 61 72 6D 43 6F 6E 66 69 67
Upper threshold	Upper threshold value	Int_8	1		-128d ... +127d (80h ... 7Fh)	80 ... 7F
Lower threshold	Lower threshold value	Int_8	1		-128d ... +127d (80h ... 7Fh)	80 ... 7F

Table 94: Example: sRA TemperatureAlarmConfig - 50 °C (upper threshold) and 0 °C (lower threshold)

CoLa A	<STX>sRA{SPC}TemperatureAlarmConfig{SPC}32{SPC}0<ETX>
	02 73 52 41 20 54 65 6D 70 65 72 61 74 75 72 65 41 6C 61 72 6D 43 6F 6E 66 69 67 20 33 32 20 30 03
CoLa B	02 02 02 02 00 00 00 1B 73 52 41 20 54 65 6D 70 65 72 61 74 75 72 65 41 6C 61 72 6D 43 6F 6E 66 69 67 20 32 00 67

13.1.4.2.19 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 95: 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 96: 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 97: 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 98: 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

13.1.4.2.20 Set to run [sMN Run]

Log out from device and activate all parameter changes.

Table 99: 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 100: 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 101: 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 102: 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

13.1.4.2.21 Reboot device [sMN mSCreboot]

This command includes saving all parameters.

Table 103: 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 104: 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 105: 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 106: 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

13.1.4.3 Measurement output telegram

13.1.4.3.1 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).

Table 107: 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	Set via Echo Filter, therefore: Other values than 0 will be ignored.	0 0	00 00
Further data channels	Remission, Angle offset, Ambient light & Quality data output RSSI, Angle offset, Ambient light & Quality data output	Uint_8	1	No values: RSSI: AINF: RSSI & AINF:	0 1 8 9	00 01 08 09
Resolution	Resolution of RSSI	Enum_8	1	8 Bit:	0	00
Unit	Unit of RSSI	Enum_8	1	Digits:	0	00
Reserved	-	Uint_8	2	Always:	0 0	00 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
Comment	Saved comment	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)**
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:	+1d (1h)	00 01

Example 1: RSSI, resolution 8Bit and output rate all scans

Table 108: Example 1: sWN LMDscandatacfg

CoLa A	<STX>sWN{SPC}LMDscandatacfg{SPC}01{SPC}00{SPC}1{SPC}0{SPC}0{SPC}00{SPC}00{SPC}0{SPC}0{SPC}0{SPC}0{SPC}+1<ETX>	
	<STX>sWN LMDscandatacfg 01 00 1 0 0 00 00 0 0 0 0 +1<ETX>	
	sWN LMDscandatacfg 01 00 1 0 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 30 20 30 20 30 30 20 30 30 20 30 20 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 00 00 00 00 00 00 00 00 00 01 43	
	73 57 4E 20 4C 4D 44 73 63 61 6E 64 61 74 61 63 66 67 20 01 00 01 00 00 00 00 00 00 00 00 00 01	

Table 109: 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		LMDscandatacfg	4C 4D 44 73 63 61 6E 64 61 74 61 63 66 67

Table 110: 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

13.1.4.3.2 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 111: 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 resolution [1]	[1/10000°] Fixed value, not changeable	Uint_32	4	0.25°:	+2500d (9C4h)	00 00 09 C4
Start angle	[1/10000°]	Int_32	4	-47.5°:	-475000d (FFF8C088h)	FF F8 C0 88
Stop angle	[1/10000°]	Int_32	4	+227.5°	+2275000d (22B6B8h)	00 22 B6 B8

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

Table 112: Example: sWN LMPoutputRange - set output data for angular resolution at 0.50° and range from 0° to 90°

CoLa A	<STX>sWN{SPC}LMPoutputRange{SPC}1{SPC}1388{SPC}0{SPC}DBBA0<ETX>	
	<STX>sWN LMPoutputRange 1 1388 0 DBBA0<ETX>	
	sWN LMPoutputRange 1 1388 0 DBBA0	
	02 73 57 4E 20 4C 4D 50 6F 75 74 70 75 74 52 61 6E 67 65 20 31 20 31 33 38 38 20 30 20 44 42 42 41 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 00 00 00 00 00 0D BB A0 F7	
	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 00 00 00 00 00 0D BB A0	

Table 113: 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 114: 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

13.1.4.3.3 Read for actual output range [sRN LMPoutputRange]

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

Table 115: 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 116: 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 117: 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°:	9C4h	00 00 09 C4
Start angle	[1/10000°]	Int_32	4	-47.5°	FFF8C088h	FF F8 C0 88
Stop angle	[1/10000°]	Int_32	4	+227.5°	22B6B8h	00 22 B6 B8

Table 118: 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

13.1.4.3.4 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).

Table 119: 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 120: 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 121: 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 104.						

13.1.4.3.5 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 100 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).

Table 122: 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 123: 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 124: 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 125: 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 126: 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)

Telegram structure: sRA LMDscandata / sSN LMDscandata						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Status info						
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		0 0	00 00
Status of digital outputs	Low byte represents output 1.	Uint_8	2 x 1	All outputs low: Always:	0 0 0	00 00 00 00
Layer angle	-		2	Angle = value/100 0/100 = 0 → Layer 2 -250/100 = -2.5 → Layer 3 250/100 = 2.5 → Layer 1 -500/100 = -5 → Layer 4	 0 FF06h FAh FE0Ch	 00 00 00 00 46 46 30 36 00 00 46 41 46 45 30 43
Frequencies						
Scan frequency	[1/100 Hz]	Uint_32	4	50 Hz:	1388h	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
Reserved						
Reserved	-	Enum_16	2	Always:	0	00 00
16 bit output channel (Distance and RSSI)						

Telegram structure: sRA LMDscandata / sSN LMDscandata						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Amount of 16 bit channels	Number of 16 bit channels that provide measured data	Enum_16	2	Output channels	1 or 3	01 or 03
Content	Defines the content of the output channel Radial distance values (DIST) in mm	String	5	Distance values: Distance values: Distance values: Energy values: Energy values: Energy values:	DIST1 DIST2 DIST3 RSSI1 RSSI2 RSSI3	44 49 53 54 31 44 49 53 54 32 44 49 53 54 33 52 53 53 49 31 52 53 53 49 32 52 53 53 49 33
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
Start angle	[1/10000°]	Int_32	4	-47.5° ... 227.5°	FFF8C088h ... 22B6B8h	FF F8 C0 88 ... 00 22 B6 B8
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 101)	9C4h	09 C4
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_16	2		0h ... FA00h	00 00 00 00 .. 00 00 FA 00
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 3	01 or 03
Content	Defines the content of the output channel	String	5		DIST1 DIST2 DIST3 RSSI1 RSSI2 RSSI3 AINF1 (Ambient light)	44 49 53 54 31 44 49 53 54 32 44 49 53 54 33 52 53 53 49 31 52 53 53 49 32 52 53 53 49 33 41 49 4E 46 31
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	-47.5° ... 227.5°	FFF8C088h ... 22B6B8h	FF F8 C0 88 ... 00 22 B6 B8

Telegram structure: sRA LMDscandata / sSN LMDscandata						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
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 101)	9C4h	09 C4
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	 DIST & RSSI: AINF (Intensity): 0: 1: 2: 3: 4: 5: 6: 7:	00h ... FFh 0h ... FFh 00 04 08 0C 10 14 18 1C	00 ... FF 00 ... FF 00 04 08 0C 10 14 18 1C
Position						
Position	Output of position data	Enum_16	2	No position data: Position data:	0 1	00 00 00 01

