OPERATING INSTRUCTIONS

picoScan150

2D LiDAR sensors

SICK
Sensor Intelligence.

Described product

picoScan150

NOTE

The functional scope of the picoScan150 depends on the selected configuration. Certain functions are supported or not supported, depending on the configured variant. The operating instructions describe the full functional scope of the picoScan.

Manufacturer

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

Legal information

Original document

This document is an original document of SICK AG.

Contents

1 About this document

1.1 Information on the operating instructions

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

The operating instructions are an integral part of the product. Store the instructions so they remain accessible to staff at all times. If the product is 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 product is integrated. Information on this can be found in the operating instructions for the machine or system.

1.2 Symbols and document conventions

Warnings and other notes

DANGER
Indicates a situation presenting imminent danger, which will lead to death or serious injuries if not prevented.

WARNING
Indicates a situation presenting possible danger, which may lead to death or serious injuries if not prevented.

CAUTION
Indicates a situation presenting possible danger, which may lead to moderate or minor injuries if not prevented.

NOTICE
Indicates a situation presenting possible danger, which may lead to property damage if not prevented.

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

Instructions to action

► The arrow denotes instructions to action.

1. The sequence of instructions is numbered.
2. Follow the order in which the numbered instructions are given.

✓ The tick denotes the results of an action.

1.3 Further information

More information can be found on the product page.

The call is made 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

Important notes

NOTE
The functional scope of the picoScan150 depends on the selected configuration. Certain functions are supported or not supported, depending on the configured variant. The operating instructions describe the full functional scope of the picoScan150.

The picoScan150 2D LiDAR sensor is a non-contact distance measuring sensor with one scan plane. It has been designed for indoor or outdoor and mobile and stationary use in stand-alone operation.

Depending on the configuration and application software, the following usage scenarios can be solved:

- Detection of objects during continuous output of measurement data as required.
- Field monitoring of freely defined areas with signaling of object detection via digital outputs or telegrams.

The picoScan150 is suitable for applications that demand precise, non-contact optical optical contour measurements and environment perception. Typical fields of application are, for example, stationary field protection, area monitoring, access control, mobile applications (navigation and anti-collision of mobile platforms) as well as profile detection. It can also be used, for example, to implement systems for collision protection, object protection or access monitoring.

The product is designed for use in industrial and logistics areas, and meets the requirements for industrial ruggedness, interfaces and data processing.

The product was developed for use in industrial environments (EN 61000-6-4).

Incorrect use, improper modification, or tampering with the product will invalidate any warranty offered by SICK AG. Furthermore, SICK AG shall not accept any responsibility or liability for any resulting damage and consequential damage.

2.2 Improper use

Impermissible use

- As a safety component as defined in the relevant applicable safety standards for machines, e.g. Machinery Directive

Impermissible ambient conditions

- Explosion-hazardous area
- Corrosive environment

2.3 Cybersecurity

Overview

To protect against cybersecurity threats, it is necessary to continuously monitor and maintain a comprehensive cybersecurity concept. 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 Qualification of personnel

Any work on the product may only be carried out by personnel qualified and authorized to do so.

Qualified personnel are able to perform tasks assigned to them and can independently recognize and avoid any potential hazards. This requires, for example:

- technical training
- experience
- knowledge of the applicable regulations and standards

2.6 Basic safety notes

Please observe the safety notes and the warnings listed here and in other sections of this product documentation to reduce the possibility of risks to health and avoid dangerous situations.

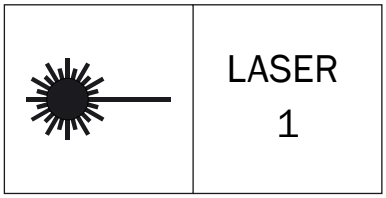
Danger due to visible radiation is product-specific. See the technical data for more information.

Laser notes

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.



This laser product complies with laser class 1 according to IEC 60825-1:2014 / EN 60825-1:2014+A11:2021.

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.

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.

Mounting and electrical installation

CAUTION
Risk of injury due to hot device surface.

The surface of the device can become hot.

- Before performing work on the device (e.g. mounting, cleaning, disassembly), switch off the device and allow it to cool down.
- Ensure good dissipation of excess heat from the device to the surroundings.

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.

Repairs and modifications

WARNING
Electric shock!

Non-insulated electrical conductors are located in the housing. Electrical voltage can cause severe injury or death.

- Do not open the housing.
- Protect the housing from damage.
- If the housing is damaged, disconnect the device from the voltage supply and do not put it into operation.

3 Product description

3.1 Scope of delivery

Depending on the chosen product version, the scope of delivery of a product will include the following components:

Table 1: Scope of delivery

No. of units	Component	Note
1	Product in the ordered type (complete device or basic device). The functional scope of the product depends on the ordered configuration.	**Complete device:** • Components are mounted at the factory (housing and system plug). **Basic device:** • Mount the housing and system plug separately.
1	Printed safety notes, multilingual	Brief information and general safety notes

The actual scope of delivery may differ for special designs, additional orders or due to the latest technical changes.

Further topics

- Mounting the system plug on the product

3.2 Product overview

Product overview

① Status indicator LED 1
② Status indicator LED 2
③ Optics cover

① Marking for the position of the light emission level
② Ventilation element (do not remove/paint over)
③ System plug (mounted at the back)

Further topics

- Dimensional drawing

3.3 Display and control elements

Overview

Figure 1: Position of the two status LEDs

① LED1
② LED2

Status LEDs

LED1 (color)	LED2 (color)	Description
● (Red)	● (Red)	Start-up, parameterization, firmware update, correctable error
-	-	Off
☀ (Red)	☀ (Red)	Fatal error

LED1 (color)	LED2 (color)	Description
-	(Green)	On / Ready for operation
-	(Yellow)	Warning
(Green)	(Yellow)	Restart after time or digital input
(Green)	(Yellow)	Contamination warning
(Red)	(Yellow)	Contamination error
(Green) (Yellow) (Red)	(Green) (Yellow) (Red)	Identify product
(Yellow)	(Green)	Object detected

● = illuminated; ✺ = flashing

Further topics

- Enable/ disable LEDs [sWN LEDEnable]
- Read state of LEDs [sRN LEDState]

3.4 Type label

Product

Information for identifying the product is located on the left side of the product.

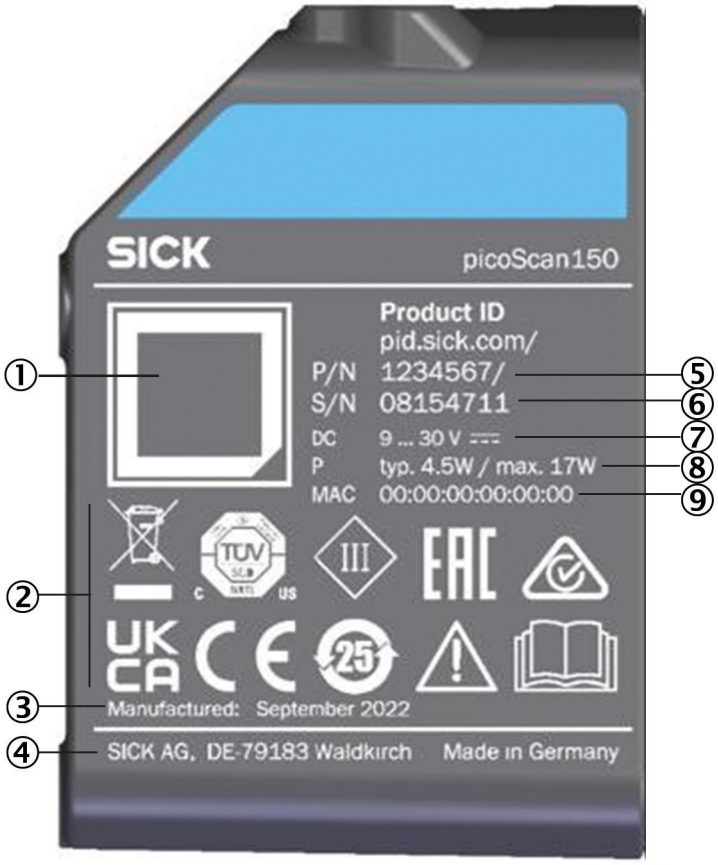


Figure 2: picoScan150 type label (example)

① Data Matrix code with product data and link to product page
② Conformity mark/certification mark, protection class, symbol: Observe the operating instructions!
③ Production date
④ Manufacturer, place of production
⑤ Part number
⑥ Serial number
⑦ Voltage supply
⑧ Typical power, max. power
⑨ MAC address

Male connector

Information for identifying the male connector is located on the connector.

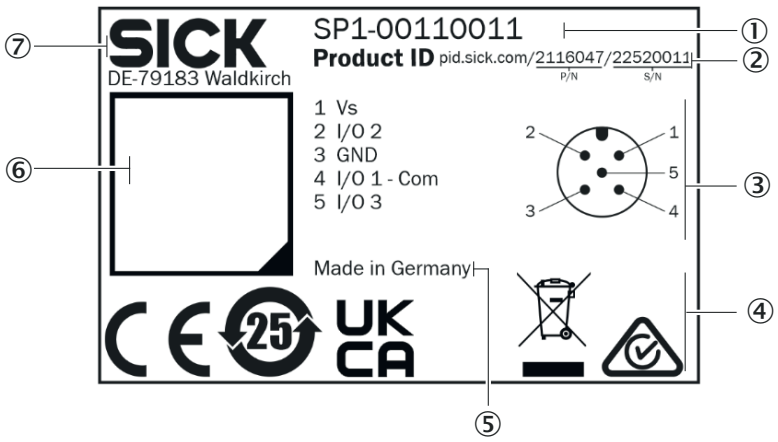


Figure 3: Type label for system plug (example)

① Type code
② Product ID with part number (P/N) and serial number (S/N)
③ Pin assignment or wire colors
④ Conformity mark/certification mark
⑤ Production site
⑥ Data Matrix code with product data and link to product page
⑦ Manufacturer

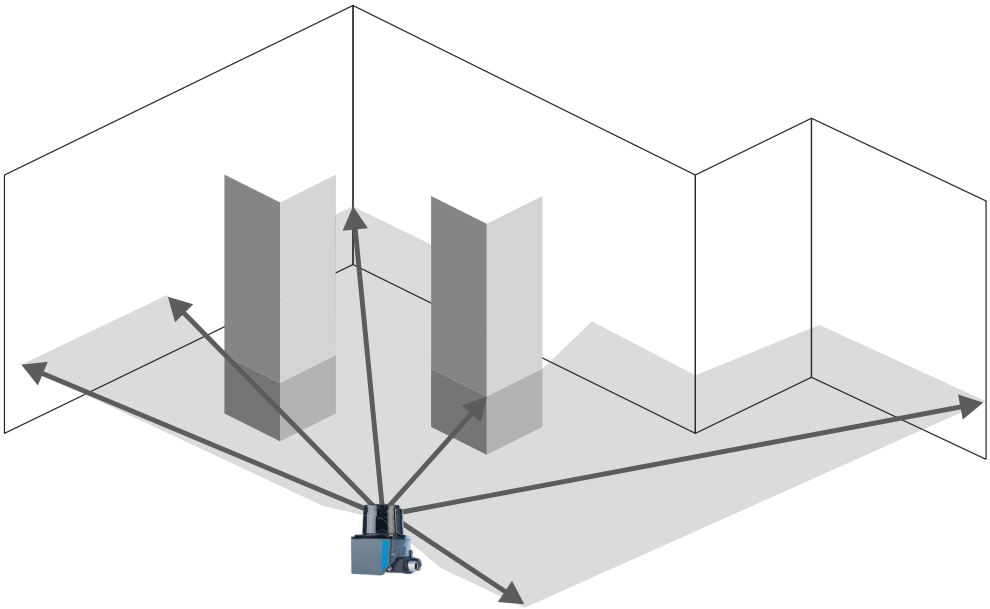
Further topics

- Read serial number [sRN SerialNumber]

3.5 Principle of operation

3.5.1 Measurement principle

The product is an opto-electronic LiDAR sensor (laser scanner) that uses laser beams for non-contact scanning of the contour of its surroundings on a plane. The product 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 direction.



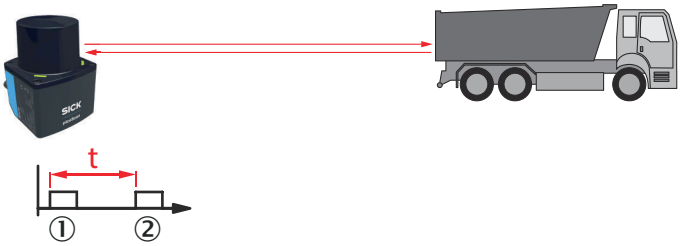
3.5.2 Distance measurement

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

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 product.

The product uses SICK's own HDDM+ (High Definition Distance Measurement plus) technology. With this measurement procedure, a measured value is formed from the statistical evaluation of multiple single pulses.

When configured at 40 Hz, for example, and an angular resolution of 0.25°, the product evaluates up to 132600 measuring points per second in multi-echo mode.



① Emitted pulse

② Receive pulse

The measured value consists not only of a single time-of-flight measurement, but includes evaluated information from numerous subpulses. This ensures a significantly more stable time and distance measurement. This gap-free scanning for objects leads to a high sensitivity even for small objects.

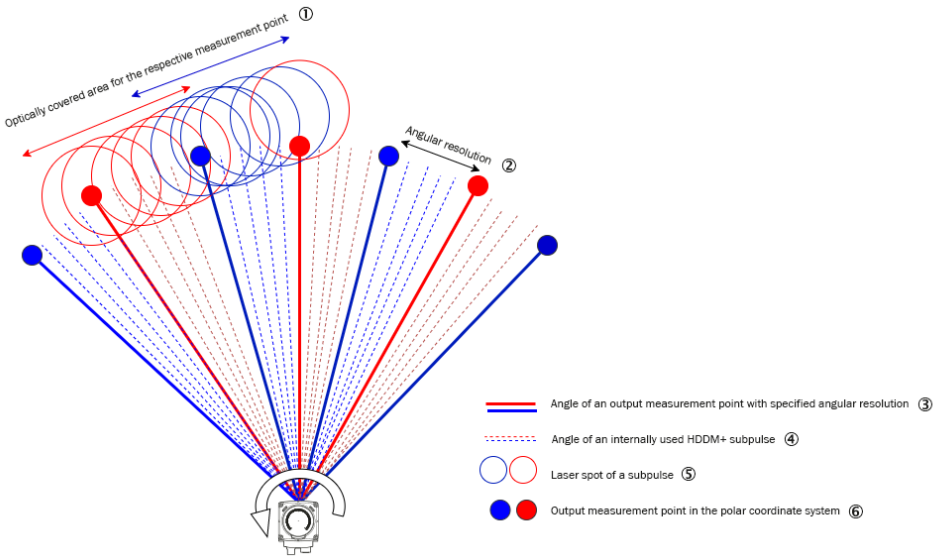


Figure 4: Gap-free scanning using the HDDM+ measurement procedure (schematic illustration)

① Optically covered area for the respective measurement point
② Angular resolution
③ Angle of an output measurement point with specified angular resolution
④ Angle of an internally used HDDM+ subpulse
⑤ Laser spot of a subpulse
⑥ Output measurement point in the polar coordinate system

- Spots overlap depending on the distance, gap-free, note divergence
- Number of subpulses differs depending on the selected sensor profile
- Information from the subpulses can be used in two measuring point calculations

The product uses a rotating mirror to deflect the emitted laser beams, thereby scanning its surroundings in a circular pattern. The angle information is generated by an internal encoder.

3.5.3 Multi-echo analysis

The distance between the product and an object is calculated via the time-of-flight of the emitted and then received again pulse. The product can evaluate up to three echo signals for each measuring beam to deliver reliable measurement results, even under adverse ambient conditions.

To be able to display echoes as separate measured values, there must be a minimum distance of approx. 0.5 m between the two objects.

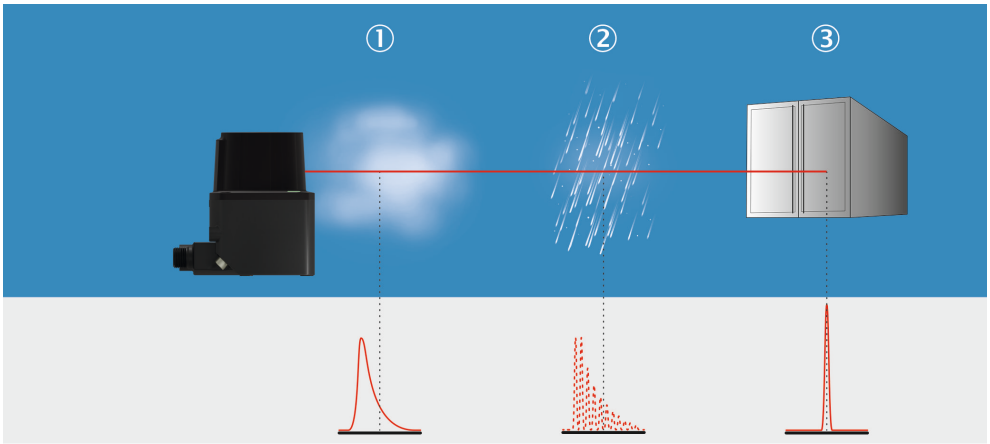


Figure 5: Multi-echo analysis using the example of building management

① Fog
② Rain
③ Measuring object

3.5.4 Dynamic Sensing Profiles

LiDAR sensors are used for a variety of sub-applications. Within these sub-applications, the requirements can change dynamically. When used in an AMR (Autonomous Mobile Robot), for example, different sensor characteristics are required depending on the area in which the AMR is employed. For example, collision protection in an area with a lot of traffic requires the shortest possible response time. If the AMR is traveling in a wide area of a warehouse, however, a large scanning range and good angular resolution is needed. When an AMR docks with a roller conveyor or a loading station, for example, the response time is no longer of such high priority and the high angular resolution to precisely detect the close range becomes more important.

If an application requires a very high angular resolution, this necessitates a low scanning frequency and vice versa. Which functions have priority should be prioritized for each specific application.

The product therefore offers a Dynamic Sensing Profiles feature, which enables the measurement core parameters to be dynamically adjusted to the specific requirements in subtasks of the LiDAR sensor. This is done during operation and without the need to restart the product.

The product therefore comes with a Scan Data Configuration function that allows quick switching between profiles.

The following profiles are available:

Profile number	Frequency in Hz	Angular resolution in °	Possible use
Profile 1	15	0.5	Object detection with large scanning ranges
Profile 2	15	0,33	TiM series retrofit
Profile 3	20	0.1	Fine positioning or recording of environment maps
Profile 4	20	0.25	Low noise measurements
Profile 5	25	0.25	LMS100 series retrofit
Profile 6	30	0.1	Fine positioning or recording of environment maps

Profile number	Frequency in Hz	Angular resolution in °	Possible use
Profile 7	40	0.25	Fast object detection at a large scanning range
Profile 8	50	0.25	Very fast object detection
Profile 9	15	0.05	Highest angular resolution
Profile 10	40	0.125	High angular resolution and high scanning frequency.
Profile 11	15	1°	TiM series retrofit

Further topics

- Set Performance Profile [sWN PerformanceProfileNumber]

3.5.5 Restrict angle range

The horizontal angle range that is output per scan can be restricted by increasing the start angle (theta start) or reducing the stop angle (theta stop).