Telegram structure: sRA LMDscandata / sSN LMDscandata						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
X position		Real as float according to IEEE754	4	Only filled if parameter is activated	-	-
Y position		Real as float according to IEEE754	4		-	-
Z position		Real as float according to IEEE754	4		-	-
X rotation		Real as float according to IEEE754	4		-	-
Y rotation		Real as float according to IEEE754	4		-	-
Z rotation		Real as float according to IEEE754	4		-	-
Rotations type		Enum_8	1		-	-
Name						
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
Comment						
Comment	Comment	Enum_16	2	No comment: Comment:	0 1	00 00 00 01
Length	Length of comment	Uint_8	1	Only filled if parameter is activated	0h ... Fh	00 ... 0F
Comment	Transmits a comment in characters	String	16		20h ... 7Ah	20 ... 7A
Reserved						
Reserved	-	Enum_16	2	Always:	0	00 00
Event info						
Event info	Display event info	Uint_16	2	No info: Transmit info:	0 1	00 00 00 01

Telegram structure: sRA LMDscandata / sSN LMDscandata						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Type	Fast digital input	String	4	Only filled if parameter is activated	FDIN	FD IN
Encoder position	Position of encoder when event happened	Uint_32	4		00000000h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
Time of event	Time (µs) of encoder when event happened	Uint_32	4		00000000h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
Angle of event	Angle of encoder when event happened	Int_32	4		0 ... 3600000	00 00 00 00 ... 00 36 EE 80

NOTE

- The grey written parts are not given out by the sensor.
- The order of events within the data structure is “newest” first.

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) or GRE (ground reference evaluation)
4d – 15d	0d	Reserved, currently not in use
≥16d	>0h	Valid measurement values

Example of a telegram stream

Table 127: 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

Telegram structure: sRA LMDscandata (Example)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Device status	(See values column)	Uint_8	2 x 1	Ok:	0 0	00 00
Status info						
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
Layer angle	-	Uint_16	2		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

Telegram structure: sRA LMDscandata (Example)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Reserved						
Reserved	-	Enum_16	2	Always:	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
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
Reserved	-	Real as float according to IEEE754	4	Always:	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	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
Reserved						
Reserved	-	Uint_16	2	Always:	0	00 00

Telegram structure: sRA LMDscandata (Example)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Event info						
Event info	Display event info	Uint_16	2	No info:	0	00 00

Table 128: 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

13.1.4.3.6 Read IMU data [sRN InertialMeasurementUnit]

Request the most recently measured IMU data.

To get continous IMU data register by `sEN EnableInertialMeasurementUnit 1` and to stop getting continous IMU data use `sEN EnableInertialMeasurementUnit 0`.

Table 129: Telegram structure: sRN InertialMeasurementUnit

Telegram structure: sRN InertialMeasurementUnit (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		sRN	73 52 4E
Command	Read IMU data	String	23		InertialMeasur-ementUnit	49 6E 65 72 74 69 61 6C 4D 65 61 73 75 72 65 6D 65 6E 74 55 6E 69 74

Table 130: Example: sRN InertialMeasurementUnit

CoLa A	<STX>sRN{SPC}InertialMeasurementUnit<ETX>	
	<STX>sRN InertialMeasurementUnit<ETX>	
	sRN InertialMeasurementUnit	
	02 73 52 E4 20 49 6E 65 72 74 69 61 6C 4D 65 61 73 75 72 65 6D 65 6E 74 55 6E 69 74 03	
CoLa B	02 02 02 02 00 00 00 1B 73 52 E4 20 49 6E 65 72 74 69 61 6C 4D 65 61 73 75 72 65 6D 65 6E 74 55 6E 69 74 03	
	73 52 E4 20 49 6E 65 72 74 69 61 6C 4D 65 61 73 75 72 65 6D 65 6E 74 55 6E 69 74	

Table 131: Telegram structure: sRA InertialMeasurementUnit

Telegram structure: sRA InertialMeasurementUnit						
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 IMU data	String	23		InertialMeasurementUnit	49 6E 65 72 74 69 61 6C 4D 65 61 73 75 72 65 6D 65 6E 74 55 6E 69 74
Acceleration X	Vector with acceleration (including gravity) along the X axes.	Real as float according to IEEE754	4	Real in m/s^2	00000000h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
Acceleration Y	Vector with acceleration (including gravity) along the Y axes.	Real as float according to IEEE754	4	Real in m/s^2	00000000h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
Acceleration Z	Vector with acceleration (including gravity) along the Z axes.	Real as float according to IEEE754	4	Real in m/s^2	00000000h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
AngularVelocity X	Vector with rotation rates along the X coordinate axes.	Real as float according to IEEE754	4	Real in rad/s	00000000h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
AngularVelocity Y	Vector with rotation rates along the Y coordinate axes.	Real as float according to IEEE754	4	Real in rad/s	00000000h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
AngularVelocity Z	Vector with rotation rates along the Z coordinate axes.	Real as float according to IEEE754	4	Real in rad/s	00000000h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
MagneticField X	Vector with magnetic field strength along the X coordinate axes.	Real as float according to IEEE754	4	Real in T	00000000h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
MagneticField Y	Vector with magnetic field strength along the Y coordinate axes.	Real as float according to IEEE754	4	Real in T	00000000h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF

Telegram structure: sRA InertialMeasurementUnit						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Magnetic-Field Z	Vector with magnetic field strength along the Z coordinate axes.	Real as float according to IEEE754	4	Real in T	00000000h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
Orientation W	Orientation in the specified coordinate system given as quaternion	Real as float according to IEEE754	4		00000000h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
Orientation X	Orientation in the specified coordinate system given as quaternion	Real as float according to IEEE754	4		00000000h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
Orientation Y	Orientation in the specified coordinate system given as quaternion	Real as float according to IEEE754	4		00000000h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
Orientation Z	Orientation in the specified coordinate system given as quaternion	Real as float according to IEEE754	4		00000000h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
Time-Stamp	The time stamp in microseconds is synchronized to the sensor time since bootup.	Uint_32	4	µs	0h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF

Table 132: Example: sRA InertialMeasurementUnit

CoLa A	<STX>sRA{SPC}InertialMeasurementUnit{SPC}00000000{SPC}00000000{SPC}00000000{SPC}00000000{SPC}00000000{SPC}00000000{SPC}00000000{SPC}00000000{SPC}00000000{SPC}00000000{SPC}00000000{SPC}00000000{SPC}00000000{SPC}0<ETX>
	02 73 57 41 20 45 6E 61 62 6C 65 49 6E 65 72 74 69 61 6C 4D 65 61 73 75 72 65 6D 65 6E 74 55 6E 69 74 20 30 30 30 30 30 30 30 30 20 30 30 30 30 30 30 30 30 20 30 30 30 30 30 30 30 30 20 30 30 30 30 30 30 30 30 20 30 30 30 30 30 30 30 30 20 30 30 30 30 30 30 30 30 20 30 30 30 30 30 30 30 30 20 30 30 30 30 30 30 30 30 20 30 30 30 30 30 30 30 30 20 30 30 30 30 30 30 30 30 20 30 30 30 30 30 30 30 30 20 30 30 30 30 30 30 30 30 20 30 30 30 30 30 30 30 30 20 30 03
CoLa B	02 02 02 02 00 00 00 54 73 52 41 20 49 6E 65 72 74 69 61 6C 4D 65 61 73 75 72 65 6D 65 6E 74 55 6E 69 74 20 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 2C

13.1.4.4 Time stamp

13.1.4.4.1 Set time synchronization [sWN TSCRole]

Set the device synchronization mode.