It can be set within a range of –138° to +138°. Please note that the angle beam orthogonal to the front side is defined as 0°, and the direction of rotation of the product is set to counterclockwise. The measurement laser is only activated in the selected angle range.

For the compact format: If a complete segment of the output data is outside the angle range, it is not output. If a segment is partially within the specified angle range, it is filled with 0 values.

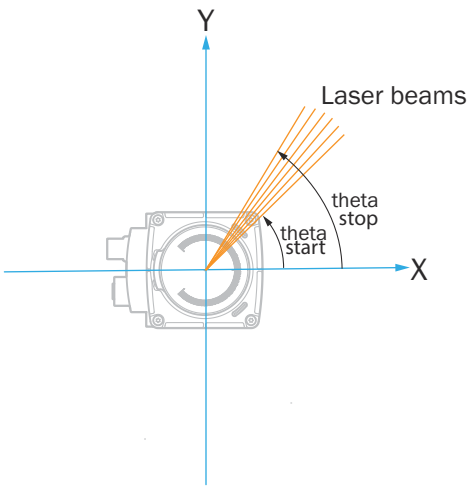


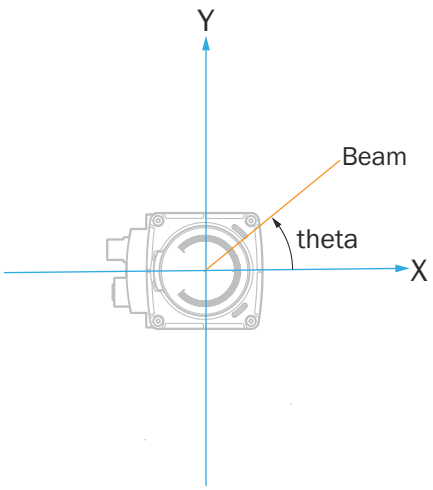
Figure 6: Definition of the theta start and theta stop angles (top view)

3.5.6 Coordinate system

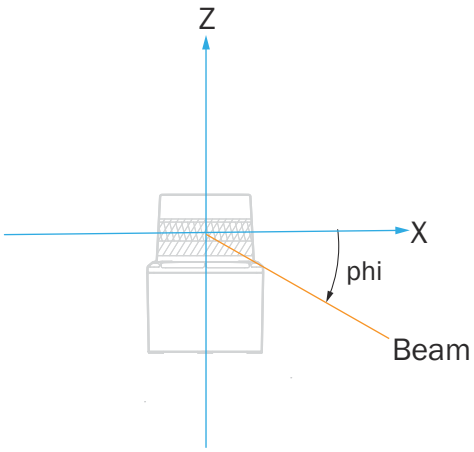
Device coordinate system

The origin of the device coordinate system (X=0, Y=0, Z=0) is a single point that serves as the origin and reference for all laser beams and the distance measurement of the product. When no translation is applied to the product in the world coordinate system, this point coincides with the origin of the world coordinate system.

The azimuthal (horizontal) angle of a beam is called theta. The beam at zero azimuth angle lies in the middle of the main viewing direction of the product so the scan is symmetrical.



The elevation angle (vertical angle) of a beam is designated phi and is measured relative to the x-y plane. For single plane sensors like the picoScan, the elevation angle is 0° when ideally mounted. Note the scan field flatness in regard to this see "Features", page 61.



The data is always output in the device coordinate system, not in the world coordinate system.

3.5.7 Filter

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

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

The filters – if present – are applied in the following order:

- Fog filter
- Echo filter
- Particle filter
- Moving average filter
- The filters for data reduction as well as the region of interest filter are then used.

3.5.7.1 Fog filter

The fog filter enables the product to eliminate unwanted echoes at close range (approx. 3 m). This considerably lowers the probability of false activations at close range in fog.

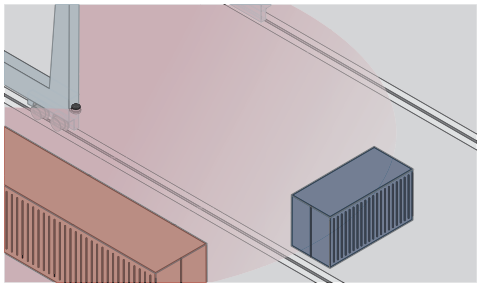


Figure 7: Without the fog filter: objects are difficult to detect through the fog due to reflections.

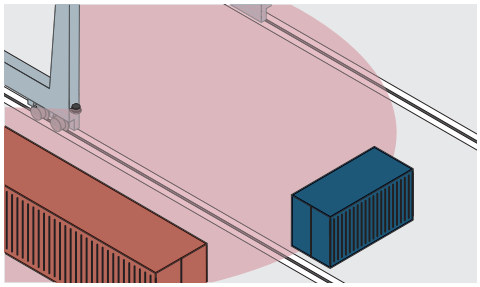


Figure 8: Using the fog filter: objects can be detected reliably because unwanted echoes are screened out.

Further topics

- Set sensitivity fog filter [sWN MCSenseLevel]

3.5.7.2 Echo filter

The echo filter screens out unwanted measurement data 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 or output.

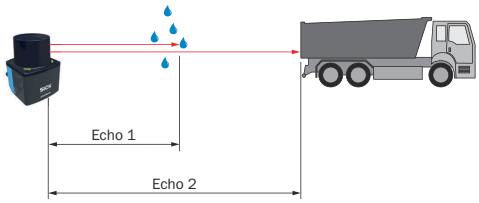


Figure 9: Without the echo filter: The product receives unwanted echoes from ambient conditions such as rain.

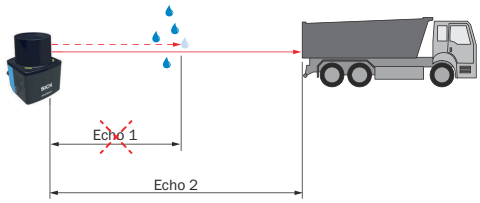


Figure 10: Using the echo filter (setting: last echo): the product screens out unwanted echoes from ambient conditions as per the settings chosen.

Further topics

- Set echo filter [sWN FREchoFilter]

3.5.7.3 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. If the distance between a measured value and its temporal spatial neighbors is greater than a defined threshold value, this measured value is discarded as faulty and replaced by the last valid measured value.

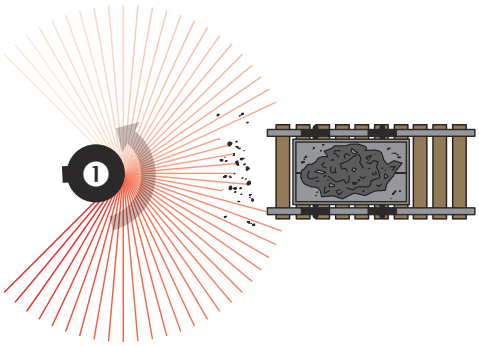


Figure 11: Without the particle filter: Violation of the contour due to dust particles in the vicinity of the object.

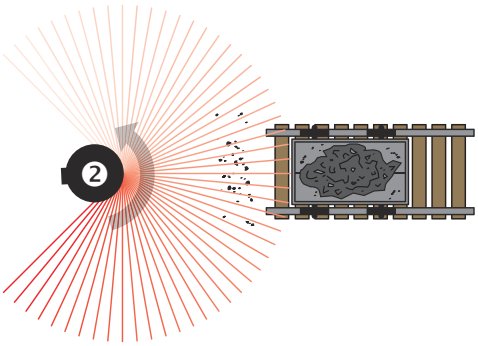


Figure 12: Using the particle filter: The response to dust particles in the detection field is delayed by one scan. Particles can thereby be blanked.

Important notes

NOTE
If the particle filter is activated, measurement data output is delayed by one scan.

Further topics

- Set particle filter [sWN LFPparticle]

3.5.7.4 Moving average filter

The sliding average filter smooths the distance value. It does this by calculating the arithmetic mean from several scans of the same point. The number of scans is configurable (2 ... 4 scans).

Table 2: Example: Moving average filter over 4 scans

	Angle (distance values in mm)									
Scan	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**...**
1	0	0	1100	1100	1150	1150	1380	1380	0	...
2	0	0	1200	1200	1190	**950**	1500	1500	0	...
3	0	0	1150	**1450**	1200	1200	1450	1450	0	...
4	0	0	1170	1170	1220	1220	1470	**1150**	0	...
1. Output value (scan 1-4)	0	0	1155	1230	1190	1130	1450	1370	0	...
5	0	0	**0**	1110	1150	1150	1380	1380	0	...
2. Output value (scan 2-5)	0	0	1173	1233	1190	1130	1450	1370	0	
6	0	0	1200	1210	1190	**0**	1500	1500	0	...
3. Output value (scan 3-6)	0	0	1173	1235	1190	1190	1450	1370	0	
7	0	**730**	1150	**0**	1200	1200	1450	1450	0	...
4. Output value (4-7)	0	730	1173	1163	1190	1190	1450	1370	0	
...	...	...	...	...	...	...	...	...	...	...

Individual outliers (shown in **bold** in the table) influence the average value.

After activating the filter, the first measured value is not output until after the configured number of scans. Therefore, there is always a time delay equivalent to the number of scans configured for averaging. The scan counter is taken from the latest scan included in the averaging process. Invalid distance values (= 0) are not included in the averaging calculation, so that in these places a smaller number of scans is used in the division calculation.

Further topics

- Set moving averaging filter [sWN LFPmovingAveragingFilter]

3.5.7.5 Data reduction filter

A data reduction filter is an algorithm that selects, based on various criteria, the relevant measurement data that should be excluded from the further processing.

3.5.7.5.1 Interval filter

The interval filter reduces the scan output rate by a configurable factor (reduction factor). When the reduction factor is set to three, for example, the output rate is reduced to one third. In this case only every third scan is output.

Further topics

- Set interval filter [sWN LFPintervalFilter]

3.5.7.6 Region of interest

A region of interest filter is an algorithm that uses various criteria to select relevant measurement data that are set to a distance value and RSSI value of zero for further processing. Angle values are retained.

3.5.7.6.1 Rectangular filter

When the rectangular filter is activated, it cuts out everything except for the parts of the scan within a rectangle. Note that this filter does not reduce the data, it sets the data points outside the rectangle to zero. The rectangle can be adjusted by setting the minimum and maximum values [mm] for the X and Y axis.

Further topics

- Set cubic area filter [sWN LFPcubicareafilter]

3.5.7.6.2 Distance filter

The distance filter affects the display of a circular area around the product by limiting the minimum and maximum radial distance that is measured.

The distance filter does not reduce the data, but sets the data points outside the radial distance to zero.

To keep a circular area, set the max area to a specific radius [mm].

To keep a ring, set the max range to a specific radius [mm] and the min range to >0 mm.

Further topics

- Set sensitivity fog filter [sWN MCSenseLevel]

3.5.8 Measurement data output

3.5.8.1 Data formats

The product offers two data output formats: MSGPACK and Compact. Both data formats allow the data to be output segment by segment via UDP/IP.

Both data formats contain information such as serial number and time stamp. While this enables MSGPACK to be easily integrated and parsed by existing libraries, it requires more bandwidth than the compact data format due to the data types used. Compact is more efficient and requires a lower bandwidth. Compact and MSGPACK support both UDP/IP Singlecast and UDP/IP Multicast.

When changing from TiM series or LMS series sensors to picoScan150, you can easily add the `LMDscandata` data format. It complements the Compact and MSGPACK data formats included as standard in the picoScan150.

The product has native ROS support for integration in a ROS environment.

Further topics

- Data format description technical information: **www.sick.com/8028133**
- Telegram listing (EN)
- Native ROS2

3.5.8.2 Segmented data output

The product records data over a scan range of 276°. The data acquired within a 276° rotation for the scan layer are referred to below as a scan. For data output, a scan is divided into segments each of which contain the scan layer data for a smaller angular range.

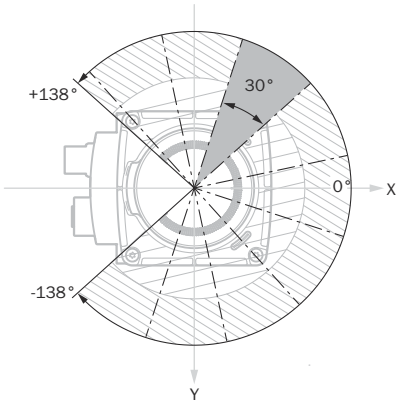


Figure 13: Example segment output at 276° scan range at 25 Hz

The segment size depends on the scanning frequency

- 30° at a scanning frequency ≤ 25 Hz
- 60° at a scanning frequency ≥ 30 Hz

The number of segments depends on the configured horizontal scan range.

Examples:

- 276° scan range at 25 Hz = 10 segments
- 276° scan range at 40 Hz = 5 segments
- 180° scan range (-90° ... +90°) at 40 Hz = 4 segments

If the scan range is adjusted, the segment boundaries remain.

In the Compact format, the measured values in hidden angle ranges are filled with zeros. If a complete segment is hidden, it is omitted from the data output.

Complementary information

- A code example can be found at: github.com/SICKAG/ScanSegmentAPI

3.5.8.3 Latency of the measurement data output

The latency value describes the data output latency between an HDDM+ laser pulse packet (measured value) and the transmission of the processed segment to a client system.

The specified latency value applies to a measurement scenario with many echoes and without the use of internal filters. In addition to this delay time, network delays, e.g. due to cables or Ethernet switches, must also be taken into account.

3.5.8.4 Native ROS2

The product provides native ROS support with the help of the Data Distribution Service (DDS) to enable efficient real-time communication and interoperability with ROS2 standards. To activate the feature, the corresponding license needs to be installed via the License Manager, and data output for MSGPACK, Compact or LMDscandata deactivated. The function can then be activated via OpenAPI using **RosConnectEnable**.

Further topics

- Communication interfaces

3.5.8.4.1 ROS driver

Suitable drivers for integrating the product into the ROS (Robot Operating System) are available for download 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).

3.5.9 Object sizes

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

The more laser pulses (HDDM+ subpulses) that hit an object, the more reliably it will be detected. Other factors that can affect the object detection are the angle of incidence of the laser beam, and the remission and surface properties of the object. Ambient conditions (e.g. ambient light) as well as movement of the product or the object can also affect the object detection.

Figure 14: Beam expansion

① Expanded laser beam

② Optical axis

Formula

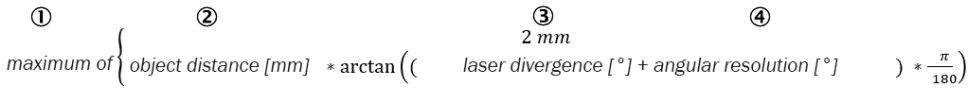


Figure 15: Formula for the minimum object size at a certain distance

① Maximum of
② Object distance In mm
③ Laser divergence in °
④ Angular resolution in °

The angular resolution can be adjusted to suit the product variant. As a general rule, the lower the configured angular resolution, the smaller the objects that can be detected.

$$\left\{4000\ mm\ *\ \arctan\left((0.27° + 0.1°) * \frac{\pi}{180}\right)\right. = 26\ \text{mm}$$

Figure 16: Example calculation: minimum object size at 4 m distance for an angular resolution of 0.1°

Diagrams

The following diagrams show typical minimum object sizes at various distances.

For reasons of improved readability, the near range up to 4 m is shown in a separate diagram.

When using the diagrams, note the specified max. scanning ranges depending on the device variant.

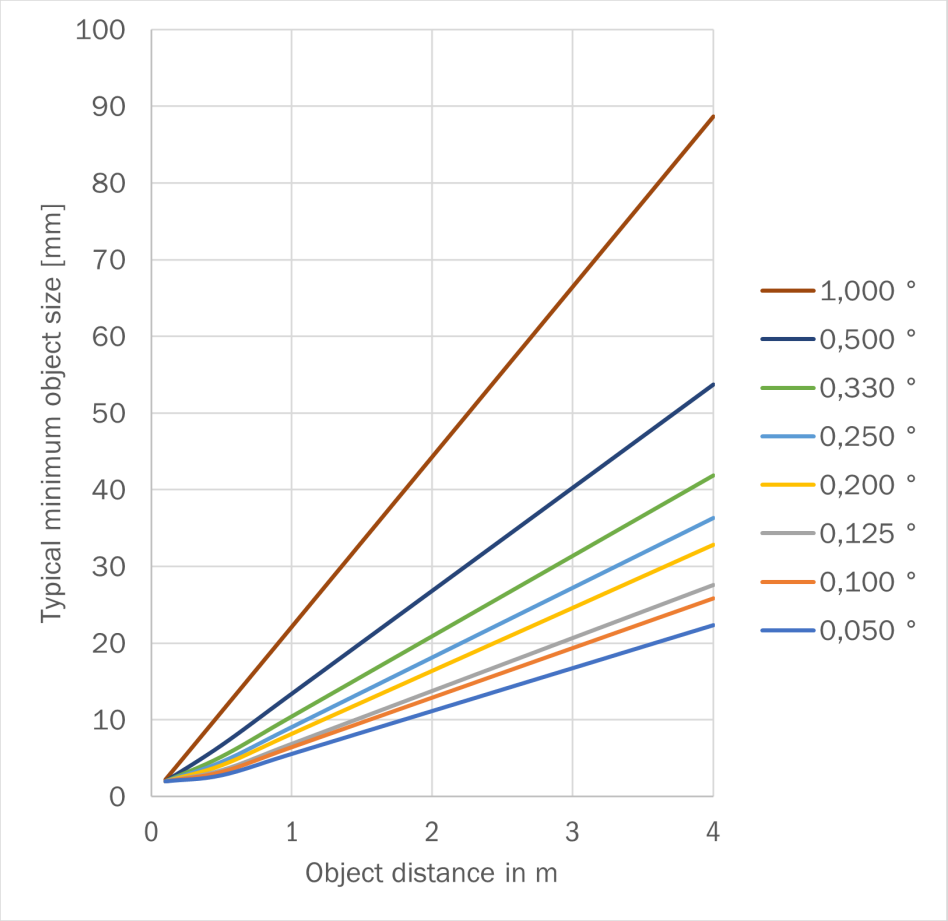


Figure 17: Typical minimum object size up to 4 m

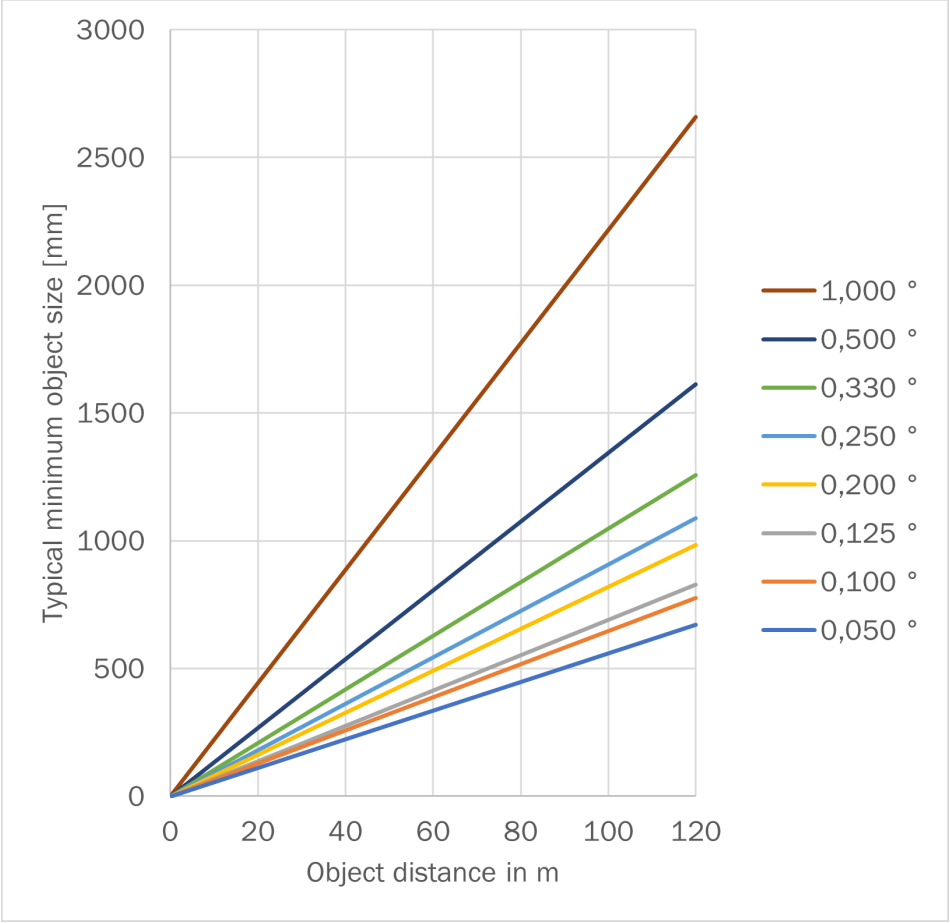


Figure 18: Typical minimum object size up to 120 m

Further topics

- Data sheet
- Dynamic Sensing Profiles

3.5.10 Impact of object surfaces on the measurement

Reflectance

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.

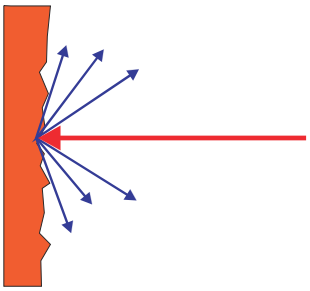


Figure 19: Reflection of light on the surface of the object

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.

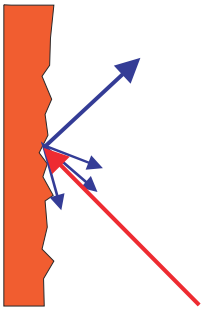


Figure 20: Angle of reflection

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 LiDAR sensor. Plastic reflectors (cat's eyes), reflective tape, and triple prisms have these properties.

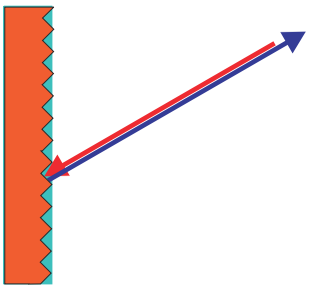


Figure 21: Retroreflection

Specular 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.

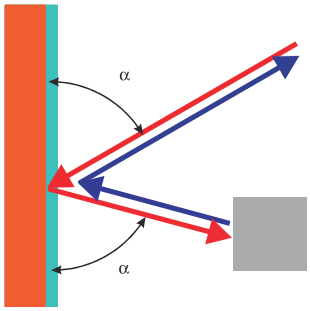


Figure 22: Specular surfaces

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.

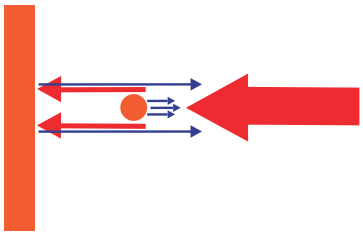


Figure 23: Object smaller than the laser beam diameter

3.5.11 Reflector detection

To determine the presence of a reflector, a value based on an evaluation of the energy reflected from the target (Received Signal Strength Indicator, RSSI) is often used. Several difficulties arise with this approach, particularly in distinguishing between bright targets and a reflector in the vicinity of the product.

The product therefore has a special reflector detection integrated into it. This function uses not only the RSSI value, but also other internally available information (e.g. signal characteristics, distances, echoes or signal delays) to ensure a near error-free reflector detection.

A reflector marking is assigned to each distance measurement value generated by a reflector. This reflector marking is available in the measurement data telegram (Compact/MSGPACK only) or can be displayed in the SOPASair web server.

Important notes

At close range, reflectors may appear wider than they really are.

Reflector detection at long range

Different settings can be selected to ensure a reflector detection with high availability. In the product under **Defaults** > **Sensitivity**, you can select various modes that offer the following optimizations:

- **Standard**
- Recommended: Setting that delivers good results in a wide range of applications. The reflector size should be 1.5 times the angular resolution depending on the distance.

- **Optimized for reliability**: This setting can be used to suppress or reduce unwanted measurement artefacts. This makes the sensor less sensitive. Caution: Targets with a very low reflectivity are only detected with difficulty.
- **Optimized for low remission**: This setting makes it possible to also detect objects with a low reflectivity. This will also lead to more frequent detection of interfering factors such as ambient light or particles, and undesired measurement artefacts can be created. This setting is not suitable for applications focusing on reflectors because it has been optimized for targets with a low remission and reflectors have a very high remission.

Further topics

- Data sheet
- Set Sensitivity Mode [sWN SensitivityMode]

3.5.12 RSSI

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 value 0 (zero) means that the received energy was too low to produce a valid measured value and also represents the lowest possible RSSI value. An RSSI value of 1 represents the highest possible measured value. A linear interpolation is applied between the values 0 and 1 using a resolution and scaling specific to the data format.

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 very 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 Inertial measuring unit (IMU)

The product is equipped with an inertial measuring unit (IMU). This can be used to identify movements of the product. The IMU can output accelerations in X, Y and Z as well as the position angle in yaw, pitch and roll. The orientation of the IMU data is based on the coordinate system of the product.

① Roll angle
② pitch angle
③ Yaw angle

Further topics

- Set IMU data enable [sWN ImuDataEnable]
- Set IMU data streaming ethernet settings [sWN ImuDataEthSettings]

3.5.14 Scanning range

Overview

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.

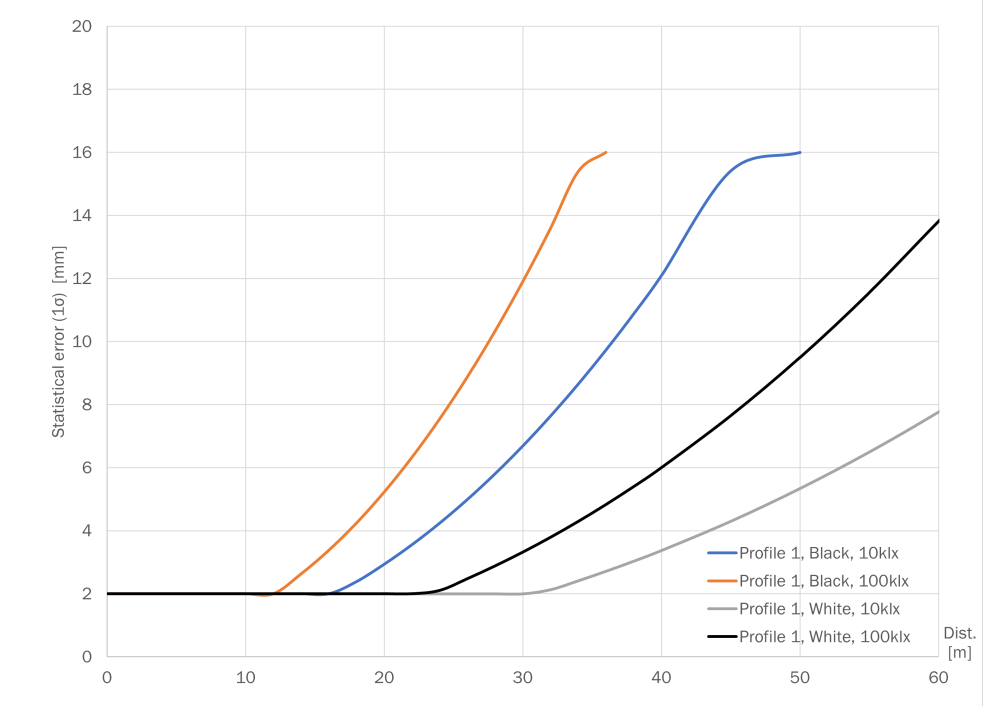


Figure 24: Profile 1 (15 Hz; 0.5 °)

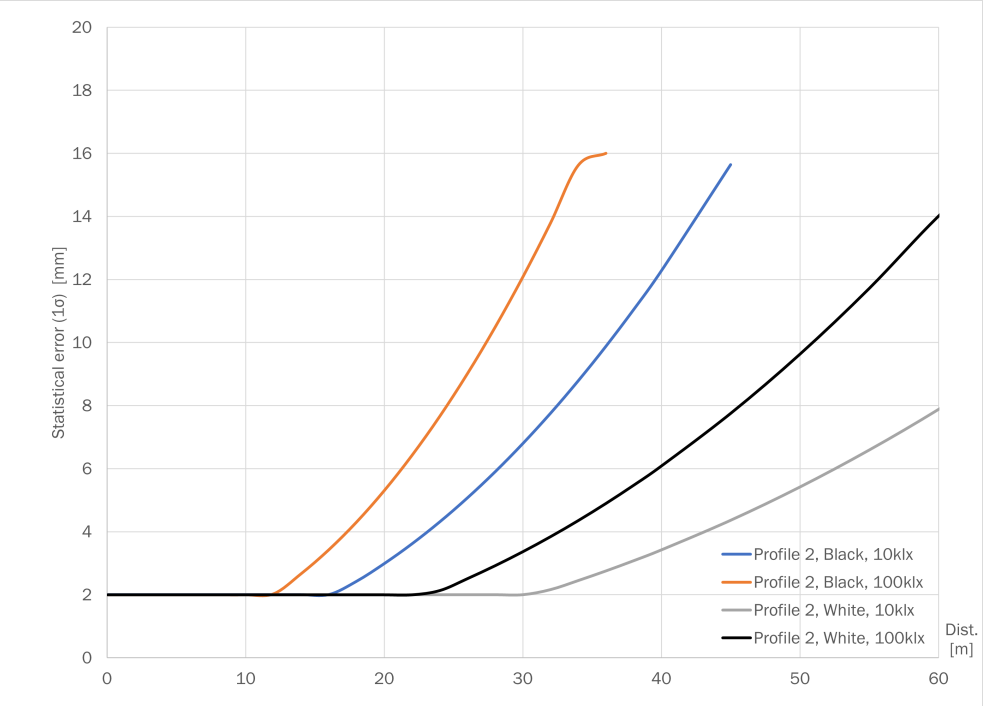


Figure 25: Profile 2 (15Hz; 0.33 °)

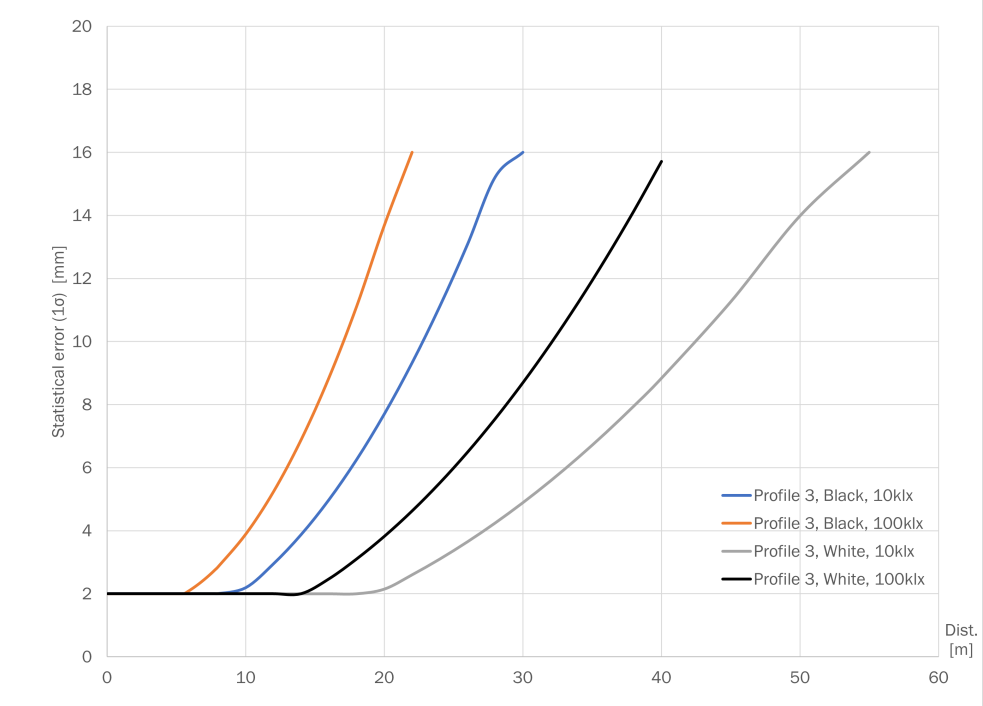


Figure 26: Profile 3 (20 Hz; 0.1°)

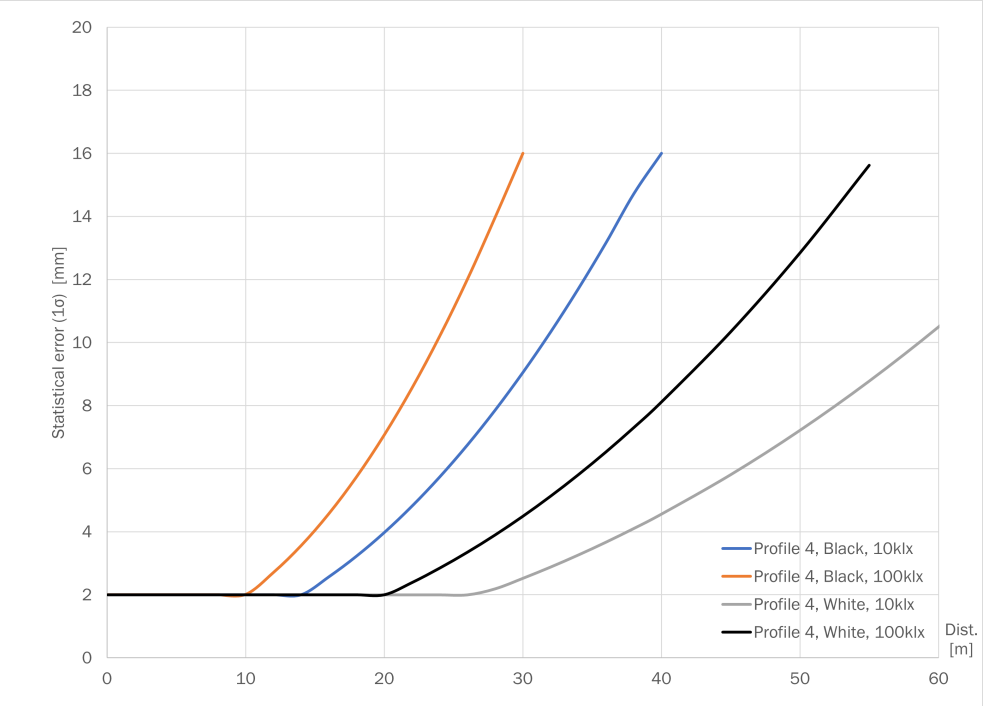


Figure 27: Profile 4 (20 Hz; 0.25°)

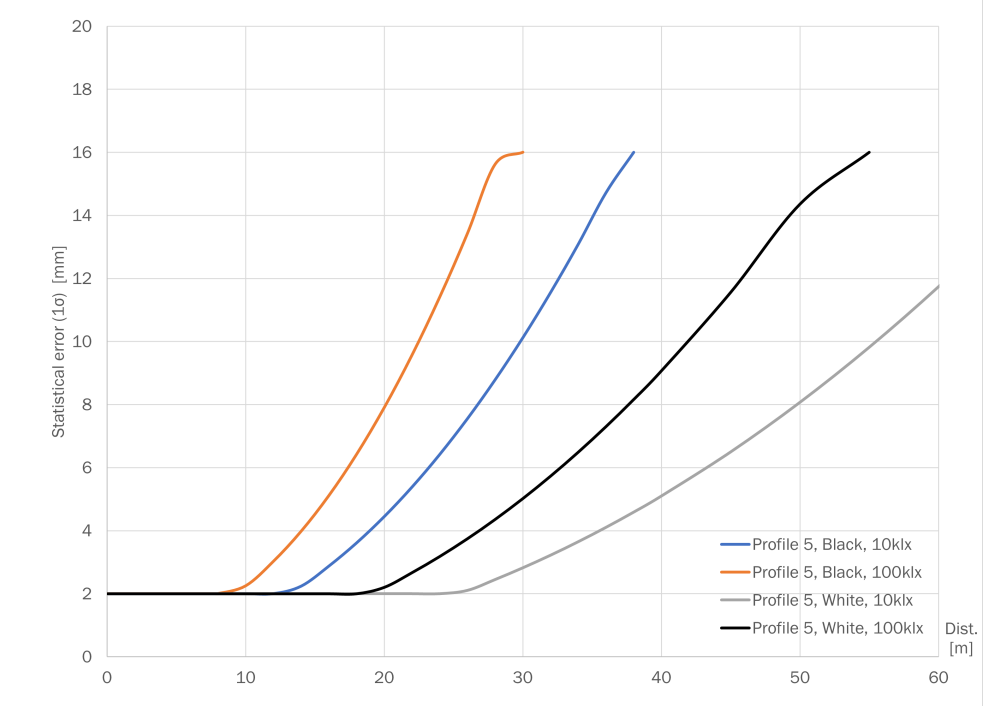


Figure 28: Profile 5 (25 Hz; 0.25°)