Table 133: 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 134: 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 135: 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 136: 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

13.1.4.4.2 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 137: 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 138: 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 139: 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 140: 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:	0	00

Table 141: Example 1, 2: sAN LSPsetdatetime

CoLa A	<STX>sAN{SPC}LSPsetdatetime{SPC}0<ETX>
	02 73 41 4E 20 4C 53 50 73 65 74 64 61 74 65 74 69 6D 65 20 30 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 00 50

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

13.1.4.4.3 Set NTP (Network Time Protocol) parameters

13.1.4.4.3.1 Set time synchronization interface [sWN TSCTCInterface]

Define the interface for device time synchronization.

Table 142: 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 synchronization 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 143: 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 144: 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 synchronization interface	String	14		TSCTCInterface	54 53 43 54 43 49 6E 74 65 72 66 61 63 65

Table 145: 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

13.1.4.4.3.2 Set time server IP address [sWN TSCTCSrvAddr]

Define the IP address from which the device will receive the time synchronization.

Table 146: 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 147: 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 148: 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 149: 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

13.1.4.4.3.3 Set time zone [sWN TSCTCtimezone]

Table 150: 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 151: 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 152: 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

Telegram structure: sWA TSCTCtimezone						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command	Set time zone	String	13		TSCTCtimezone	54 53 43 54 43 74 69 6D 65 7A 6F 6E 65

Table 153: 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

13.1.4.4.3.4 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 154: 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 155: 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 156: 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 157: 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

13.1.4.5 Filters

13.1.4.5.1 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 22.

Table 158: 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
Thresh-old[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 159: 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	

Table 160: 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

1) 1)

Table 161: 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

13.1.4.5.2 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 23.

Table 162: 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 163: 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 164: 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
Command	Set average filter	String	13		LFPmeanfilter	4C 46 50 6D 65 61 6E 66 69 6C 74 65 72

Table 165: 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

13.1.4.5.3 Set echo filter [sWN FREchoFilter]

NOTE
Only available with firmware versions > V1.10.

Select which measurement value(s) shall be send via LMDscanata, if the meausrement of one angular position returns several distance values (see "Multi-echo analysis", page 19)

Table 166: Telegram structure: sWN FREchoFilter

Telegram structure: sWN FREchoFilter (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 echo filter	String	12		FREchoFilter	46 52 45 63 68 6F 46 69 6C 74 65 72
Status code	Code number	Enum_8	1	First echo: All echos: Last echo:	0 1 2	00 01 02

Table 167: Example: sWN FREchoFilter

CoLa A	<STX>sWN{SPC}FREchoFilter{SPC}1<ETX>	
	<STX>sWN FREchoFilter 1<ETX>	
	sWN FREchoFilter 1	
	02 73 57 4E 20 46 52 45 63 68 6F 46 69 6C 74 65 72 20 31 03	
CoLa B	02 02 02 02 00 00 00 12 73 57 4E 20 46 52 45 63 68 6F 46 69 6C 74 65 72 20 01 7E	
	73 57 4E 20 46 52 45 63 68 6F 46 69 6C 74 65 72 20 01	

Table 168: Telegram structure: sWA FREchoFilter

Telegram structure: sWA FREchoFilter						
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 echo filter	String	12		FREchoFilter	46 52 45 63 68 6F 46 69 6C 74 65 72

Table 169: Example: sWa FREchoFilter

CoLa A	<STX>sWA{SPC}FREchoFilter<ETX>
	02 73 57 41 20 46 52 45 63 68 6F 46 69 6C 74 65 72 03
CoLa B	02 02 02 02 00 00 00 11 73 57 41 20 46 52 45 63 68 6F 46 69 6C 74 65 72 20 70

13.1.4.5.4 Set sensitivity fog filter [sWN MCSenseLevel]

Filter out disturbances in the meausurement data caused by fog or steam.

Table 170: Telegram structure: sWN MCSenseLevel

Telegram structure: sWN MCSenseLevel (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	Sense level	String	12		MCSenseLevel	4D 43 53 65 6E 73 65 4C 65 76 65 6C
Sensitivity level	Enable or disable fog filter and Sense Level	Uint_8	1	Fog Filter off Fog Filter on	0 1	00 01

Table 171: Example: sWN MCSenseLevel

CoLa A	<STX>sWN{SPC}MCSenseLevel{SPC}1<ETX>	
	<STX>sWN MCSenseLevel 1<ETX>	
	sWN MCSenseLevel 1	
	02 73 57 4E 20 4D 43 53 65 6E 73 65 4C 65 76 65 6C 20 31 03	
CoLa B	02 02 02 02 00 00 00 10 73 57 4E 20 4D 43 53 65 6E 73 65 4C 65 76 65 6C 20 01 70	
	73 57 4E 20 4D 43 53 65 6E 73 65 4C 65 76 65 6C 20 01	

Table 172: Telegram structure: sWA MCSenseLevel

Telegram structure: sWA MCSenseLevel						
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	Sense level	String	12		MCSenseLevel	4D 43 53 65 6E 73 65 4C 65 76 65 6C

Table 173: Example: sWA MCSenseLevel

CoLa A	<STX>sWA{SPC}MCSenseLevel<ETX>
	02 73 57 41 20 4D 43 53 65 6E 73 65 4C 65 76 65 6C 20 03
CoLa B	02 02 02 02 00 00 00 0F 73 57 41 20 4D 43 53 65 6E 73 65 4C 65 76 65 6C 20 73

13.1.4.5.5 Activate Median filter [sWN LFPmedianfilter]

Activate a 3x1 Median filter (floating evaluation of 3 measurement points within one scan) for distance values.

Table 174: Telegram structure: sWN LFPmedianfilter

Telegram structure: sWN LFPmedianfilter (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

Telegram structure: sWN LFPmedianfilter (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command	Sector function	String	15		LFPmedianfilter	4C 46 50 6D 65 64 69 61 6E 66 69 6C 74 65 72
Status code	Enable or disable Median filter	Bool_1	1	Disable: Enable:	0 1	00 01
Reserved	-	Uint_16	2	Always:	3	00 03

Enable median filter

Table 175: Example: sWN LFPmedianfilter

CoLa A	<STX>sWN{SPC}LFPmedianfilter{SPC}1{SPC}3<ETX>
	<STX>sWN LFPmedianfilter 1 3<ETX>
	sWN LFPmedianfilter 1 3
	02 73 57 4E 20 4C 46 50 6D 65 64 69 61 6E 66 69 6C 74 65 72 20 31 20 33 03
CoLa B	02 02 02 02 00 00 00 17 73 57 4E 20 4C 46 50 6D 65 64 69 61 6E 66 69 6C 74 65 72 20 01 00 03 38
	73 57 4E 20 4C 46 50 6D 65 64 69 61 6E 66 69 6C 74 65 72 20 01 00 03

Table 176: Telegram structure: sWA LFPmedianfilter

Telegram structure: sWA LFPmedianfilter						
Telegram part	**Description**	**Variable**	**Length**	**Additoinal details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sWA	73 57 41
Command	Sector function	String	15		LFPmedianfilter	4C 46 50 6D 65 64 69 61 6E 66 69 6C 74 65 72

Table 177: Example: sWA LFPmedianfilter

CoLa A	<STX>sWA{SPC}LFPmedianfilter<ETX>
	02 73 57 41 20 4C 46 50 6D 65 64 69 61 6E 66 69 6C 74 65 72 03
CoLa B	02 02 02 02 00 00 00 14 73 57 41 20 4C 46 50 6D 65 64 69 61 6E 66 69 6C 74 65 72 20 35

13.1.4.6 Inputs and Outputs

13.1.4.6.1 Read Port Configration of all I/Os [sRN PortConfiguration]

Read the configuration of the digital inputs and outputs.