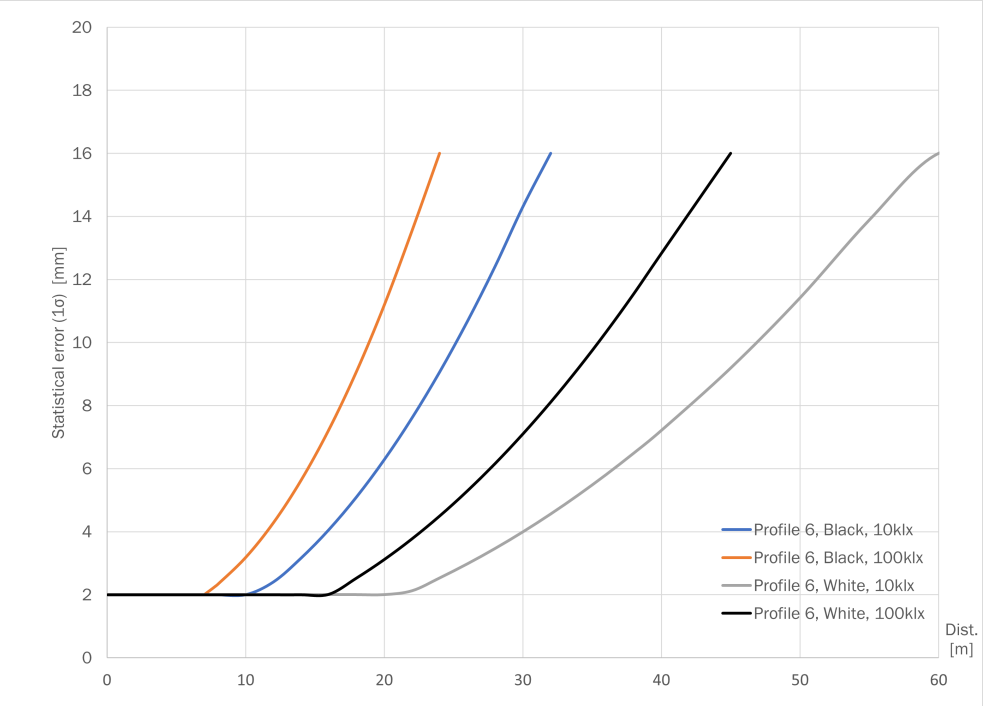


Figure 29: Profile 6 (30 Hz; 0.1°)

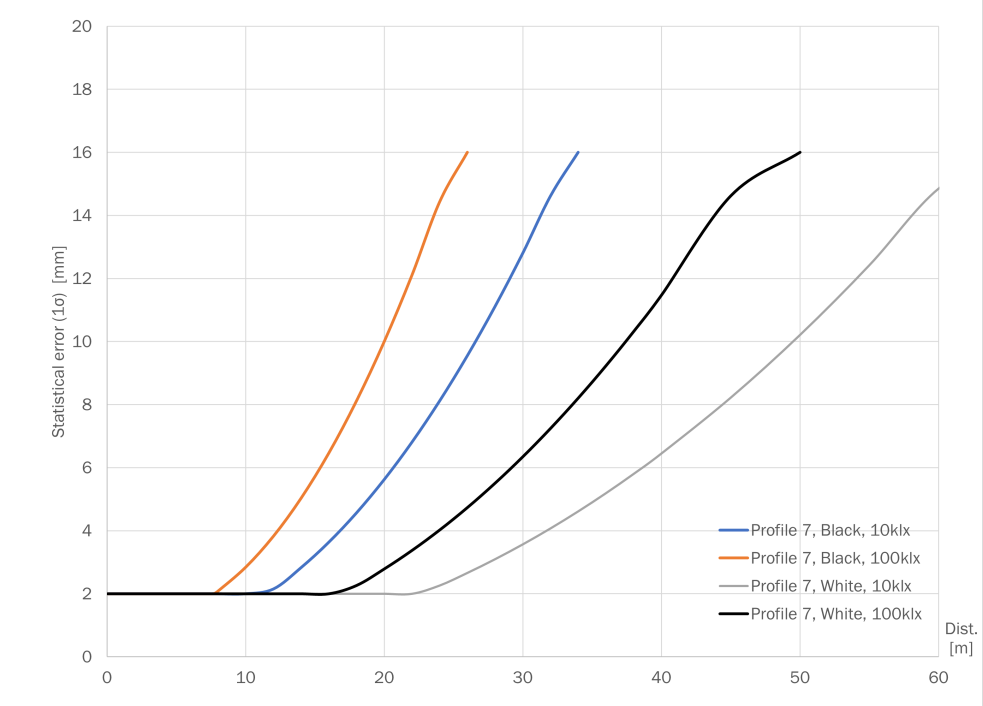


Figure 30: Profile 7 (40 Hz; 0.25°)

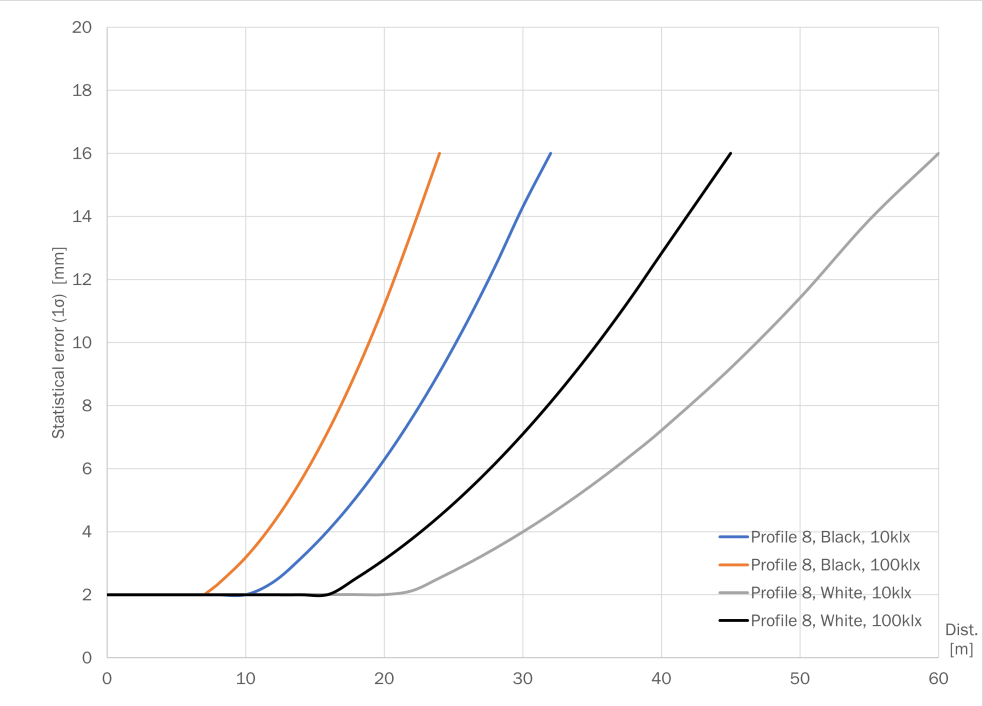


Figure 31: Profile 8 (50 Hz; 0.25°)

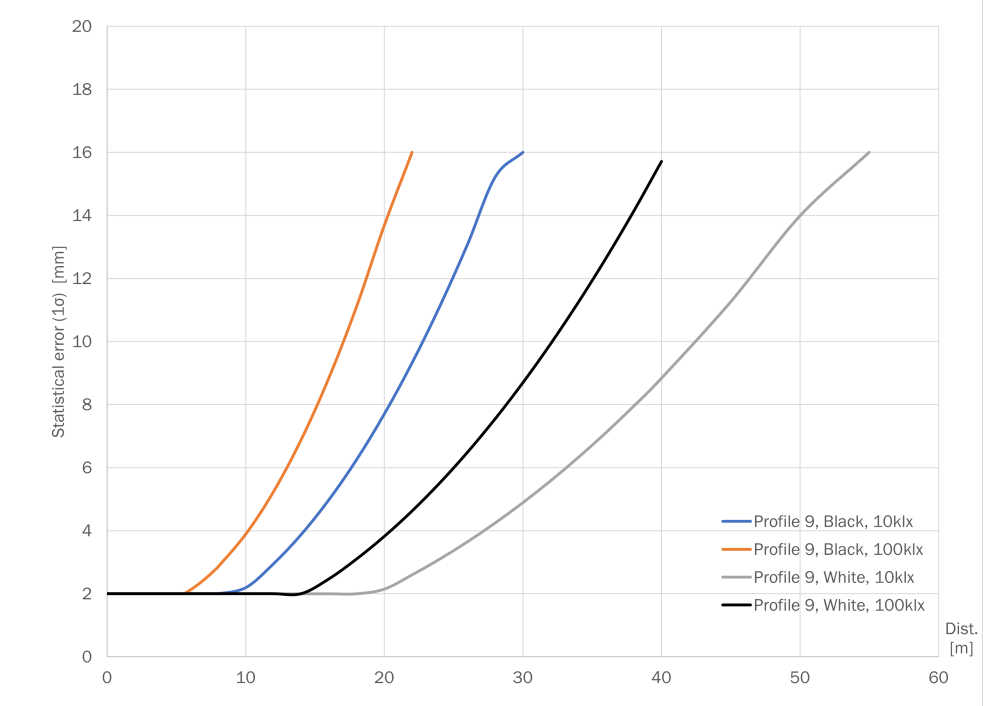


Figure 32: Profile 9 (15 Hz; 0.05°)

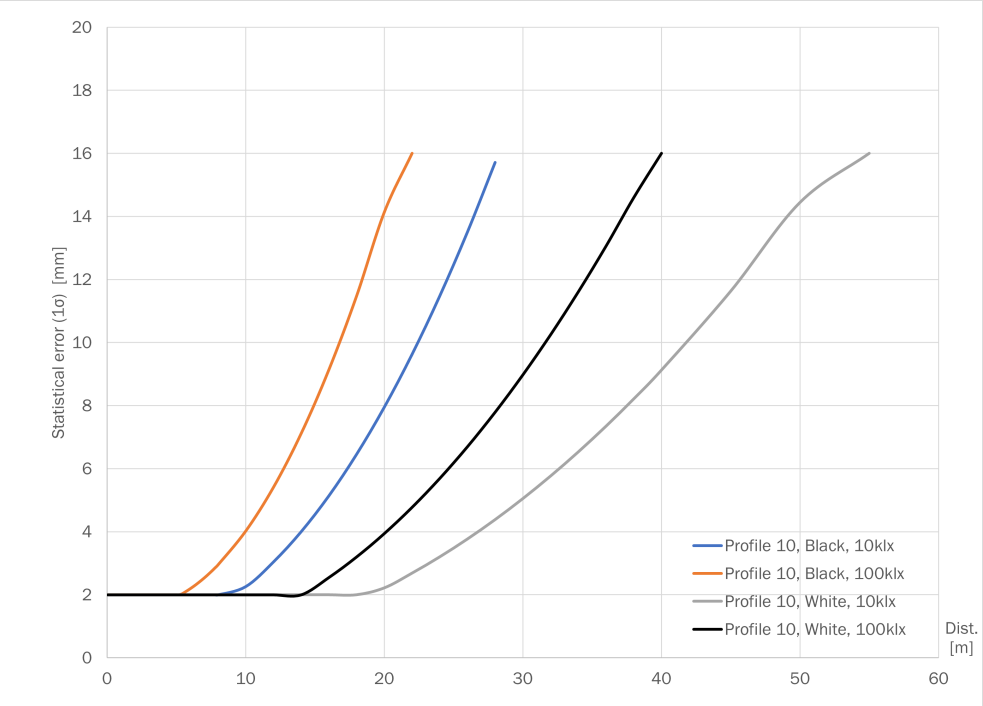


Figure 33: Profile 10 (40 Hz; 0.125°)

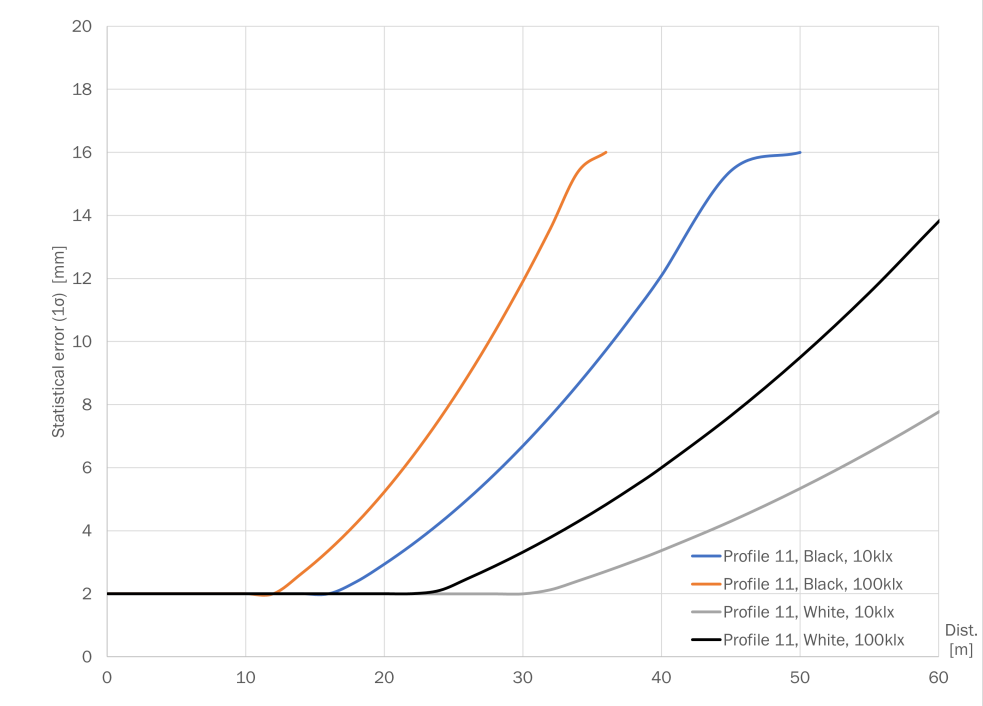


Figure 34: Profile 11 (15 Hz; 1 °)

3.5.15 Extended scanning range

The picoScan150 Pro Performance variant offers the option of switching to high reflectivity targets such as reflectors for scanning ranges of up to 120 m. This switching changes the measuring core parameters so they are optimized for large scanning ranges.

Extended scanning range should only be activated if measuring objects will actually be measured at a distance of more than 60 m. Activating it will slightly worsen the noise characteristics at close range and increase the susceptibility to mutual interference when using multiple devices. Both effects are only at a low level, but increased compared to normal operation.

For the Compact and LMDscandata data formats, the distance resolution is changed from 1 mm to 2 mm when extended scanning range is active since these data formats transmit the measurement data as a 16-bit value. When using MSGPACK, the distance resolution remains at 1 mm even with the extended scanning range activated since MSGPACK works with 32 bits.

Further topics

- Activate long range mode [sWN EnableLongRangeMode]

3.5.16 Contamination indication

The device has an optics cover to protect it. This optics cover 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 factor than they actually have and, from a certain degree 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 optics cover is not cleaned and contamination increases, then a contamination error is output.

The device supports a contamination indication over the full 276°. 7 sectors are monitored independently. Each sector can be individually enabled for the contamination indication.

You can select different settings, depending on the application in which the device is used.

Warning/error output

- **All sectors contaminated:** If all active sectors have the same or a higher value, the device status “Contamination warning” or “Contamination error” is displayed.
- **One sector contaminated:** At least one sector must have a level of “Warning” or “Error” for a contamination warning or contamination error to be displayed.
- **No output:** Warning and error device status display is deactivated. The contamination measurement for the individual sectors continues to be performed but has no effect on the device status.

Sensitivity: Low, medium, high: Threshold for triggering contamination warnings and errors. The parameter makes it possible to tailor the display to the specific requirements of the application.

Sector preset: If an accessory is used that limits the angular range used, this can be specified in the contamination indication.

Response time: This can be used to define how quickly the contamination should result in an error or warning.

Default setting: In the default setting, the contamination display is deactivated. The monitoring of individual sectors is active and can be seen for each sector in the user interface and the command interface to assist with integrating the device into the application.

Strategy and evaluation are deactivated so no device warnings or errors are triggered. All associated parameters can be permanently changed.

Important information

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.
- Sectors that are not relevant should be deactivated to ensure a higher availability.
- High illumination reduces the contamination level.
- Moisture in the form of drops creates a high contamination value.
- Contamination warnings and contamination errors are indicated on the display elements of the device
- If there are objects close to the product, a contamination indication in this segment is not possible. In this case, the relevant segment is evaluated as a blind zone.

Further topics

- Display and control elements

4 Transport and storage

4.1 Unpacking

Procedure

- ► Check the components for completeness and integrity for all parts.
- ► In the event of complaints, contact the responsible SICK subsidiary.

5 Mounting

5.1 Installation site

The following information and recommendations are to be observed when positioning and installing the product:

General

- It can be mounted in any position.
- The mounting site is suitable for the weight of the device.
- The base is considered the reference surface for the scan field plane.
- When mounting the device to the rear, it is recommended to use the SICK mounting bracket (part number 2134874).
- During mounting, make sure there is no reflective surface behind the reference target.
- Make sure that the status indicator is clearly visible.
- Do not affix any labels or stickers to the optics cover.
- To avoid inaccurate measurements when installing multiple devices: Make sure that the laser spot of one device is not in the visible range of another device.

Vibrations

- Mount the device in a shock and vibration insulated manner.
- In applications with severe vibrations or shocks caused by vibrations, jolts or abrupt changes in directions (e.g., when mounted to a manned forklift truck), mounting with vibration dampers is to be carried out. Mount the device in a freely suspended manner.
- In applications with strong vibrations, use the three lower mounting points as these offer a higher load capability than the two rear mounting points.

Temperature and humidity

- Protect the sensor from direct and indirect sunlight.
- To prevent condensation, avoid exposing the device to rapid changes in temperature.
- Protect the device from moisture, contamination, and damage.
- The ventilation element must not be sealed off during installation.
- The device must be mounted in such a way that no water can pool on the ventilation element. When using a mounting bracket, it is recommended to allow a narrow gap between the ventilation element and the mounting bracket. This is already appropriately provided for when using the SICK accessories.

Further topics

- Ventilation element
- Dimensional drawing
- Mounting multiple products

5.2 Ventilation element

Overview

The ventilation element ensures an improved pressure equalization and allows the exchange of air and heat between the housing and surroundings.

Prerequisites

- Do not affix any labels or stickers to the ventilation element.
- Do not paint over the ventilation element.
- Products that have been subjected to a long period of moisture or very rapid temperature changes need to first equilibrate after being switched on. In some circumstances, therefore, a period of time should be allowed before measurement readiness of the product because any moisture in the housing must first be taken up by the air in the housing, which is heated up through the operation of the product, so that it can then escape via the ventilation element. Depending on the nature of the precipitated moisture, this time period might be several minutes or even up to hours.

Ventilation element

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

Further topics

- Product overview
- Installation site

5.3 Mounting multiple products

Overview

The product has been designed to minimize the probability of mutual interference, including between different LiDAR sensors. For applications where it is not possible to separate the scan planes, the product exhibits an extremely reliable measurement behavior on account of the HDDM+ process.

To rule out even the slightest effects on the measurement accuracy, the products should be arranged in such a way that as few laser beams as possible are received from other products.

Important notes

NOTICE RISK OF INTERFERENCE FROM OTHER DEVICES!
Radiation sources with a wavelength of 905 nm can cause interference if they affect the device directly.

Recommended minimum distances

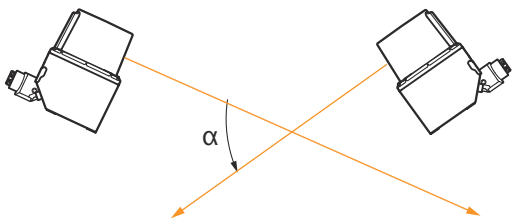


Figure 35: Angle ≥ 6 °

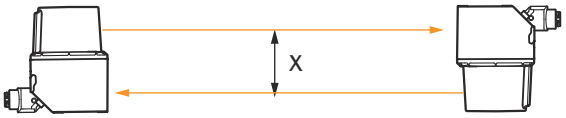


Figure 36: Distance ≥ 200 mm

5.4 Mounting the system plug on the product

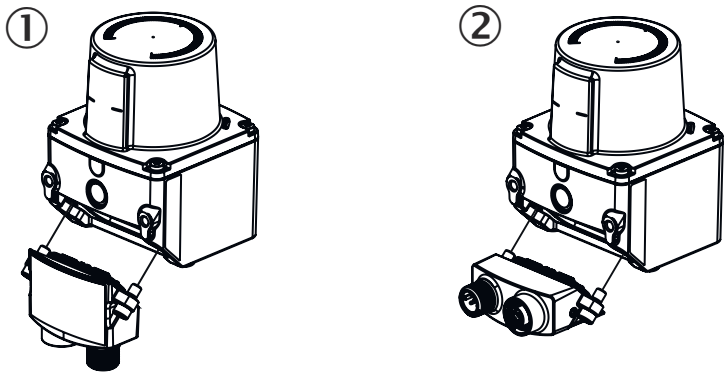
Overview

This mounting step is only required for the basic device product variant.

Prerequisites

- The system plug, the seal on the product, and the entire connection area are free of contamination and moisture and show no signs of damage.

Fitting the system plug



① System plug, installation at bottom
② System plug, installation at rear

1. Ensure the voltage supply is not connected.
2. Attach the system plug at the desired installation position on the product.
3. Tighten the screws (tightening torque: max. 2 Nm).

Further topics

- Mounting the product

5.5 Mounting the product

Important notes

NOTICE
Risk of damage to the device

the device will be damaged if the tightening torque of the mounting screws is too high or if the maximum screw-in depth of the blind hole threads is exceeded.

- ► Observe maximum tightening torque.
- ► Use suitable mounting screws for the blind hole threads of the device. Observe the maximum screw-in depth.

Prerequisites

- The connector is mounted on the product.

Procedure

1. Mount the product in a suitably prepared bracket using the fixing holes provided. Mounting brackets are available as accessories.
2. Make the electrical connection. Attach and tighten a voltage-free cable.
3. Align the vertical center line of the viewing range of the product 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.
4. Switch on the supply voltage.

✓ The status LED 2 lights up green after successful initialization. The product is ready for operation.

5. Align the vertical center line of the viewing range of the product with the center of the area to be monitored. The marking (0° axis) on the upper side of the optics cover serves as a alignment aid.

Further topics

- Mounting the system plug on the product
- Dimensional drawing
- Connecting the product electrically

6 Electrical installation

6.1 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

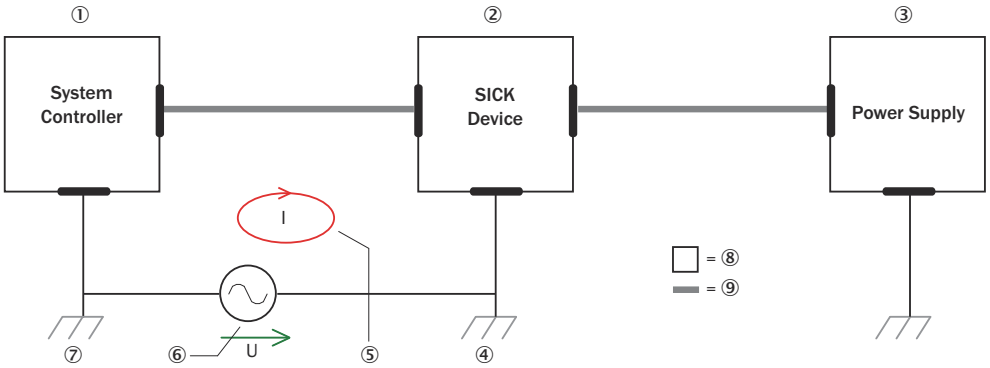


Figure 37: Example: Occurrence of equipotential bonding currents in the system configuration

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.

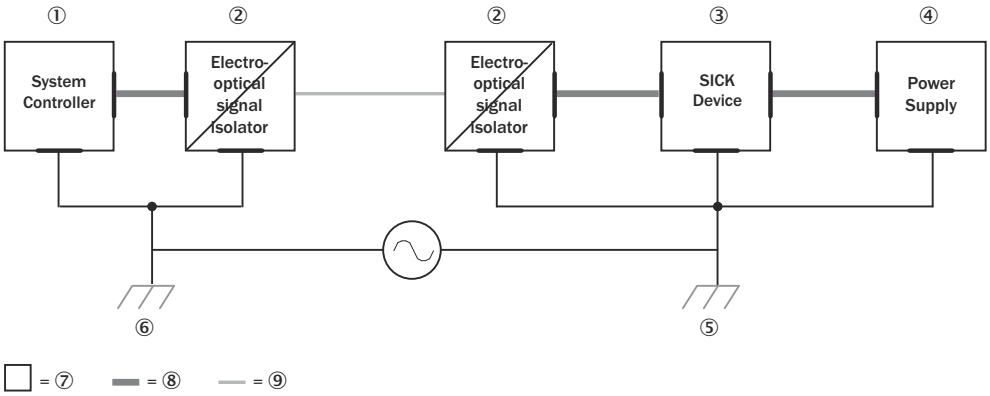


Figure 38: Example: Prevention of equipotential bonding currents in the system configuration by the use of electro-optical signal isolators

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 Optical fiber

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.

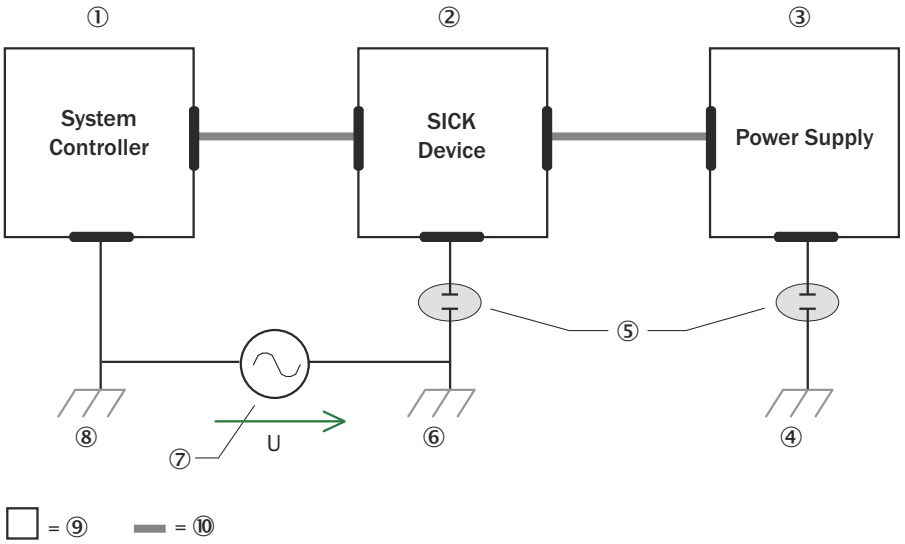


Figure 39: Example: Prevention of equipotential bonding currents in the system configuration by the insulated mounting of the device

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.

6.2 Calculation instructions for voltage drop and length of cable

Overview

The following formulas can be used to estimate the required cable lengths or supply voltages. Other conditions of the system must be considered in detail.

Formula for the voltage drop to be considered

$$\Delta V = \frac{I \bullet 2 \bullet L}{A} \bullet \rho \bullet (1 + \alpha \bullet (T - T_0))$$

Formula for permissible length of cable

$$L = \frac{\Delta V \cdot A}{2 \cdot I \cdot \rho \cdot (1 + \alpha \cdot (T - T_0))}$$

Example

Prerequisites:

- Steady state of the voltage supply
- Only applies for copper cable material

Table 3: Values used in both example calculations

Cable properties	
$A = 0.34 \cdot 10^{-6}\ m^2$	Cross-section of the cable surface [m^2]
$\rho = 1.72 \cdot 10^{-8}\ \Omega m$	Specific resistance of copper [Ωm]
$\alpha = 3.9 \cdot 10^{-3}\ K^{-1}$	Temperature coefficient of copper [1/K]
Ambient conditions	
$T_0 = 20\ °C$	Reference temperature [°C]
$T = 80\ °C$	Cable temperature [°C]
Cable load	
I = P/U = 1.46 A P = 35 W U= 24 V	Load current I [A] Supply voltage U [V] Maximum expected power consumption P

Table 4: Example: voltage drop to be considered for cable part no. 2096241

L = 10 m	Cable length [m]
$\Delta V = \frac{I \cdot 2 \cdot L}{A} \cdot \rho \cdot (1 + \alpha \cdot (T - T_0)) = 1.82\ V$	Voltage drop ΔV [V]

Table 5: Calculation of the cable length for allowed voltage drop of 1.82 V

ΔV = 1.82 V	Voltage drop on the cable [V]
$L = \frac{\Delta V \cdot A}{2 \cdot I \cdot \rho \cdot (1 + \alpha \cdot (T - T_0))} = 10\ m$	Permissible length of cable [m]

6.3 Connections and pin assignment

Overview

The connections depend on the mounted system plug.

Prerequisites

- Connect the connecting cables in a de-energized state. Do not switch on the supply voltage until installation is complete and all connecting cables are connected to the product and controller.
- Wire cross-sections in the supply cable from the user's power system must be implemented in accordance with the applicable standards.
- Use shielded cables.
- In the case of open end cables, make sure that bare wire ends do not touch. Wires must be properly insulated from each other.
- Do not lay cables over long distances in parallel with voltage supply cables and motor cables in cable ducts.

PWR/IOs connection

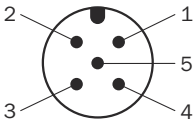


Figure 40: Male connector, M12, 5-pin, A-coded

Table 6: Pin assignment for PWR & 3 I/Os connection (part no. 2116047)

Contact	Signs	Description	Wire color, part number 2095733 [1]
1	Vs	Supply voltage: +9 ... +30 V DC	Brown
2	IN2 / OUT2	Digital input 2 / digital output 2	White
3	GND	Supply voltage: 0 V	Blue
4	IN1 / OUT1	Digital input 1 / digital output 1	Black
5	IN3 / OUT3	Digital input 3 / digital output 3	Gray

1) Data only valid when using the specified connecting cable with flying leads, which is available as an accessory

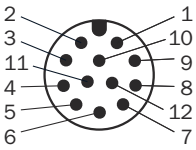


Figure 41: M12 male connector, 12-pin, A-coded

Table 7: Pin assignment for PWR & 6 I/Os connection (part no. 2130754)

Contact	Labels	Description
1	IN1/OUT1	Digital input 1 / digital output 1
2	GND	Supply voltage: 0 V
3	IN2/OUT2	Digital input 2 / digital output 2
4	–	–
5	–	–
6	IN3/OUT3	Digital input 3 / digital output 3
7	IN4/OUT4	Digital input 4 / digital output 4
8	IN6/OUT6	Digital input 6 / digital output 6
9	Vs	Supply voltage: +9 ... +30 V DC
10	IN5/OUT5	Digital input 5 / digital output 5
11	–	–
12	–	–

Table 8: Pin assignment for PWR & 6 I/Os connection (part no. 2130871)

Contact	Labels	Description
1	GND	Supply voltage: 0 V
2	Vs	Supply voltage: +9 ... +30 V DC
3	IN1/OUT1	Digital input 1 / digital output 1
4	IN2/OUT2	Digital input 2 / digital output 2
5	IN3/OUT3	Digital input 3 / digital output 3
6	IN4/OUT4	Digital input 4 / digital output 4
7	IN5/OUT5	Digital input 5 / digital output 5

Contact	Labels	Description
8	IN6/OUT6	Digital input 6 / digital output 6
9	–	–
10	–	–
11	–	–
12	–	–

Ethernet connection

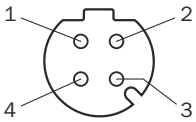


Figure 42: M12 female connector, 4-pin, D-coded

Table 9: Pin assignment for Ethernet connection

Contact	Signs	Description
1	TX+	Sender+
2	RX+	Receiver+
3	TX-	Sender-
4	RX-	Receiver-

Complementary information

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).

Further topics

- Information on interfaces: Data sheet

6.4 Voltage supply

Prerequisites

- All circuits connected to the product must be configured as SELV or PELV circuits. SELV = safety extra-low voltage, PELV = protective extra-low voltage.
- Connect the connecting cables in a de-energized state. Do not switch on the supply voltage until installation is complete and all connecting cables are connected to the product and controller.
- Protect the product with an external slow-blow fuse rated at min. 1 A/max. 3 A at the start of the supply cable.
- To ensure protection against short-circuits/overload in the customer's supply cables, choose and implement wire cross-sections in accordance with the applicable standards.

6.5 Output load and power consumption

The maximum power consumption depends on how the product is employed. If no outputs are used, no additional power consumption by them needs to be considered. The typical power consumption is then as follows: P_{typ} = 4.5 W (at 24 V supply voltage, 25 °C ambient temperature, continuous scan data output, no output load)

If outputs are used, the power consumption increases according to the external sinks. This can be up to 6 configurable I/Os, depending on the system plug used.

Number of outputs used	Output current per output	P_{output}
2	200 mA	P_{output} = 2 x 30 V x 200 mA = 12 W
3	133 mA	P_{output} = 3 x 30 V x 133 mA = 12 W
4	100 mA	P_{output} = 4 x 30 V x 100 mA = 12 W
6	67 mA	P_{output} = 6 x 30 V x 67 mA = 12 W

The maximum power consumption is the sum of the typical power consumption and the power consumption of loaded outputs: $P_{max} = P_{typ} + P_{output} + P_{temp\_res}$

For the example with 6 loaded outputs, this gives:

- P_{typ} = 4.5 W
- P_{output} = 12 W
- P_{temp_res} = 0.5 W
- P_{max} = 4.5 W + 12 W + 0.5 W = 17 W

6.6 Connecting the product electrically

Prerequisites

- Ensure that the power supply unit can provide the necessary voltage and current for operating the product. Particular attention must be given to the voltage drop across the supply cable, and for digital outputs the additional voltage drop in the opposite direction and the required start-up power, without which the product cannot start reliably.

Procedure

1. Ensure the voltage supply is not connected.
2. Connect the product according to the connection example.
3. Any unused electrical M12 connections of the system plug are closed with a protective element of the corresponding type (as in the delivery state).

Further topics

- Calculation instructions for voltage drop and length of cable
- Data sheet

7 Commissioning

7.1 Starting SOPASair

Overview

SOPASair is used for operation, parameterization, and servicing purposes (e.g., diagnostics, data logger).

Prerequisites

- Computer with Ethernet port
- Device is connected to the computer via Ethernet.
- Voltage supply is connected.
- Computer and device are located on the same network.
- Computer and device have different IP addresses.

Procedure

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

✓ The user interface is displayed.

3. To make changes, log in to the device.

7.1.1 Logging in to the device

Overview

The device has different user levels. Only the **Service** user level is activated in the delivered state.

The user levels have different authorizations for configuring the device.

The current user level is displayed in the **Log in** panel.

Procedure

Activate the user level during initial commissioning of the device

1. Click on the button.

✓ The **Log in to device** input screen opens. The **Service** user level is selected.

2. Enter the password servicelevel and click on **Log in**.
3. Activate the desired user level.

Logging in to the device

1. User levels have been activated.
2. Click on the button.

✓ The **Log in to device** input screen opens.

3. Select user level(**User level**), enter a password(**Password**) then click **Login** in.

User levels	Password
Maintenance (deactivated in the delivery state)	`main`
Authorized Client (deactivated in the delivery state)	`client`
Service	`servicelevel`

Important information

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

7.1.2 Password management

7.1.2.1 Changing password

Procedure

1. Establish a connection to the product in the web browser.
2. Select:
3. Log in with the last assigned password.
4. Select **Change password**.
5. Assign a new password.

✓ The new password is valid immediately.

7.1.2.2 Resetting the password

Procedure

1. Establish a connection to the product in the web browser.
2. Select:
3. Select **Password forgotten?**.
4. Send **Device key**, **Serial number** and **Part number** in an e-mail to the responsible SICK sales company or the responsible SICK service partner, see www.sick.com/worldwide.
5. Click **Next** to confirm.

✓ The window for entering a code appears.
You can get the code from the responsible SICK sales company or the responsible SICK service partner. The code is only valid once for the reset process. You can close the window by clicking on the **x** without interrupting the reset process. If you select **Cancel** or enter an incorrect code several times, the current reset process is terminated. The requested code is no longer valid. The process must be restarted.

6. Optional: close the window by clicking on **x**. At a later time, open the window for entering the code via and **Password forgotten?**.
7. Enter the code.
8. Click **Reset** to confirm.

✓ The password is reset to the default password **servicelevel**. Parameters are not changed.

7.1.3 Data backup

Manual data backup using project file

The parameter set can be manually saved on the computer as a project file (*.sopas). This is the generally recommended procedure. Using the project file, the parameter set can be transferred to a replacement product via download.

Parameter cloning

As the existing system plug continues to be in use, the product make it possible for the last parameter values used in the system plug to be automatically passed on to a product of the same type (cloning). Accordingly, the process overwrites all the existing parameter values in the product.

During cloning, the replacement product accepts the following configurable parameters:

- Addressing mode
- IP address
- Subnet mask
- Default gateway
- Device name

The following parameters are not included when cloning:

- Parameter settings such as filter, field geometries
- Device-specific parameters:
  - Serial number
  - MAC address
  - Licenses
  - Operating hours counter
  - Error memory

After being switched on, the product automatically checks the system plug memory. The subsequent product behavior depends on the content of the system plug memory. The goal is for the internal parameter set and the parameter set saved on the system plug to always be identical.

Content of the system plug memory	Device behavior
Empty	Once the parameter set is permanently saved on the product, the product also saves the internal parameter set on the system plug. The prerequisite is that the **Save permanently** function has been activated via SOPASair or SOPAS ET.
No parameter set possible to interpret	
Parameter set possible to interpret	After being switched on, the product automatically loads the compatible parameter set from the system plug into the temporary parameter memory of the product. The product then starts with its new valid parameter set. To permanently transfer the parameter set to the product, activate the **Save permanently** function via SOPASair or SOPAS ET.

Further topics

- Saving the parameter set

7.1.4 Saving the parameter set

Overview

The product is configured for the application using SOPASair. The parameter set can be permanently saved in SOPASair. To be able, for example in the event of a product failure, to restore the parameter set to a replacement product or to also save the parameter set in the cloning parameter memory of the system plug.

Procedure

1. Permanently saving the parameters in SOPASair:

✓ The product stores the parameter set internally in the permanent parameter memory and in the cloning parameter memory of the system plug.

Further topics

- Data backup

7.2 Starting SOPAS ET

Overview

The saved parameters can be manually saved, imported, and exported as a project file on the computer in the SOPAS ET configuration software.

Prerequisites

- Computer with the SOPAS ET software installed
  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.
- Ethernet connection
- SDD file (device description file)
  You can install the SDD file using the device catalog in SOPAS ET. Use the wizard in SOPAS ET to do this. The SDD file can be installed from the device or the SICK website. To install it from the SICK website, you need an Internet connection.
- Access data

User levels	Password
Maintenance (deactivated in the delivery state)	`main`
Authorized Client (deactivated in the delivery state)	`client`
Service	`servicelevel`

- Device is ready for use.