Table 178: Telegram structure: sRN PortConfiguration

Telegram structure: sRN PortConfiguration						
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 PortConfiguration						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command	Ask for port configuration	String	12		PortConfiguration	50 6F 72 74 43 6F 6E 66 69 67 75 72 61 74 69 6F 6E

Table 179: Example: sRN PortConfiguration

CoLa A	<STX>sRN{SPC}PortConfiguration<ETX>
	<STX>sRN PortConfiguration<ETX>
	sRN PortConfiguration
	02 73 52 4E 20 50 6F 72 74 43 6F 6E 66 69 67 75 72 61 74 69 6F 6E 03
CoLa B	02 02 02 02 00 00 00 15 73 52 4E 20 50 6F 72 74 43 6F 6E 66 69 67 75 72 61 74 69 6F 6E 26
	73 52 4E 20 50 6F 72 74 43 6F 6E 66 69 67 75 72 61 74 69 6F 6E

Table 180: Telegram structure: sRA PortConfiguration

Telegram structure: sRA PortConfiguration						
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	Configuration of all I/Os	String	12		PortConfiguration	50 6F 72 74 43 6F 6E 66 69 67 75 72 61 74 69 6F 6E
Start of loop, number of loops = amount of all current and future Inputs and Outputs of device family						
Port Type	Input or Output	Enum_8	1	Input: Output:	0 1	00 01
Port Name	Amount of characters of the following port name	Uint_16	2		0h ... 20h	00 00 ... 00 20
	Port name	String	16 (depending on string length)		[Port name]	[Port name]
Input Settings						
Logic	Logic of the input	Bool_1	1	Active high: Active low:	0 1	00 01
Debouncing	Select debouncing time in ms	Uint_16	2	(max. 10,000ms)	0h ... 2710h	00 00 ... 27 10
Reserved	-	Enum_8	1	Always:	0	00
Reserved	Reserved value 1	Uint_16	2		0	00 00
Reserved	Reserved value 2	Uint_16	2		0	00 00
Output Settings						

Telegram structure: sRA PortConfiguration						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Logic	Logic of the input	Bool_1	1	Active high: Active low:	0 1	00 01
Output Mode	PNP, NPN or Push-Pull	Enum_8	1	PNP: NPN: Push-Pull:	0 1 2	00 01 02
Restart type	Restart behavior of output after event: immediatly or after specific time	Enum_8	1	Immediately: Time:	0 1	00 01
Restart time	[Only with restart type = Time], time in ms	Uint_32	4	(20 ms ... 600,000 ms)	14h ... 927C0h	00 00 00 14 ... 00 09 27 C0
Combination	Combining multiple Events and/or Inputs	Enum_8	1	AND: OR: XOR:	0 1 2	00 01 02
Reserved	Reserved value 3	Uint_16	2		0	00 00
Reserved	Reserved value 4	Uint_16	2		0	00 00
Sources	Amount (n) of combined sources	Uint_16	2		0h ... FFFFh	00 00 ... FF FF
	Start of source loop, number of loops = amount of combined sources					
	Source name	String	4	Device Ready: Sopas Command: (xx = Number of output port) Indexsignal: SyncOutByClock: Input = Port Name: (INxx = Number of input port)	[Source] DRDY SCxx SROT SCLK INxx	[Source]
	Source Inverted or not	Bool_1	1	Not inverted: Inverted:	0 1	00 01
	Reserved value 5	Uint_8	1		0	00
	Reserved value 6	Uint_8	1		0	00
	Stop of source loop					
Reserved	Reserved value 7	Uint_16	2		0	00 00
Reserved	Reserved value 8	Uint_16	2		0	00 00
Reserved	Reserved value 9	Uint_16	2		0	00 00
Reserved	Reserved value 10	Uint_16	2		0	00 00
Stop of loop						

13.1.4.6.2 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 130.

Table 181: 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 182: 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 183: 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
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 184: 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

13.1.4.6.3 Read state of the outputs [sRN LIDoutputstate]

Status of all outputs

Table 185: 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 186: 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 187: 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 130.						

13.1.4.6.4 Receive outputstate by event [sEN LIDoutputstate]

Output telegram is sent every time an output state changes.

Table 188: 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

Telegram structure: sEN LIDoutputstate						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
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 189: 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 190: 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
Time	States code	Uint_16	2	No time data: Time data:	0 1	00 00 00 01

Telegram structure: sRA/sSN LIDoutputstate						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
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 191: 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}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 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 00 01 07 D9 02 12 0C 29 0E 00 09 75 E0 06

13.1.4.6.5 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 192: 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
Command	Set output state	String	12		mDOSetOutput	6D 44 4F 53 65 74 4F 75 74 70 75 74
Output state		Enum_8	1	Not active: Active:	0 1	00 01

Table 193: 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 194: 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 195: 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

13.1.4.6.6 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 130).

Table 196: 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 197: 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 198: 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 199: 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

13.1.4.7 Status

13.1.4.7.1 Read firmware version [sRN DeviceIdent]

Table 200: 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 201: 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 202: 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

Telegram structure: sRA DeviceIdent						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
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
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 203: 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

13.1.4.7.2 Read the device state [sRN SCdevicestate]

This telegram reads the general device state.

Table 204: 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 205: 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 206: 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

Telegram structure: sRA SCdevicestate						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command	Read the device state	String	13		SCdevicestate	53 43 64 65 76 69 63 65 73 74 61 74 65
Status code	Code number	Enum_8	1	Busy / logged-in: Ready: Error:	0 1 2	00 01 02

Table 207: 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

13.1.4.7.3 Read device order number [sRN DIornr]

This telegram reads the device order number.

Table 208: 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 209: 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 210: 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	00 00 00 00 00 00 00 ... FF FF FF FF FF FF FF

13.1.4.7.4 Read device type [sRN DItype]

This telegram asks for the device type of the product familiy.

Table 211: 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 212: 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 213: 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)

13.1.4.7.5 Read operating hours [sRN ODoprh]

Table 214: 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 215: 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 216: 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 217: 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

13.1.4.7.6 Read power on counter [sRN ODpwrc]

Shows the number of power on cycles.

Table 218: 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 219: 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 220: 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 221: 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

13.1.4.7.7 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 222: 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 223: 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 224: 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)**
Temperature data	[°C]	Real as float according to IEEE754	4	(-50 °C ... +100 °C)	C2480000h ... 42C80000h	C2 48 00 00 ... 42 C8 00 00

Example: sRA OPcurtmpdev (35 °C)

Table 225: 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

13.1.4.7.8 Set device name [sWN LocationName]

Give the device a specific description name such as its location.

Table 226: 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 227: 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 228: 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

Telegram structure: sWA LocationName						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command	Set device name	String	12		LocationName	4C 6F 63 61 74 69 6F 6E 4E 61 6D 65

Table 229: 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

13.1.4.7.9 Read device name [sRN LocationName]

Read the given name of the device (Default is the serial number of the device).

Table 230: 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 231: 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 232: 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 226, page 140						

13.1.4.7.10 FieldEvalLoggingList by Event [sEN FieldEvalLoggingList]

Start/Stop of the Event.

Table 233: Telegram structure: sEN FieldEvalLoggingList

Telegram structure: sEN FieldEvalLoggingList						
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		String	20		FieldEvalLog-gingList	46 69 65 6C 64 45 76 61 6C 4C 6F 67 67 69 6E 67 4C 69 73 74
		Bool_1	1	Start: Stop:	1 0	01 00

Table 234: Example sEN FieldEvalLoggingList 1

CoLa A	<STX>sEN{SPC}FieldEvalLoggingList{SPC}1<ETX>	
	<STX>sEN FieldEvalLoggingList 1<ETX>	
	sEN FieldEvalLoggingList 1	
	02 73 45 4E 20 46 69 65 6C 64 45 76 61 6C 4C 6F 67 67 69 6E 67 4C 69 73 74 20 31 03	
CoLa B	02 02 02 02 00 00 00 1A 73 45 4E 20 46 69 65 6C 64 45 76 61 6C 4C 6F 67 67 69 6E 67 4C 69 73 74 20 01 64	
	73 45 4E 20 46 69 65 6C 64 45 76 61 6C 4C 6F 67 67 69 6E 67 4C 69 73 74 20 01	

Acknowledge for Start/Stop of the Event

Table 235: Telegram structure: sEA FieldEvalLoggingList

Telegram structure: sEA FieldEvalLoggingList						
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		String	20		FieldEvalLog-gingList	46 69 65 6C 64 45 76 61 6C 4C 6F 67 67 69 6E 67 4C 69 73 74
		Bool_1	1	Start: Stop:	1 0	01 00

Table 236: Example: sEA FieldEvalLoggingList 1

CoLa A	<STX>sEA{SPC}FieldEvalLoggingList{SPC}1<ETX>
	02 73 45 41 20 46 69 65 6C 64 45 76 61 6C 4C 6F 67 67 69 6E 67 4C 69 73 74 20 31 03
CoLa B	02 02 02 02 00 00 00 1A 73 45 41 20 46 69 65 6C 64 45 76 61 6C 4C 6F 67 67 69 6E 67 4C 69 73 74 20 01 6B

This is a readonly telegram

Table 237: Telegram structure: sSN Field EvalLogging List

Telegram structure: sSN FieldEvalLogging List						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sSN	73 53 4E
Command		String	20		FieldEvalLog-gingList	46 69 65 6C 64 45 76 61 6C 4C 6F 67 67 69 6E 67 4C 69 73 74
Array size	0..100	Uint_16	2		0h ... 00 64h	00 00 ... 00 64
Index of Array	0..65535	Uint_16	2		0h ... FFFFh	00 00 .. FF FF
Year	0..2105	Uint_16	2		0h ... 839h	00 00 ... 08 39
Month	1..12	Uint_8	1		1h ... Ch	01 ... 0C
Day	1..31	Uint_8	1		1h ... 1Fh	01 ... 1F
Hour	0..23	Uint_8	1		0h ... 17h	00 ... 17
Minute	0..59	Uint_8	1		0h ... 3Bh	00 ... 3B
second	0..59	Uint_8	1		0h ... 3Bh	00 ... 3B
microsec-ond	0..999	Uint_32	4		0h ... 3E7	00 ... 03E7
Evaluation-Number	1..255	Uint_8	1		0h ... FFh	00 ... FF
X Coordinate	X-Position of first vio-lation in world coordi-nates [mm] -2147483648 ..21474 83647	Int_32	4			
Y Coordinate	Y-Position of first vio-lation in world coordi-nates [mm] -2147483648 ..21474 83647	Int_32	4			
Z Coordinate	Z-Position of first vio-lation in world coordi-nates [mm] -2147483648 ..21474 83647	Int_32	4			
RadialDis-tance	0..4294967295	Uint_32	4		0h ... FF FF FF FFh	00 00 00 00 ... FF FF FF FF
Horizonta-lAngle	-1800000.. 1800000	Int_32	4			
VerticalAn-gle	-9000..9000	Int_16	2			
ObjectSize in mm	0..4294967295	Uint_32	4		0h ... FF FF FF FFh	00 00 00 00 ... FF FF FF FF
Reserved 1	0..4294967295	Uint_32	4		0h ... FF FF FF FFh	00 00 00 00 ... FF FF FF FF
Reserverd 2	0..4294967295	Uint_32	4		0h ... FF FF FF FFh	00 00 00 00 ... FF FF FF FF

Table 238: Example: sSN Field EvalLogging List

CoLa A	<STX>sSN{SPC}FieldEvalLogging-List{SPC}64{SPC}1{SPC}7E4{SPC}C{SPC}16{SPC}9{SPC}3A{SPC}20{SPC}13{SPC}1{SPC}140{SPC}FFFFFEEC{SPC}0{SPC}1A7{SPC}FFF9B4AC{SPC}0{SPC}E2{SPC}0{SPC}0{SPC}2... <ETX>
	02 73 53 4E 20 46 69 65 6C 64 45 76 61 6C 4C 6F 67 67 69 6E 67 4C 69 73 74 20 36 34 20 31 20 37 45 34 20 43 20 31 36 20 39 20 33 41 20 32 30 20 31 33 20 31 20 31 34 30 20 46 46 46 46 46 45 45 43 20 30 20 31 41 37 20 46 46 46 39 42 34 41 43 20 30 20 45 32 20 30 20 30 20 32 ... 03
CoLa B	02 02 02 02 00 00 12 13 73 53 4E 20 46 69 65 6C 64 45 76 61 6C 4C 6F 67 67 69 6E 67 4C 69 73 74 20 00 64 00 01 07 E4 0C 16 09 3A 20 00 13 01 00 00 01 40 FF FF FE EC 00 00 00 00 00 00 01 A7 FF F9 B4 AC 00 00 00 00 00 00 E2 00 00 00 00 00 00 00 00 ... 00 00 7A

13.1.4.7.11 FieldEvalLoggingList on Request [sRN FieldEvalLoggingList]

Request the actual Field Eval logging List

Table 239: Telegram structure: sRN FieldEvalLoggingList

Telegram structure: sRN FieldEvalLoggingList						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Event	String	3		sRN	73 52 4E
Command		String	20		FieldEvalLog-gingList	46 69 65 6C 64 45 76 61 6C 4C 6F 67 67 69 6E 67 4C 69 73 74

Table 240: Example sRN FieldEvalLoggingList

CoLa A	<STX>sRN{SPC}FieldEvalLoggingList<ETX>	
	<STX>sRN FieldEvalLoggingList<ETX>	
	sRN FieldEvalLoggingList	
	02 73 52 4E 20 46 69 65 6C 64 45 76 61 6C 4C 6F 67 67 69 6E 67 4C 69 73 74 03	
CoLa B	02 02 02 02 00 00 00 18 73 52 4E 20 46 69 65 6C 64 45 76 61 6C 4C 6F 67 67 69 6E 67 4C 69 73 74 52	
	73 52 4E 20 46 69 65 6C 64 45 76 61 6C 4C 6F 67 67 69 6E 67 4C 69 73 74	

Table 241: Telegram structure: sRA FieldEvalLoggingList

Telegram structure: sRA FieldEvalLoggingList						
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	20		FieldEvalLog-gingList	46 69 65 6C 64 45 76 61 6C 4C 6F 67 67 69 6E 67 4C 69 73 74
Please check the list in sEN FieldEvalLoggingList (see "FieldEvalLoggingList by Event [sEN FieldEvalLoggingList]", page 141).						