Procedure

1. Install the latest version of the SOPAS ET configuration software. In this case, select the “Complete” option as suggested by the installation wizard. Administrator rights may be required on the computer to perform the installation.
2. Start “SOPAS ET” after completing the installation.

✓ SOPAS ET automatically starts the search for connected devices. Connected devices are displayed in the **Device Search** window.

3. Select the desired device in the list of available devices.
   The following IP addresses are configured by default on the device:
   - IP address: 192.168.0.1
   - Subnet mask: 255.255.255.0

   If necessary, install an updated device description file for the device.
4. Click on **Add** to establish communication.

✓ SOPAS ET establishes communication with the device, loads its current device description (parameters), and displays the device in the **New Project** window.

5. Log into the device.

7.3 Installing firmware updates

Prerequisites

- Computer with installed software or SICK AppManager
  The current version of SICK AppManager can be downloaded from www.sick.com/SICK_AppManager. To install SICK AppManager, open the installation file (*.exe) and follow the instructions on the screen.

Procedure

1. Open SICK AppManager.
2. Drag and drop the file into the **Firmware** window.
3. In the **Firmware** window, select the file to be installed.

4. In the bottom right window, click the **Install** button.
✓ The firmware update is installed.

8 Maintenance

8.1 Maintenance

Table 10: Maintenance schedule

Maintenance work	Interval
Check device and connecting cables for damage at regular intervals.	Depends on ambient conditions and climate.
Check housing and optics cover for contamination and clean if necessary, .	Depends on ambient conditions and climate.
Check the screw connections and plug connectors.	Depends on the place of use, ambient conditions or operating requirements. Recommended: At least every 6 months.
Check the assembly accessories used.	Depends on the place of use, ambient conditions or operating requirements. Recommended: At least every 6 months.
Check that all unused connections are sealed with protective caps.	Depends on ambient conditions and climate. Recommended: At least every 6 months.

8.2 Cleaning the product

Important information

NOTICE

- Never use sharp objects for cleaning.
- Recommendation: Use anti-static cleaning agents.
- Recommendation: Use anti-static plastic cleaners and lens cloths from SICK.

NOTICE

If the optics cover is scratched or damaged, take the product out of operation and have it repaired by SICK.

Procedure

1. If possible, switch off the voltage supply to the product.
2. Use only a clean, damp, lint-free cloth and a mild anti-static lens cleaning fluid to clean the optics cover.
3. Remove any dust from the housing using a soft brush.

9 Troubleshooting

9.1 Troubleshooting

Faults, warnings and errors

Table 11: Troubleshooting questions and replies

Question / status	Response / remedial actions
Both LEDs flash red.	Device error: Read out the error via the SOPASair or SOPAS ET PC software and remedy the cause of the error.
LEDs indicate an undefined status.	Check the device status, if necessary contact the SICK Service department.
All LEDs are off	Check the voltage supply to the device. In SOPASair, check whether the LEDs were switched off.
All LEDs of the device light up red at startup and do not change to green.	Check the voltage supply to the device. The power supply unit may not be supplying the required current or the required voltage to start the device.
All LEDs of the device flash red.	The device may not be able to recognize the system plug. Check that the system plug is mounted correctly and that both contact sides are clean and dry.
Measurement data show anomalies.	Optics cover contaminated: Clean the optics cover.
When accessing the device via a web browser, the SOPASair user interface is not loaded, the SOPASair loading screen is permanently displayed.	Try connecting again. If this does not work: Restart the device. Check the network settings. Standard IP address: 192.168.0.1 The IP address range 169.254.1.{1-15} is locked for device-internal communication. The device cannot be assigned any of these 15 IP addresses.
SOPASair is not started in the browser.	Check the IP address of device and network adapter (e.g., using device search in SOPAS ET) and adjust if necessary. Then try to establish the connection again.

Diagnostics using SOPASair

The error log contains current error messages.

You can create and download a diagnostic file in the configuration software for service purposes. The diagnostic file contains data required for error analysis by SICK Service. The diagnostic file is encrypted and can only be opened by SICK Service.

Complementary information

For faults that cannot be rectified based on the error description, please contact SICK Service. To help us to resolve the matter quickly, please note down the details on the type label.

9.2 Repair

Repairs on the device may only be performed by qualified and authorized personnel from SICK AG. Interference with or modifications to the device on the part of the customer will invalidate any warranty claims against SICK AG.

10 Decommissioning

10.1 Removing the product

Procedure

1. Switch off the supply voltage.
2. Mark the position and alignment of the device on the bracket or surroundings.
3. Disconnect and remove the connecting cables of the device.
4. Remove the device from the bracket.

Complementary information

If you replace the product, you can transfer the parameter values to the replacement product by downloading them.

10.2 Disposal of the product

Procedure

► Always dispose of unusable products in accordance with national waste disposal regulations.

Complementary information

SICK will be glad to help you dispose of these products on request.

11 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.

11.1 Data sheet

Features

Measurement principle	HDDM+ statistical measurement procedure
Application	Indoor and outdoor
Light source	Infrared (wavelength 905 nm, maximum output power < 5 W, maximum output power per area 0.2 W/mm², pulse duration 1.2 ns, average power < 10 mW)
Laser class	Laser class 1 (IEC 60825-1:2014, EN 60825-1:2014+A11:2021) 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	276°
Scanning frequency [1]	Core: 15 Hz, 25 Hz Prime: 15 Hz, 20 Hz, 40 Hz [2] Pro: 15 Hz ... 50 Hz
Angular resolution [1]	Core: 0.33°; 0.25°; 1° Prime: 0.10°; 0.25°; 1° [2] Pro: 0.05° ... 1°
Working range	Core: 0.05 m ... 25 m Prime: 0.05 m ... 60 m Pro: 0.05 m ... 120 m
Blind zone	0 m ... 0.05 m
Spot size	8 mm (at the optics cover) In 5 m: 32 mm In 10 m: 55 mm In 25 m: 126 mm Prime, Pro: In 50 m: 244 mm
Laser divergence	typ. 0.27 [°] / 4.8 [mrad] max. 0.3 [°] / 5.3 [mrad]
Number of echoes evaluated	3 (output options: first echo, last echo or all echoes)
Minimum distance between echoes	0.5 m
Scanning range at 10 % remission, > 99 % detection probability and 10 klx	Core: typ. 12 m Prime: typ. 25 m Pro: typ. 40 m For more details, see "Scanning range and working range diagrams", page 65

Scanning range at 90 % remission, > 99 % detection probability and 10 klx	Core: typ. 25 m Prime: typ. 47 m Pro: typ. 75 m For more details, see "Scanning range and working range diagrams", page 65
Reflector detection working range [3]	Core: typ. 0.3 m ... 25 m Prime: typ. 0.3 m ... 60 m Pro: typ. 0.3 m ... 120 m
Scan field flatness	±1°

1) Depends on the Dynamic Sensing Profile
2) Extendable with Dynamic Sensing Profile
3) On a sufficiently large Diamond Grade reflector (part no. 5320565) and orthogonal angle of incidence

Mechanics and electronics

Connection type	Depending on the mounted system plug, 2 x M12 round connectors
Supply voltage (V_S)	9 V DC ... 30 V DC, reverse polarity protected
Permissible residual ripple	± 5%
Power consumption	Typ. 4.5 W max. 17 W with loaded digital outputs, depending on the mounted system plug For more details, see "Output load and power consumption", page 51
Digital inputs	**Voltage range:** • low: -3 V ... 0.45 x V_S • high: 0.72 x V_S ... V_S **Switching frequency range:** • ≤100 Hz
Digital outputs	**Output mode (configurable):** • Push/pull • NPN • PNP **Voltage range:** • low: 0 V... 1 V • high: (V_S - 1 V) ... V_S **Output current per output, short-circuit protected:** • 2x max. 200 mA • 4x max. 100 mA • > 4x 50 mA
Material	Housing: Aluminum with Suretec650 coating Optics cover: Polycarbonate, scratch-resistant coating System plug: passivated zinc, black lacquered
Housing color	Anthracite gray (RAL 7016)
Enclosure rating [1]	IP65, IP67 (IEC 60529:1989+AMD1:1999+AMD2:2013) **Test conditions:** • Water spray volume: 14 l/min ... 16 l/min • Water pressure/temperature: 10000 KPa (100 bar) / 80 °C • Flat jet nozzle distance: 100 mm ... 150 mm • Spray angle: 0°, 30°, 60°, 90° • Cycle: 30 seconds per position • Rotational speed of test specimen: 5 rpm
Protection class	III (IEC 61140:2016-11)

Electrical safety	IEC 61010-1:2010-06+AMD1:2016
Weight	220 g, without system plug
Dimensions (L x W x H)	60 mm x 60 mm x 82 mm
Usage period	Typ. 12 years [2]
MTBF	> 100 years
$MTTF_D$	> 100 years, at 25 °C ambient temperature (EN ISO 13849-1:2015)

1) **Prerequisites:**
- The system plug is mounted.
- The cables plugged into the electrical connections must be screwed tight.
- Unused electrical connections are sealed off with a protective cap.

2) Based on IEC/EN 61508-2 and continuous usage at 25 °C

Performance

Data output per scan segment	Core, Prime, Pro: 30° at a scanning frequency ≤ 25 Hz Prime, Pro: 60° at a scanning frequency ≥ 30 Hz
Scan/frame rate [1]	Up to 265808 measuring point(s)
Latency of the measurement data output [2]	• Core, Prime, Pro: Segment size 30° at < 25 Hz: ≤ 10 ms (3 σ) • Prime, Pro: Segment size 60° at ≥ 30 Hz: ≤ 15 ms (3 σ) For more details, see "Latency of the measurement data output", page 25
Profile switching delay time	With equal motor frequencies: < 1 s With different motor frequencies: < 3 s
Delay time of field evaluation	
Power-up time	typ. 9.5 s
Systematic error	typ. ± 20 mm [3] max. ± 30 mm Temperature drift: Typically ± 0.5 mm/K
Statistical error (1 σ) [3]	≤ 2 mm (0.05 m ... 5 m) ≤ 16 mm (5 m ... 60 m) For more details, see "Scanning range", page 32
Zero point error (0° beam)	typ. ±0.3°
Statistical angle error	≤ 0.025° (1 σ)
Integrated application	Measurement data output (based on ordered configuration), field evaluation (based on ordered configuration)
Filter	Fog filter, echo filter, particle filter, moving average filter

1) Depending on the selected Dynamic Sensing Profile and the number of echoes.
2) Typical value; real value depends on ambient conditions and the selected Dynamic Sensing Profile.
3) 10 and 100 klx

Interfaces

Ethernet	✓ **Functions:** • Measurement data output (distance, RSSI, time stamp, reflector mark) • Field status information • DHCP • NTP • PTP • OPC DA **Protocols and data formats:** • UDP/IP Singlecast (Compact, MSGPACK) • UDP/IP Multicast (Compact, MSGPACK) • TCP/ IP (LMDscandata) • Native ROS2 (Data Distribution Service, DDS) **Data transmission rate:** • 10/100 Mbit/s half/full-duplex
Digital inputs/outputs	I/O (6 multiports) depending on the mounted system plug
Inertial measuring unit (IMU)	BOSCH BHI160B with 3-axis gyroscope and 3-axis accelerometer Sampling rate: 100 Hz **Relative position of the IMU to the optical origin:** • X: +12.4 mm • Y: +18.5 mm • Z: -48.4 mm
Optical displays	2 LEDs
Configuration software and interfaces	SOPASair (web server), SOPAS ET (software), REST API, CoLa A/B
Driver	ROS1, ROS2, C++, Python

Ambient data

Remission factor	1.8% ... > 1,000% (reflector)
Electromagnetic compatibility (EMC)	**Radiation emitted:** • Industrial area (IEC 61000-6-4:2018 / EN IEC 61000-6-4:2019 IEC 61000-6-4:2006+A1:2010 / EN 61000-6-4:2007+A1:2011) • Business and commercial areas as well as small enterprises (IEC 61000-6-8:2020 / EN IEC 61000-6-8:2020) **Electromagnetic immunity:** • Industrial environment (IEC 61000-6-2:2016 / EN IEC 61000-6-2:2019 / IEC 61000-6-2:2005 / EN 61000-6-2:2005 / EN 61000-6-2:2005 / AC:2005)
Vibration resistance [1]	Sine resonance scan: 10 Hz ... 1,000 Hz; 1 g (IEC 60068-2-6:2007-12) Sine test: 10 Hz ... 500 Hz, 10 g, 10 cycles (IEC 60068-2-6:2007-12) Noise test: 10 Hz ... 500 Hz, 13.5 g RMS, 5 h (IEC 60068-2-64:2008)
Shock resistance [1]	Single shock according to IEC 60068-2-27:2008-02: 100 g, 6 ms, 3 shocks per axis Continuous shock according to IEC 60068-2-27:2008-02: 40 g, 6 ms, 4,000 shocks per axis Continuous shock according to IEC 60068-2-27:2008-02: 50 g, 3 ms, 5,000 shocks per axis

Impact resistance	IEC 60068-2-75: 2014, Test Ehc (0.5 J, vertical strike) IEC 61010-1: 2017, based on IK08 (5 J, vertical strike)
Ambient operating temperature	-33 °C ... +50 °C
Storage temperature	- 40 °C ... + 70 °C
switch-on temperature	-33 °C ... +50 °C (permissible switch-on temperature) -5 °C ... +50 °C (immediately ready for operation)
temperature change	–33 °C ... +50 °C, 10 cycles (EN 60068-2-14:2009)
damp heat	+25 °C ... +55 °C, 95 % RH, 6 cycles (EN 60068-2-30:2005)
Permissible relative humidity	Operation: max. 80% RH, non-condensing (EN 60068-2-30:2005) Storage: max. 90% RG, non-condensing (EN 60068-2-30:2005)
Altitude	< 5,000 m above sea level
Ambient light immunity	100 klx (indirect)

1) Short restriction in measurement data availability possible

11.1.1 Scanning range and working range diagrams

Table 12: Scanning range of picoScan150 Core

Dynamic Sensing Profile	Minimum		Typical			
	100 klx		10 klx		100 klx	
	10 %	90 %	10 %	90 %	10 %	90 %
15 Hz and 0.33°	10 m	25 m	12 m	25 m	10 m	25 m
15 Hz and 1°	10 m	25 m	12 m	25 m	10 m	25 m
25 Hz and 0.25°	10 m	25 m	12 m	25 m	10 m	25 m

Table 13: Scanning range of picoScan150 Prime

Dynamic Sensing Profile	Minimum		Typical				On highly reflective targets and reflectors 1)
	100 klx		10 klx		100 klx		
	10 %	90 %	10 %	90 %	10 %	90 %	
15 Hz and 0.33°	–	–	34 m	51 m	23 m	44 m	60 m
15 Hz and 0.5°	–	–	34 m	51 m	23 m	44 m	60 m
15 Hz and 1°	–	–	34 m	51 m	23 m	44 m	60 m
20 Hz and 0.1°	14 m	27 m	23 m	38 m	16 m	31 m	45 m
20 Hz and 0.25°	–	–	29 m	51 m	20 m	38 m	60 m
25 Hz and 0.25°	17 m	33 m	28 m	51 m	19 m	36 m	60 m
30 Hz and 0.1°	–	–	21 m	26 m	15 m	26 m	30 m
40 Hz and 0.25°	15 m	29 m	25 m	47 m	17 m	32 m	60 m
50 Hz and 0.25°	–	–	23 m	44 m	16 m	31 m	55 m
15 Hz and 0.05°	–	–	21 m	21 m	15 m	21 m	25 m
40 Hz and 0.125°	–	–	21 m	21 m	14 m	26 m	30 m

1) On a sufficiently large Diamond Grade reflector (part no. 5320565) and orthogonal angle of incidence

Table 14: Scanning range of picoScan150 Pro

Dynamic Sensing Profile	Minimum		Typical				On highly reflective targets and reflectors [1]	
	100 klx		10 klx		100 klx			
	10 %	90 %	10 %	90 %	10 %	90 %	without extended scanning range	with extended scanning range
15 Hz and 0.33°	–	–	40 m	75 m	27 m	52 m	60 m	120 m
15 Hz and 0.5°	–	–	40 m	75 m	27 m	52 m	60 m	120 m
15 Hz and 1°	–	–	40 m	75 m	27 m	52 m	60 m	120 m
20 Hz and 0.1°	17 m	32 m	27 m	45 m	19 m	36 m	45 m	–
20 Hz and 0.25°	–	–	34 m	65 m	24 m	45 m	60 m	120 m
25 Hz and 0.25°	20 m	39 m	33 m	62 m	22 m	43 m	60 m	110 m
30 Hz and 0.1°	–	–	25 m	30 m	17 m	30 m	30 m	–
40 Hz and 0.25°	18 m	34 m	29 m	55 m	20 m	38 m	60 m	70 m
50 Hz and 0.25°	–	–	27 m	52 m	19 m	36 m	55 m	–
15 Hz and 0.05°	–	–	25 m	25 m	17 m	25 m	25 m	–
40 Hz and 0.125°	–	–	24 m	25 m	17 m	30 m	30 m	–

[1] On a sufficiently large Diamond Grade reflector (part no. 5320565) and orthogonal angle of incidence

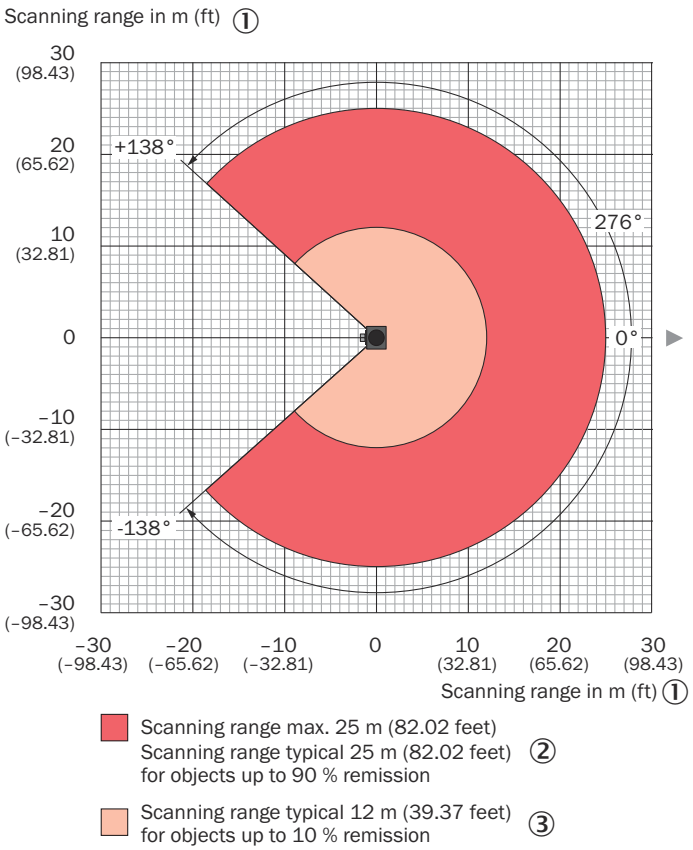


Figure 43: Working range diagram for picoScan150 Core

① Scanning range in m

② Max. scanning range 25 m/scanning range typically 25 m for objects with 90 % remission