Table 242: Example: sRA FieldEvalLoggingList

CoLa A	<STX>sRA{SPC}FieldEvalLogging- List{SPC}64{SPC}1{SPC}7E4{SPC}C{SPC}16{SPC}9{SPC}3A{SPC}20{SPC}13{SPC}1{SPC}140{SPC}FFFFFEEC{SPC}0{SPC}1A7{SPC}FFF9B4AC{SPC}0{SPC}E2{SPC}0{SPC}0{SPC}2...<ETX>
	02 73 52 41 20 46 69 65 6C 64 45 76 61 6C 4C 6F 67 67 69 6E 67 4C 69 73 74 20 36 34 20 31 20 37 45 34 20 43 20 31 36 20 39 20 33 41 20 32 30 20 31 33 20 31 20 31 34 30 20 46 46 46 46 46 45 45 43 20 30 20 31 41 37 20 46 46 46 39 42 34 41 43 20 30 20 45 32 20 30 20 30 20 32 20 30 20 32 ... 03
CoLa B	02 02 02 02 00 00 12 13 73 52 41 20 46 69 65 6C 64 45 76 61 6C 4C 6F 67 67 69 6E 67 4C 69 73 74 20 00 64 00 01 07 E4 0C 16 09 3A 20 00 13 01 00 00 01 40 FF FF FE EC 00 00 00 00 00 00 01 A7 FF F9 B4 AC 00 00 00 00 00 E2 00 00 00 00 00 00 00 00 00 02 ... F5

13.1.4.8 Interfaces

13.1.4.8.1 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 243: 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
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 244: 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 245: 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 246: 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

13.1.4.8.2 Read IP address [sRN EIIpAddr]

Read the IP address of the device.

Table 247: 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 248: 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 249: 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

Telegram structure: sRA EIIpAddr						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
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 250: 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

13.1.4.8.3 Set Ethernet gateway [sWN EIgate]

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 251: Telegram structure: sWN EIgate

Telegram structure: sWN EIgate (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		EIgate	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 252: Example: sWN EIgate 192.168.0.1

CoLa A	<STX>sWN{SPC}EIgate{SPC}C0{SPC}A8{SPC}00{SPC}01<ETX>	
	<STX>sWN EIgate C0 A8 00 01<ETX>	
	sWN EIgate 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 253: Telegram structure: sWA EIgate

Telegram structure: sWA EIgate						
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		EIgate	45 49 67 61 74 65

Table 254: Example: sWA EIgate

CoLa A	<STX>sWA{SPC}EIgate<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

13.1.4.8.4 Read Ethernet gateway [sRN EIgate]

Read for the Ethernet gateway (TCP/IP)

Table 255: Telegram structure: sRN EIgate

Telegram structure: sRN EIgate						
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		EIgate	45 49 67 61 74 65

Table 256: Example: sRN EIgate

CoLa A	<STX>sRN{SPC}EIgate<ETX>	
	<STX>sRN EIgate<ETX>	
	sRN EIgate	
	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 257: Telegram structure: sRA EIgate

Telegram structure: sRA EIgate						
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 gateway address	String	6		EIgate	45 49 67 61 74 65

Telegram structure: sRA EIgate						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
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 258: Example: sRA EIgate 192.168.0.1

CoLa A	<STX>sRA{SPC}EIgate{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

13.1.4.8.5 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 259: 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 260: 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 261: 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 262: 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

13.1.4.8.6 Read IP mask [sRN EImask]

Read the subnet mask of the device.

Table 263: 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 264: 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 265: 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

Telegram structure: sRA EImask						
Telegram part	Description	Variable	Length	Sensor	Values CoLa A (ASCII)	Values CoLa B (Binary)
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 266: 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

13.1.4.8.7 Set Host/ UDP port number [sWN EIHstPort, sWN EIUDPPort]

Define the TCP/IP host or UDP port number.

Table 267: 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 Ethernet host UDP/ IP	EIHstPort EIUDPPort	45 49 48 73 74 50 6F 72 74 45 49 55 44 50 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 268: 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	

Example: sWN EIUDPPort +2214 (UDP Port 2214)

Table 269: Example: sWN EIUDPPort +2214

CoLa A	<STX>sWN{SPC}EIUDPPort{SPC}+2214<ETX>
	<STX>sWN EIUDPPort +2214<ETX>
	sWN EIUDPPort +2214
	02 73 57 4E 20 45 49 55 44 50 50 6F 72 74 20 2B 32 32 31 34 03
CoLa B	02 02 02 02 00 00 00 10 73 57 4E 20 45 49 55 44 50 50 6F 72 74 20 08 A6 B0
	73 57 4E 20 45 49 55 44 50 50 6F 72 74 20 08 A6

Table 270: 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 Ethernet host UDP/ IP	EIHstPort EIUDPPort	45 49 48 73 74 50 6F 72 74 45 49 55 44 50 50 6F 72 74

Example: sWA EIHstPort (Host Port)

Table 271: 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

Example: sWA EIUDPPort (UDP Port)

Table 272: Example: sWA EIUDPPort

CoLa A	<STX>sWA{SPC}EIUDPPort<ETX>
	02 73 57 41 20 45 49 55 44 50 50 6F 72 74 03
CoLa B	02 02 02 02 00 00 00 0E 73 57 41 20 45 49 55 44 50 50 6F 72 74 20 11

13.1.4.8.8 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.

Table 273: 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

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	Set hardware settings of host interface	String	7		EIHstCola EIUDPCola	45 49 48 73 74 43 6F 6C 61 45 49 55 44 50 43 6F 6C 61
Command language	Host port Command language	Enum_8	1	CoLa ASCII: CoLa binary:	0 1	00 01

Example: sWN EIHstCola 1 (Host Port CoLa binary)

Table 274: 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

Example: sWN EIUDPCola 0 (UDP Port CoLa ASCII)

Table 275: Example: sWN EIUDPCola 0

CoLa A	<STX>sWN{SPC}EIUDPCola{SPC}0<ETX>
	<STX>sWN EIUDPCola 0<ETX>
	sWN EIUDPCola 0
	02 73 57 4E 20 45 49 55 44 50 43 6F 6C 61 20 30 03
CoLa B	02 02 02 02 00 00 00 0F 73 57 4E 20 45 49 55 44 50 43 6F 6C 61 20 00 06
	73 57 4E 20 45 49 55 44 50 43 6F 6C 61 20 00

Table 276: 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 EIUDPCola	45 49 48 73 74 43 6F 6C 61 45 49 55 44 50 43 6F 6C 61

Example: sWA EIHstCola (Host Port)

Table 277: 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

Example: sWA EIUDPCola (UDP Port

Table 278: Example: sWA EIUDPCola

CoLa A	<STX>sWA{SPC}EIUDPCola<ETX>
	02 73 57 41 20 45 49 55 44 50 43 6F 6C 61 03
CoLa B	02 02 02 02 00 00 00 0E 73 57 41 20 45 49 55 44 50 43 6F 6C 61 20 09

13.1.4.9 Application

13.1.4.9.1 Request minimal and maximal perpendicular distance once [sRN LFEperpdistresult]

Set the 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.

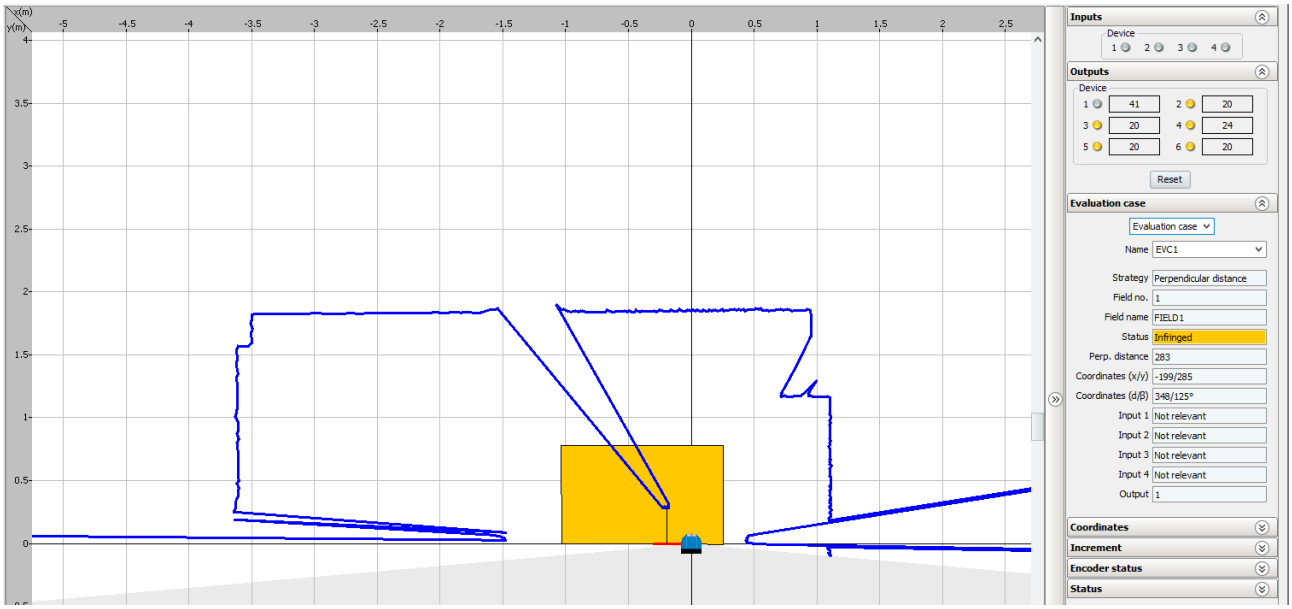


Table 279: Telegram structure: sRN LFEperpdistresult

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
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 280: 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 281: 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	1-16	UInt_16	2		0...16d (0 ... 10h)	00 00 .... 00 10
EVC number	1-16	UInt_8	1		0...16d (0 ... 10h)	0 ... 10
Perpendicular distance	0 m ... 64 m	UInt_32	4		0 ... 64000d (0 ... FA00h)	00 00 00 00 ... 00 00 FA 00
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]	0 m ... 64 m	Int_32	4		0 ... 64000d (0 ... FA00h)	00 00 00 00 ... 00 00 FA 00
Y-Pos [mm]	0 m ... 64 m	Int_32	4		0 ... 64000d (0 ... FA00h)	00 00 00 00 ... 00 00 FA 00
Reserved	default 0	Int_32	4			
Radial distance	0 m ... 64 m	UInt_32	4		0 ... 64000d (0 ... FA00h)	00 00 00 00 ... 00 00 FA 00
Radial distance angle	-47,5°... 227,5°	Int_32	4		-475000d ... +2275000d (FFF8C088h ... 22B6B8h)	FF F8 C0 88 ... 00 22 B6 B8
Reserved	default 0	Int_32	4		0	00 00 00 00
Reference line angle	-47,5°... 227,5°	Int_32	4		-475000d ... +2275000d (FFF8C088h ... 22B6B8h)	FF F8 C0 88 ... 00 22 B6 B8
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: sRA LFEperpdistresult (no required user level)						
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 282: 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{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 38

13.1.4.9.2 Request minimal and maximal perpendicular distance continiously on event [sEN LFEperp-distresult]

Set the 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.

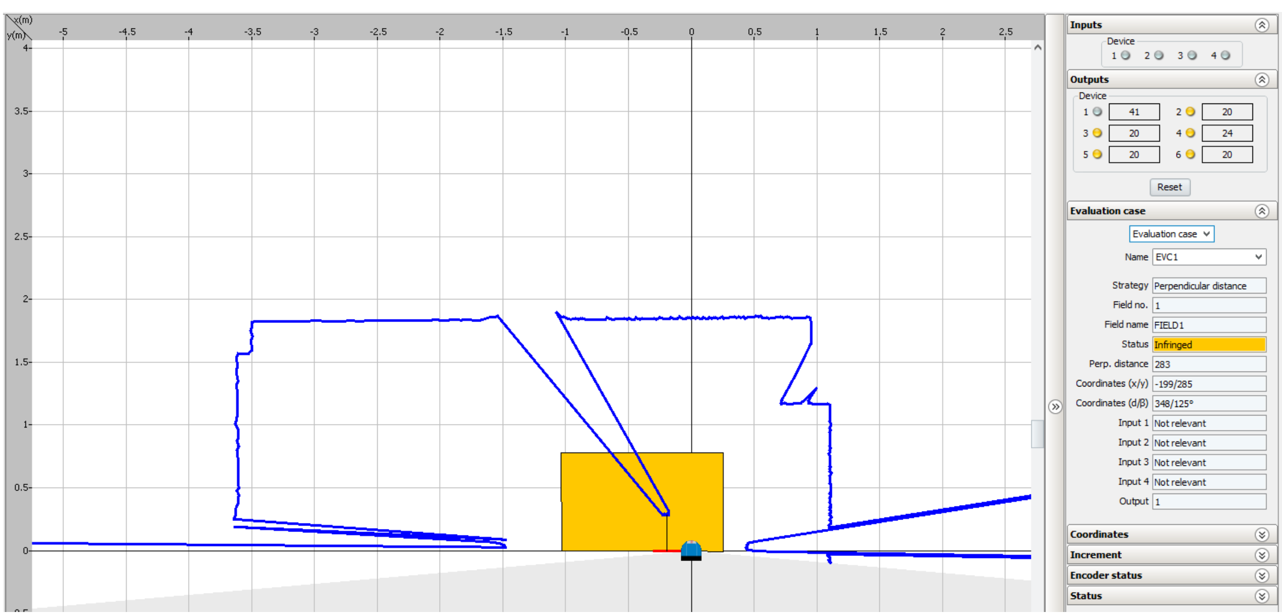


Table 283: Telegram structure: sEN LFEperpdistresult

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
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 284: 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 285: 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

Table 286: 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 287: 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