③ Scanning range typically 12 m for objects with 10 % remission

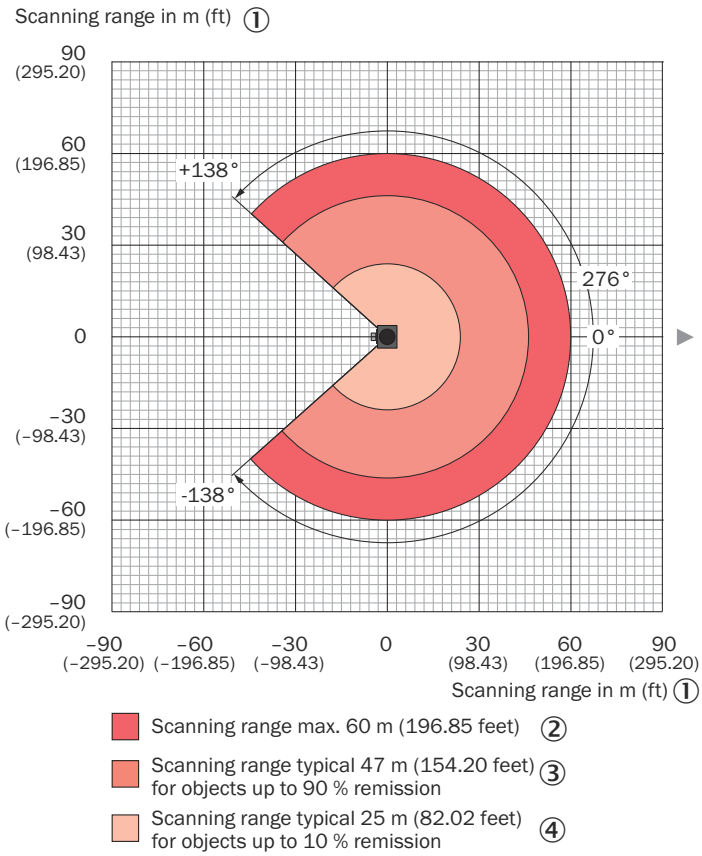


Figure 44: Working range diagram for picoScan150 Prime

① Scanning range in m

② Max. scanning range 60 m

③ Scanning range typically 47 m for objects with 90 % remission

④ Scanning range typically 25 m for objects with 10 % remission

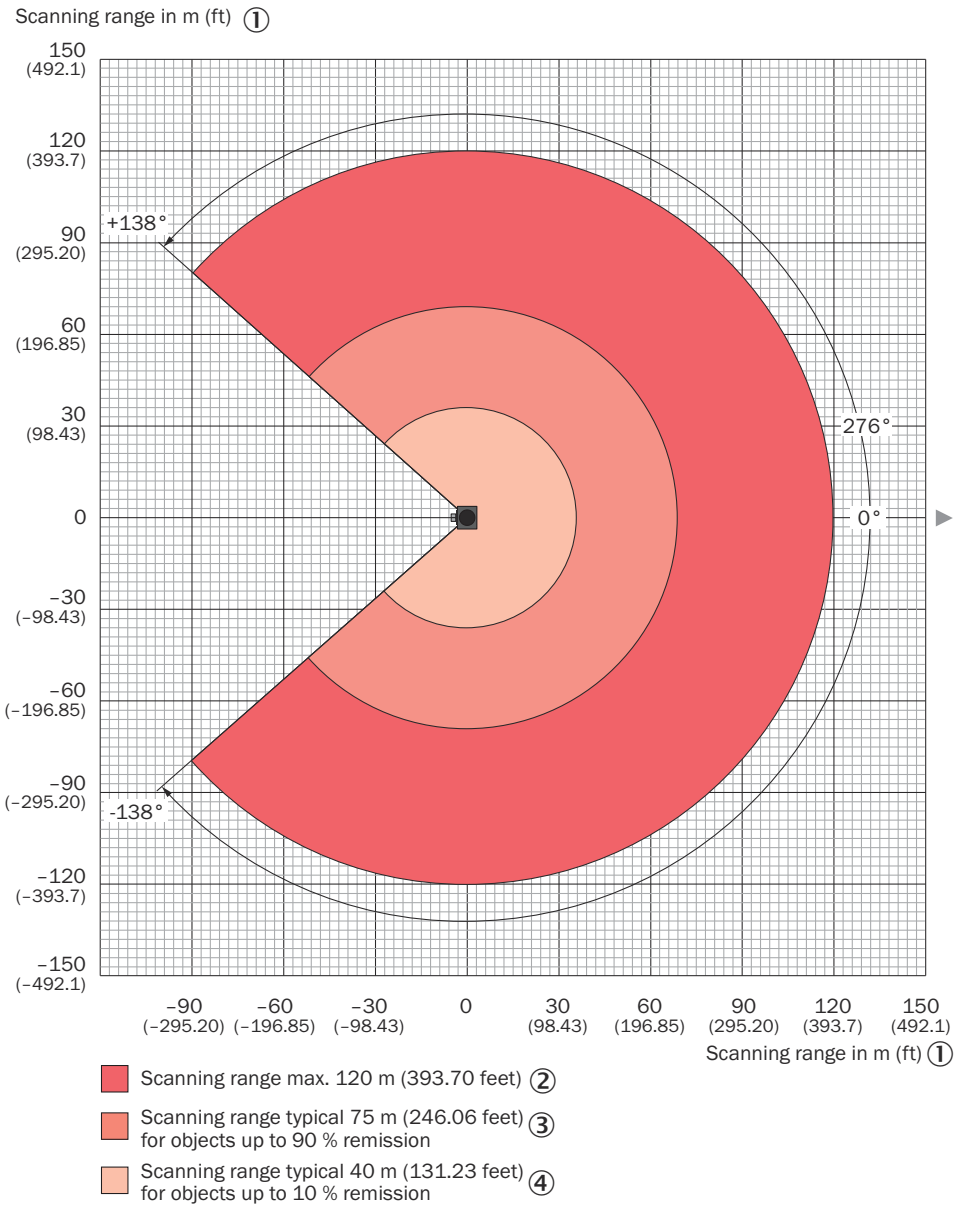


Figure 45: Working range diagram for picoScan Pro

① Scanning range in m

② Max. scanning range 120 m

③ Scanning range typically 75 m for objects with 90 % remission

④ Scanning range typically 40 m for objects with 10 % remission

11.2 Dimensional drawing

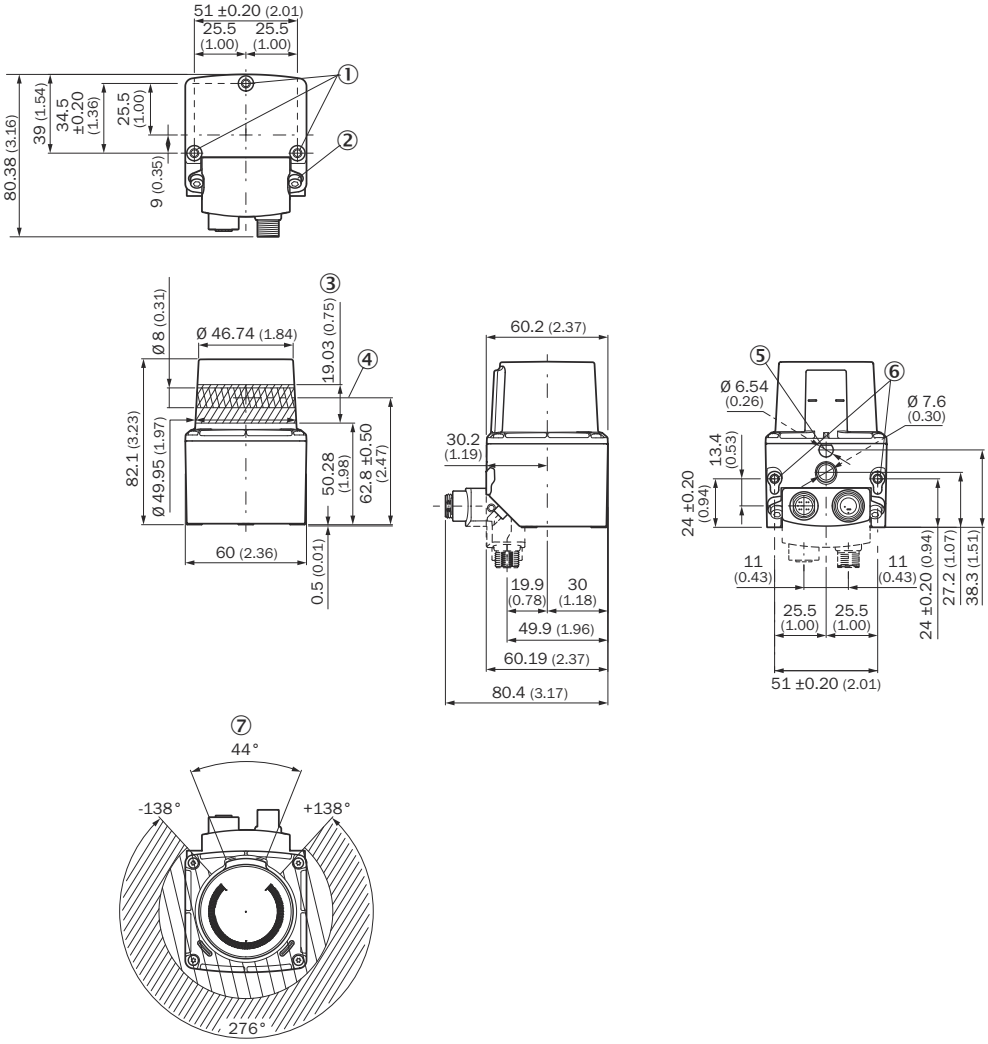


Figure 46: Dimensional drawing, unit of measurement: mm (inch), decimal separator: period

① M4 threaded mounting hole; 4.2 mm deep, tightening torque 2 Nm
② Tightening torque 2 Nm, screw included in plug unit
③ Receiving range
④ Transmission axis
⑤ Support point
⑥ M4 threaded mounting hole; 5.4 mm deep, tightening torque 2 Nm
⑦ Area in which no reflective surfaces are permitted when the device is mounted

12 Accessories

Accessories and, if applicable, associated mounting information can be found on the product page.

The call is made 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 (supportportal.sick.com, registration required) you will find, besides useful service and support information for your product, further detailed information on the available accessories and their use.

13 Annex

13.1 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).

13.2 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.

See the GNU General Public License for more information.

For license texts, see www.sick.com/licensetexts. Printed copies of the license texts are also available on request.

13.3 Communication interfaces

Communication with the device is possible via CoLa A/B and REST. For more information, see the following English telegram listing.

Under Downloads on the product page, you can download, for example, the Open API file.

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).

13.4 Telegram listing (EN)

Contents

13.4.1.1 About this document

13.4.1.1.1 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.

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

The document 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:

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):

-

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.

13.4.1.1.2 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

13.4.1.2 Communication format

13.4.1.2.1 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 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 15: 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

13.4.1.2.2 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 16: 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)

13.4.1.2.3 Variable types

Variable type	Length (byte)	Value range	Sign
Bool_1	1	0 or 1	No
Uint_8	1	0 ... 255	No
Int_8	1	-128 ... +127	Yes
Uint_16	2	0 ... 65,535	No
Int_16	2	-32,768 ... +32,767	Yes
Uint_32	4	0 ... 4,294,967,295	No
Int_32	4	-2,147,483,648 ... +2,147,483,647	Yes
Enum_8	1	Certain values defined in a list of Choices (0 ... 255)	No
Enum_16	2	Certain values defined in a list of Choices (0 ... 65535)	No
String	Context-dependent	Strings are not terminated in zeroes	

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

Data length is always given in Bytes!

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

13.4.1.2.4 Command basics

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

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

13.4.1.2.5 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 79. 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 86.

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

13.4.1.3 Workflows

13.4.1.3.1 Parameterize the scan

Log in: `sMN SetAccessMode` see "Log in [sMN SetAccessMode] ", page 79

Store parameters: `sMN mEEwriteall` see "Save parameters permanently [sMN mEEwriteall]", page 85

Log out: `sMN Run` see "Set to run [sMN Run]", page 86

13.4.1.3.2 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.

13.4.1.4 Telegrams

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

Table 17: 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 18: 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 19: 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 20: 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"

13.4.1.4.1 Log in [sMN SetAccessMode]

A log in to the device is necessary to change parameters. In most cases, the user level 'Authorized client' is needed. Changed paramters will be reset to the previous state via a reboot unless the are saved. To save paramter changes the command "sMN mEEwriteall" (see "Save parameters permanently [sMN mEEwriteall]", page 85) must be send before log out.

Table 21: 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 22: 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 23: 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 24: 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

13.4.1.4.2 Basic Settings

13.4.1.4.2.1 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 25: 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 26: 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 27: 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 28: 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

13.4.1.4.2.2 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 29: 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 30: 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 31: 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

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

Table 32: 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

13.4.1.4.2.3 Check password [sMN CheckPassword]

Table 33: 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 34: 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 35: 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

Telegram structure: sAN CheckPassword						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Success	Confirmation	Int_8	1	Failed: Success:	0 1	00 01

Table 36: 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

13.4.1.4.2.4 Set Sensitivity Mode [sWN SensitivityMode]

Table 37: Telegram structure: sWN SensitivityMode

Telegram structure: sWN SensitivityMode (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	Sensitivity Mode	String	15		SensitivityMode	53 65 6E 73 69 74 69 76 69 74 79 4D 6F 64 65
Sensitivity	Choose Sensitivity Mode	Enum_8	1	Maximum robustness: Standard (recommended): Maximum detectivity:	0 1 2	00 01 02

Table 38: Example: sWN SensitivityMode

CoLa A	<STX>sWN{SPC}SensitivityMode{SPC}1<ETX>	
	<STX>sWN SensitivityMode 1<ETX>	
	sWN SensitivityMode 1	
	02 73 57 4E 20 53 65 6E 73 69 74 69 76 69 74 79 4D 6F 64 65 20 31 03	
CoLa B	02 02 02 02 00 00 00 15 73 57 4E 20 53 65 6E 73 69 74 69 76 69 74 79 4D 6F 64 65 20 01 05	
	73 57 4E 20 53 65 6E 73 69 74 69 76 69 74 79 4D 6F 64 65 20 01	

Table 39: Telegram structure: sWA SensitivityMode

Telegram structure: sWA SensitivityMode						
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	Sensitivity Mode	String	15		SensitivityMode	53 65 6E 73 69 74 69 76 69 74 79 4D 6F 64 65

Table 40: Example: sWA SensitivityMode

CoLa A	<STX>sWA{SPC}SensitivityMode<ETX>
	02 73 57 41 20 53 65 6E 73 69 74 69 76 69 74 79 4D 6F 64 65 03
CoLa B	02 02 02 02 00 00 00 14 73 57 41 20 53 65 6E 73 69 74 69 76 69 74 79 4D 6F 64 65 20 0B

13.4.1.4.2.5 Set Performance Profile [sWN PerformanceProfileNumber]

Table 41: Telegram structure: sWN PerformanceProfileNumber

Telegram structure: sWN PerformanceProfileNumber (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 Dynamic Sensing Profile	String	24		Performance-ProfileNumber	50 65 72 66 6F 72 6D 61 6E 63 65 50 72 6F 66 69 6C 65 4E 75 6D 62 65 72
Perform-ance Pro-file	Selection of Dynamic Sensing Profile	Enum_8	1	Profile 1: Profile 2: Profile 3: Profile 4: Profile 5: Profile 6: Profile 7: Profile 8: Profile 9: Profile 10:	1 2 3 4 5 6 7 8 9 10	01 02 03 04 05 06 07 08 09 10

Table 42: Example: sWN PerformanceProfileNumber

CoLa A	<STX>sWN{SPC}PerformanceProfileNumber{SPC}2<ETX>	
	<STX>sWN PerformanceProfileNumber 2<ETX>	
	sWN ContaminationConfig 1 3 2 0 1 1	
	02 73 57 4E 20 50 65 72 66 6F 72 6D 61 6E 63 65 50 72 6F 66 69 6C 65 4E 75 6D 62 65 72 20 32 03	
CoLa B	02 02 02 02 00 00 00 1E 73 57 4E 20 50 65 72 66 6F 72 6D 61 6E 63 65 50 72 6F 66 69 6C 65 4E 75 6D 62 65 72 20 02 58	
	73 57 4E 20 50 65 72 66 6F 72 6D 61 6E 63 65 50 72 6F 66 69 6C 65 4E 75 6D 62 65 72 20 02	

Table 43: Telegram structure: sWA PerformanceProfileNumber

Telegram structure: sWA PerformanceProfileNumber						
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 PerformanceProfileNumber						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command	Set Performance Profile	String	24		Performance-ProfileNumber	50 65 72 66 6F 72 6D 61 6E 63 65 50 72 6F 66 69 6C 65 4E 75 6D 62 65 72

Table 44: Example: sWA PerformanceProfileNumber

CoLa A	<STX>sWA{SPC}PerformanceProfileNumber<ETX>
	02 73 57 41 20 50 65 72 66 6F 72 6D 61 6E 63 65 50 72 6F 66 69 6C 65 4E 75 6D 62 65 72 03
CoLa B	02 02 02 02 00 00 00 18 73 57 41 20 50 65 72 66 6F 72 6D 61 6E 63 65 50 72 6F 66 69 6C 65 4E 75 6D 62 65 72 20

13.4.1.4.2.6 Save parameters permanently [sMN mEEwriteall]

Table 45: 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 46: 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 47: 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 48: 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

13.4.1.4.2.7 Set to run [sMN Run]

Log out from device and activate all parameter changes.

Table 49: 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 50: 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 51: 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
Status code	Accepted when value is 1	Bool_1	1	Error: Success:	0 1	00 01

Table 52: 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

13.4.1.4.2.8 Reboot device [sMN mSCreboot]

This command includes saving all parameters.

Table 53: 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 54: 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 55: 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 56: 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

13.4.1.4.3 Measurement output telegram

13.4.1.4.3.1 Poll one telegram [sRN LMDscandata]

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

NOTE
During the bootup phase of the device, there will be no data telegram or answer from the device.

Table 57: 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

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

Table 58: 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 59: 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 88.						

13.4.1.4.3.2 Send data permanently [sEN LMDscandata]

NOTE
During the bootup phase of the device, there will be no data telegram or answer from the device.