Telegram structure: sSN 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
Array	1-16	UInt_16	2		0...16d (0... 10h)	00 00 ... 00 10
EVC number	1-16	UInt_8	1		0 ...16d (0 ... 10h)	0 ... 10
Perpendicular distance	0 m ... 64 m	UInt_32	4		0 ... 64000d (0 ... FA00h)	00 00 00 00 ... 00 00 FA 00
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]	0 m ... 64 m	Int_32	4		-64000d ... 64000d (FFFF0600h ... FA00h)	FF FF 06 00 ... 00 00 FA 00
Y-Pos [mm]	0 m ... 64 m	Int_32	4		-64000d ... 64000d (FFFF0600h ... FA00h)	FF FF 06 00 ... 00 00 FA 00
Reserved	default 0	Int_32	4		0	00 00 00 00
Radial distance	0 m ... 64 m	UInt_32	4		0 ... 64000d (0 ... FA00h)	00 00 00 00 ... 00 00 FA 00
Radial distance angle	-47,5°... 227,5°	Int_32	4		-475000d ... +2275000d (FFF8C088h ... 22B6B8h)	FF F8 C0 88 ... 00 22 B6 B8
Reserved	default 0	Int_32	4		0	00 00 00 00
Reference line angle	-47,5°... 227,5°	Int_32	4		-475000d ... +2275000d (FFF8C088h ... 22B6B8h)	FF F8 C0 88 ... 00 22 B6 B8
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 288: 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 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 4A 00 00 01 5E 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 15

13.1.4.9.3 Set activate evaluation [sMN ActivateEvaluation]

Table 289: Telegram structure: sMN ActivateEvaluation

Telegram structure: sMN ActivateEvaluation (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	Activate evaluation	String	18		ActivateEvaluation	41 63 74 69 76 61 74 65 45 76 61 6C 75 61 74 69 6F 6E
Amount of evaluations to activate/ deactivate		Uint_16	2		1 ... 17	00 01 ... 00 11
Evaluation number		UInt_16	2		1 ... 17	00 01 ... 00 11
Activate/ Deactivate evaluation		Bool_1	1	Inactive: Active:	0 1	00 01

Table 290: Example: sMN ActivateEvaluation – Activate first evaluation

CoLa A	<STX>sMN{SPC}ActivateEvaluation{SPC}1{SPC}1{SPC}1<ETX>
	<STX>sMN ActivateEvaluation 1 1 1<ETX>
	sMN ActivateEvaluation 1 1 1
	02 73 4D 4E 20 41 63 74 69 76 61 74 65 45 76 61 6C 75 61 74 69 6F 6E 20 31 20 31 20 31 03
CoLa B	02 02 02 02 00 00 00 1C 73 4D 4E 20 41 63 74 69 76 61 74 65 45 76 61 6C 75 61 74 69 6F 6E 20 00 01 00 01 01 7E
	73 4D 4E 20 41 63 74 69 76 61 74 65 45 76 61 6C 75 61 74 69 6F 6E 20 00 01 00 01 01

Table 291: Telegram structure: sAN ActivateEvaluation

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

Telegram structure: sAN ActivateEvaluation (Run)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command	Activate evaluation	String	18		ActivateEvaluation	41 63 74 69 76 61 74 65 45 76 61 6C 75 61 74 69 6F 6E
Amount of activated/ deactivated evaluations		UInt_16	2		1 ... 17	00 01 ... 00 11
Success		Bool_1	1	Activation/Deactivation failed: Successfully activated/deactivated:	0 1	00 01

Table 292: Example: sAN ActivateEvaluation – Answer to successfully activated evaluation number 1

CoLa A	<STX>sAN{SPC}ActivateEvaluation{SPC}1{SPC}1{SPC}1<ETX>
	02 73 41 4E 20 41 63 74 69 76 61 74 65 45 76 61 6C 75 61 74 69 6F 6E 20 31 20 31 20 31 03
CoLa B	02 02 02 02 00 00 00 1A 73 41 4E 20 41 63 74 69 76 61 74 65 45 76 61 6C 75 61 74 69 6F 6E 20 00 01 01 73

13.1.5 Diagnostics

13.1.5.1 SOPAS error codes

sFA ErrorCode

Table 293: 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
Sopas_Error_BUFFER_UNDERFLOW	More data was expected, the allocated buffer could not be filled.	8	8

Telegram structure: sFA ErrorCode			
Error code	**Description**	**Dec.**	**Hex.**
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 294: 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

8020494/1SUL/2025-09-05/en

Australia
Phone +61 (3) 9457 0600
1800 33 48 02 – tollfree
E-Mail sales@sick.com.au

Austria
Phone +43 (0) 2236 62288-0
E-Mail office@sick.at

Belgium/Luxembourg
Phone +32 (0) 2 466 55 66
E-Mail info@sick.be

Brazil
Phone +55 11 3215-4900
E-Mail comercial@sick.com.br

Canada
Phone +1 905.771.1444
E-Mail cs.canada@sick.com

Czech Republic
Phone +420 234 719 500
E-Mail sick@sick.cz

Chile
Phone +56 (2) 2274 7430
E-Mail chile@sick.com

China
Phone +86 20 2882 3600
E-Mail info.china@sick.net.cn

Denmark
Phone +45 45 82 64 00
E-Mail sick@sick.dk

Finland
Phone +358-9-25 15 800
E-Mail sick@sick.fi

France
Phone +33 1 64 62 35 00
E-Mail info@sick.fr

Germany
Phone +49 (0) 2 11 53 010
E-Mail info@sick.de

Greece
Phone +30 210 6825100
E-Mail office@sick.com.gr

Hong Kong
Phone +852 2153 6300
E-Mail ghk@sick.com.hk

Hungary
Phone +36 1 371 2680
E-Mail ertekesites@sick.hu

India
Phone +91-22-6119 8900
E-Mail info@sick-india.com

Israel
Phone +972 97110 11
E-Mail info@sick-sensors.com

Italy
Phone +39 02 27 43 41
E-Mail info@sick.it

Japan
Phone +81 3 5309 2112
E-Mail support@sick.jp

Malaysia
Phone +603-8080 7425
E-Mail enquiry.my@sick.com

Mexico
Phone +52 (472) 748 9451
E-Mail mexico@sick.com

Netherlands
Phone +31 (0) 30 204 40 00
E-Mail info@sick.nl

New Zealand
Phone +64 9 415 0459
0800 222 278 – tollfree
E-Mail sales@sick.co.nz

Norway
Phone +47 67 81 50 00
E-Mail sick@sick.no

Poland
Phone +48 22 539 41 00
E-Mail info@sick.pl

Romania
Phone +40 356-17 11 20
E-Mail office@sick.ro

Singapore
Phone +65 6744 3732
E-Mail sales.gsg@sick.com

Slovakia
Phone +421 482 901 201
E-Mail mail@sick-sk.sk

Slovenia
Phone +386 591 78849
E-Mail office@sick.si

South Africa
Phone +27 10 060 0550
E-Mail info@sickautomation.co.za

South Korea
Phone +82 2 786 6321/4
E-Mail infokorea@sick.com

Spain
Phone +34 93 480 31 00
E-Mail info@sick.es

Sweden
Phone +46 10 110 10 00
E-Mail info@sick.se

Switzerland
Phone +41 41 619 29 39
E-Mail contact@sick.ch

Taiwan
Phone +886-2-2375-6288
E-Mail sales@sick.com.tw

Thailand
Phone +66 2 645 0009
E-Mail marcom.th@sick.com

Turkey
Phone +90 (216) 528 50 00
E-Mail info@sick.com.tr

United Arab Emirates
Phone +971 (0) 4 88 65 878
E-Mail contact@sick.ae

United Kingdom
Phone +44 (0)17278 31121
E-Mail info@sick.co.uk

USA
Phone +1 800.325.7425
E-Mail info@sick.com

Vietnam
Phone +65 6744 3732
E-Mail sales.gsg@sick.com

Detailed addresses and further locations at **www.sick.com**

SICK AG | Waldkirch | Germany | www.sick.com

SICK
Sensor Intelligence.