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]" 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 60: 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 61: 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 3C
	73 45 4E 20 4C 4D 44 73 63 61 6E 64 61 74 61 20 01

Table 62: 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
Measure-ment	Start/stop	Enum_8	1	Stop: Start:	0 1	00 01

Table 63: 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 64: 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, num-ber): YYWWxxxx	Uint_32	4		0h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
Device sta-tus	(See values column)	Uint_8	2 x 1	Ok: Error:	0 0 1 0	00 00 01 00
Status info						

Telegram structure: sRA LMDscandata / sSN LMDscandata						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Telegram counter	Number of measurement telegrams finished in the scanner and given to the interface. Does not count how many telegrams were really given out; is relevant if not all scans are delivered from the scan core.	Uint_16	2		0h ... FFFFh	00 00 ... FF FF
Scan counter	Number of scans which were created in the device; counts how many scans were really done.	Uint_16	2		0h ... FFFFh	00 00 ... FF FF
Time since start up in µs	Counting the time since power up the device; starting with 0. In the output telegram this is the time at the zero index before the measurement itself starts.	Uint_32	4		0h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
Time of transmission in µs	Time in µs when the complete scan is transmitted to the buffer for data output; starting with 0 at scanner bootup.	Uint_32	4		0h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
Status of digital inputs	Low byte represents input 1.	Uint_8	2 x 1	All inputs low: Input 1 high: Input 2 high: Input 1+2 high: All inputs high:	0 0 1 0 2 0 3 0 F 0	00 00 01 00 02 00 03 00 0F 00
Status of digital outputs	Low byte represents output 1.	Uint_8	2 x 1	All outputs low: All outputs high:	0 0 F 0	00 00 0F 00
Reserved	-	Uint_16	2	Always:	0	00 00 00 00
Frequencies						
Scan frequency	[1/100 Hz]	Uint_32	4	15 Hz:	5DCh	05 DC
Measurement frequency	Inverse of the time between two measurement shots (in 100 Hz).	Uint_32	4	Example: 15 Hz, 1/3° (360*3)/(1/15) = 16.2 kHz	A2h	00 00 00 A2 00 00 0A D1
Reserved						
Reserved	-	Enum_16	2	Always:	0	00 00
16 bit output channel (Distance)						
16 bit channel	16 bit channel that provides measured data	Enum_16	2	Always:	1	00 01
Content	Defines the content of the output channel	String	5	Distance values:	DIST1	44 49 53 54 31

Telegram structure: sRA LMDscandata / sSN LMDscandata						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Scale factor	Scale factor or factor of the measurement values	Real as float according to IEEE754	4	Always Factor x 1:	3F800000h	3F 80 00 00
Reserved	-	Real as float according to IEEE754	4	Always:	00000000	00 00 00 00
Start angle	[1/10000°]	Int_32	4	-45° ... 225°	FFF92230h ... 225510h	FF F9 22 30 ... 00 22 55 1
Size of single angular step	Output format in degree: 1/10000°	Uint_16	2	0.333°: (depends on the angular resolution see "Configure aperture angle of the scandata for output [sWN LMPoutputRange]")	+333d (D05h)	0D 05
Amount of data	Number of transmitted measurement points	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 ... 61A8h	00 00 00 00 ... 00 00 61 A8
8 bit output channel (RSSI)						
8 bit channel	8 bit channel that provides measured RSSI data if activated	Enum_16	2	RSSI active: RSSI inactive:	0 1	00 00 00 01
Content	Defines the content of the output channel (RSSI)	String	5		RSSI1	52 53 53 49 31
Scale factor	Scale factor of the RSSI values	Real as float according to IEEE754	4	Always Factor x 1:	3F800000h	3F 80 00 00
Reserved	-	Real as float according to IEEE754	4	Always:	0h	00 00 00 00
Start angle	Output format: [1/10000°]	Uint_32	4	-45° ... 225°	FFF92230h ... 225510h	FF F9 22 30 ... 00 22 55 1
Size of single angular step	Output format: 1/10000°	Uint_16	2	0.333°: (depends on the angular resolution see "Configure aperture angle of the scandata for output [sWN LMPoutputRange]")	+333d (D05h)	0D 05
Amount of data	Number of transmitted measurement points	Uint_16	2		0h ... FFFFh	00 00 ... FF FF
Data_1 Data_n	Data stream starting Data_1 to Data_n	Uint_8	1		00h ... FFh	00 ... FF
Reserved						

Telegram structure: sRA LMDscandata / sSN LMDscandata						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Reserved	-	Enum_16	2	Always:	0	00 00
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
Reserved						
Reserved	-	Enum_16	2	Always:	0	00 00
Reserved						
Reserved	-	Enum_16	2	Always:	0	00 00
Reserved						
Reserved	-	Uint_16	2	Always:	0	00 00

LMDscandata - reserved values

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

DIST	RSSI	Description
0d	0h	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)	Invalid measurement value, device was dazzled or blinded, e.g. by measuring into the sun
2d	0h	Implausible measurement values
4d – 15d	0h	Reserved, currently not in use
≥16d	>0h	Valid measurement values

max. measurement valu: Dez: 25 000 mm --> Hex: 61A8

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

Example of a telegram stream

Table 65: 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						

Telegram structure: sRA LMDscandata (Example)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Device number	Defined with SOPAS	Uint_16	2		1	00 01
Serial number	Production period (year, calendar week, number): YYWWxxxx	Uint_32	4		1516376h (22111094d)	01 51 63 76
Device status	(See values column)	Uint_8	2 x 1		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		C4C6h (50374d)	C4 C6
Scan counter	Number of scans which were created in the device; counts how many scans were really done.	Uint_16	2		C4E3h (50403d)	C4 E3
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		D22FF57Bh (3526358395d)	D2 2F F5 7B
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		D23019CBh (3526367691d)	D2 30 19 CB
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 Output1: low Output2: low Output3: low Output4:high	8 0	08 00
Reserved	-	Uint_16	2	Always:	0	00 00
Frequencies						
Scan frequency	[1/100 Hz]	Uint_32	4	1500/100 = 15 Hz	5DC (1500d)	00 00 05 DC

Telegram structure: sRA LMDscandata (Example)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Measurement frequency	Inverse of the time between two measurement shots (in 100 Hz). Example: 15 Hz, 1/3° resolution → 1080 shots/67 ms → 16.2 kHz	Uint_32	4		A2h (162d)	00 00 00 A2
Reserved	-	Enum_16	2	Always:	0	00 00
16 bit output channel (Distance)						
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		3F800000h	3F 80 00 00
Reserved	-	Real as float according to IEEE754	4	Always:	0	00 00 00 00
Start angle	[1/10000°]	Uint_32	4	45/10000 = 0.0045° = 0°	FFFFFFD3h (45d)	FF FF FF D3
Size of single angular step	Output format in degree: 1/10000°	Uint_16	2	3333/10000 = 0.33°	D05h (3333d)	0D 05
Amount of data	Number of transmitted measurement points	Uint_16	2	16 Measurement points	10h (16d)	00 10
Data_1 Data_16	Data stream starting Data_1 to Data_16	Uint_16	2	Measurement data Min. 223 mm: DFh Max. 377 mm: 179h	179 165 158 167 150 14F 115 F4 F1 E0 E2 DF E6 E7 D7 D6	01 79 01 65 01 58 01 67 01 50 01 4F 01 15 00 F4 00 F1 00 E0 00 E2 00 DF 00 E6 00 E7 00 D7 00 D6
8 bit output channel (RSSI)						
8 bit channel	8 bit channel that provides measured RSSI data if activated	Enum_16	2	RSSI active:	1	00 01
Content	Defines the content of the output channel (RSSI)	String	5		RSSI1	52 53 53 49 31

Telegram structure: sRA LMDscandata (Example)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Scale factor	Scale factor of the RSSI values	Real as float according to IEEE754	4	Always Factor x 1:	3F800000h	3F 80 00 00
Reserved	-	Real as float according to IEEE754	4	Always:	0h	00 00 00 00
Start angle	Output format: [1/10000°]	Uint_32	4	45/10000 = 0.0045° = 0°	FFFFFFD3h	FF FF FF D3
Size of single angular step	Output format in degree: 1/10000°	Uint_16	2	3333/10000 = 0.33°	D05h (3333d)	0D 05
Amount of data	Number of transmitted measurement points	Uint_16	2		0000h ... FFFFh	00 00 ... FF FF
Data_1 Data_16	Data stream starting Data_1 to Data_16	Uint_8	Length depends on the configured angle range and angular resolution	RSSI data RSSI 124 of 255 (values are not linear): 7Ch	7C 81 86 7C 86 7C 81 77 72 77 6D 72 6D 68 6D 68	7C 81 86 7C 86 7C 81 77 72 77 6D 72 6D 68 6D 68
Reserved						
Reserved	-	Enum_16	2	Always:	0	00 00
Name						
Name	Device name	Enum_16	2	Device name defined:	1	00 01
Length	Length of name	Uint_16	2		B	00 0B
Name	Device name in characters	String	16	The device name is "not defined"	not defined	6E 6F 74 20 64 65 66 69 6E 65 64
Reserved						
Reserved	-	Uint_16	2	Always:	0	00 00
Reserved						
Reserved	-	Uint_16	2	Always:	0	00 00
Reserved						
Reserved	-	Uint_16	2	Always:	0	00 00

Table 66: Example: sRA LMDscandata

CoLa A	<STX>sRA{SPC}LMDscandata{SPC}1{SPC}1{SPC}1516376{SPC}0{SPC}0{SPC}C4C6{SPC}C4E3{SPC}D22FF57B{SPC}D23019CB{SPC}0{SPC}0{SPC}8{SPC}0{SPC}0{SPC}5DC{SPC}A2{SPC}0{SPC}1{SPC}DIST1{SPC}3F800000{SPC}00000000{SPC}FFFFFFD3{SPC}D05{SPC}10{SPC}179{SPC}165{SPC}158{SPC}167{SPC}150{SPC}14F{SPC}115{SPC}F4{SPC}F1{SPC}E0{SPC}E2{SPC}DF{SPC}E6{SPC}E7{SPC}D7{SPC}D6{SPC}1{SPC}RSSI1{SPC}3F800000{SPC}00000000{SPC}FFFFFFD3{SPC}D05{SPC}10{SPC}7C{SPC}81{SPC}86{SPC}7C{SPC}7C{SPC}86{SPC}7C{SPC}81{SPC}77{SPC}72{SPC}77{SPC}6D{SPC}72{SPC}6D{SPC}68{SPC}6D{SPC}68{SPC}0{SPC}1{SPC}B{SPC}not defined{SPC}0{SPC}0{SPC}0<ETX>
CoLa B	02 02 02 02 00 00 00 99 73 52 41 20 4C 4D 44 73 63 61 6E 64 61 74 61 20 00 01 00 01 01 51 63 76 00 00 C4 C6 C4 E3 D2 2F F5 7B 35 26 36 76 91 00 00 08 00 00 00 00 00 05 DC 00 00 00 A2 00 00 00 01 44 49 53 54 31 3F 80 00 00 00 00 00 00 FF FF FF D3 0D 05 00 10 01 79 01 65 01 58 01 67 01 50 01 4F 01 15 00 F4 00 F1 00 E0 00 E2 00 DF 00 E6 00 E7 00 D7 00 D6 00 01 52 53 53 49 31 3F 80 00 00 00 00 00 00 FF FF FF D3 0D 05 00 10 7C 81 86 7C 86 7C 81 77 72 77 6D 72 6D 68 6D 68 00 00 00 01 79

13.4.1.4.3.3 Set scan data enable [sWN ScanDataEnable]

Enables/ Disables streaming data output

Table 67: Telegram structure: sWN ScanDataEnable

Telegram structure: sWN ScanDataEnable (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 E4
Command	Enables/ Disables streaming data output.	String	14		ScanDataEnable	53 63 61 6E 44 61 74 61 45 6E 61 62 6C 65
Data		Bool	1	Off: On:	0d (00h) +1d (01h)	00 ... 01

Table 68: Example: sWN ScanDataEnable 0 - Disable the streaming data output

CoLa A	<STX>sWN{SPC}ScanDataEnable{SPC}0<ETX>	
	<STX>sWN ScanDataEnable 0<ETX>	
	sWN ScanDataEnable 0	
	02 73 57 E4 20 53 63 61 6E 44 61 74 61 45 6E 61 62 6C 65 20 30 03	
CoLa B	02 02 02 02 00 00 00 14 73 57 E4 20 53 63 61 6E 44 61 74 61 45 6E 61 62 6C 65 20 00 44	
	73 57 E4 20 53 63 61 6E 44 61 74 61 45 6E 61 62 6C 65 20 00	

Table 69: Telegram structure: sWA ScanDataEnable

Telegram structure: sWA ScanDataEnable (Required User Level authorized clinet)						
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	Enables/ Disables streaming data output.	String	14		ScanDataEnable	53 63 61 6E 44 61 74 61 45 6E 61 62 6C 65

Table 70: Example: sWA ScanDataEnable

CoLa A	<STX>sWA{SPC}ScanDataEnable<ETX>
	02 73 57 41 20 53 63 61 6E 44 61 74 61 45 6E 61 62 6C 65 03

CoLa B	02 02 02 02 00 00 00 13 73 57 41 20 53 63 61 6E 44 61 74 61 45 6E 61 62 6C 65 20 4B

13.4.1.4.3.4 Set streaming ethernet settings [sWN ScanDataEthSettings]

Ethernet settings for the scan data streaming functionality of the device

Table 71: Telegram structure: sWN ScanDataEthSettings

Telegram structure: sWN ScanDataEthSettings (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	String	19		ScanDataEth-Settings	53 63 61 6E 44 61 74 61 45 74 68 53 65 74 74 69 6E 67 73
Protocol	Transport protocol for streaming data	Enum_8	1	UDP: TCP:	+1d (01h) +2d (02h)	01 02
IPAddress	IP address of the destination for data receiver	Array	4		0 ...+255d (00...FF) 0 ...+255d (00..FF) 0 ...+255d (00...FF) 0 ...+255d (00...FF)	00 ... FF 00 ... FF 00 ... FF 00 ... FF
Port	Port destination of the data reseiver	Uint_16	2		0 .. +65535d(00 00...FF FF)	00 00 ... FF FF

Example: sWN ScanDataEthSettings +1 +192 +168 +0 +100 +2115

Protocol is set to UPD (1), IPAddress (192.168.0.100), Port (2115)

Table 72: Example: sWN ScanDataEthSettings

CoLa A	<STX>sWN{SPC}ScanDataEthSettings{SPC}+1{SPC}+192{SPC}+168{SPC}+0{SPC}+100{SPC}+2115<ETX>	
	<STX>sWN ScanDataEthSettings +1 +192 +168 +0 +100 +2115<ETX>	
	sWN ScanDataEthSettings +1 +192 +168 +0 +100 +2115	
	02 73 57 4E 20 53 63 61 6E 44 61 74 61 45 74 68 53 65 74 74 69 6E 67 73 20 31 20 43 30 20 41 38 20 30 20 36 34 20 38 34 33 03	
CoLa B	02 02 02 02 00 00 00 1F 73 57 4E 20 53 63 61 6E 44 61 74 61 45 74 68 53 65 74 74 69 6E 67 73 20 01 C0 A8 00 64 08 43 5F	
	73 57 4E 20 53 63 61 6E 44 61 74 61 45 74 68 53 65 74 74 69 6E 67 73 20 01 C0 A8 00 64 08 43	

Table 73: Telegram structure: sWA ScanDataEthSettings

Telegram structure: sWA ScanDataEthSettings						
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 ScanDataEthSettings						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command	String	String	19		ScanDataEth-Settings	53 63 61 6E 44 61 74 61 45 74 68 53 65 74 74 69 6E 67 73

Table 74: Example: sWA ScanDataEthSettings

CoLa A	<STX>sWA{SPC}ScanDataEthSettings<ETX>
	02 73 57 41 20 53 63 61 6E 44 61 74 61 45 74 68 53 65 74 74 69 6E 67 73 03
CoLa B	02 02 02 02 00 00 00 18 73 57 41 20 53 63 61 6E 44 61 74 61 45 74 68 53 65 74 74 69 6E 67 73 20 16

13.4.1.4.3.5 Set IMU data enable [sWN ImuDataEnable]

Enables/ Disables streaming IMU data output.

Table 75: Telegram structure: sWN ImuDataEnable

Telegram structure: sWN ImuDataEnable (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 E4
Command	Set streaming IMU data output.	String	13		ImuDataEnable	49 6D 75 44 61 74 61 45 6E 61 62 6C 65
IMU data stream	Enable/ disable	Bool	1	Disbale: Enable:	0d (00h) +1d (01h)	00 ... 01

Table 76: Example: sWN ImuDataEnable 0 - Disable the streaming IMU data output

CoLa A	<STX>sWN{SPC}ImuDataEnable{SPC}0<ETX>	
	<STX>sWN ImuDataEnable 0<ETX>	
	sWN ImuDataEnable 0	
	02 73 57 E4 20 49 6D 75 44 61 74 61 45 6E 61 62 6C 65 20 30 03	
CoLa B	02 02 02 02 00 00 00 13 73 57 E4 20 49 6D 75 44 61 74 61 45 6E 61 62 6C 65 20 00 2A	
	73 57 E4 20 49 6D 75 44 61 74 61 45 6E 61 62 6C 65 20 00	

Table 77: Telegram structure: sWA ImuDataEnable

Telegram structure: sWA ImuDataEnable						
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 streaming IMU data output.	String	13		ImuDataEnable	49 6D 75 44 61 74 61 45 6E 61 62 6C 65

Table 78: Example: sWA ImuDataEnable

CoLa A	<STX>sWA{SPC}ImuDataEnable<ETX>
	02 73 57 41 20 49 6D 75 44 61 74 61 45 6E 61 62 6C 65 03
CoLa B	02 02 02 02 00 00 00 12 73 57 41 20 49 6D 75 44 61 74 61 45 6E 61 62 6C 65 20 25

13.4.1.4.3.6 Set IMU data streaming ethernet settings [sWN ImuDataEthSettings]

Ethernet settings for the IMU data streaming functionality of the device

Table 79: Telegram structure: sWN ImuDataEthSettings

Telegram structure: sWN ImuDataEthSettings (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 IMU data streaming ethernet settings	String	18		ImuDataEthSettings	49 6D 75 44 61 74 61 45 74 68 53 65 74 74 69 6E 67 73
Protocol	Transport protocol for streaming IMU data	Enum_8	1	UDP:	+1d (01h)	01
IP address	IP address of the destination for IMU data receiver	Array	4		0 ...+255d (00...FF) 0 ...+255d (00..FF) 0 ...+255d (00...FF) 0 ...+255d (00...FF)	00 ... FF 00 ... FF 00 ... FF 00 ... FF
Port	Port destination of the IMU data reseiver	Uint_16	2		0 .. +65535d(00 00...FF FF)	00 00 ... FF FF

Example: sWN ImuDataEthSettings +1 +192 +168 +0 +100 +7503

Protocol is set to UPD (1), IPAddress (192.168.0.100), Port (7503)

Table 80: Example: sWN ImuDataEthSettings

CoLa A	<STX>sWN{SPC}ImuDataEthSettings{SPC}+1{SPC}+192{SPC}+168{SPC}+0{SPC}+100{SPC}+7503<ETX>	
	<STX>sWN ImuDataEthSettings +1 +192 +168 +0 +100 +7503<ETX>	
	sWN ImuDataEthSettings +1 +192 +168 +0 +100 +7503	
	02 73 57 4E 20 49 6D 75 44 61 74 61 45 74 68 53 65 74 74 69 6E 67 73 20 31 20 43 30 20 41 38 20 30 20 36 34 20 31 44 34 46 03	
CoLa B	02 02 02 02 00 00 00 1E 73 57 4E 20 49 6D 75 44 61 74 61 45 74 68 53 65 74 74 69 6E 67 73 20 01 C0 A8 00 64 1D 4F 28	
	73 57 4E 20 49 6D 75 44 61 74 61 45 74 68 53 65 74 74 69 6E 67 73 20 01 C0 A8 00 64 1D 4F	

Table 81: Telegram structure: sWA ImuDataEthSettings

Telegram structure: sWA ImuDataEthSettings						
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 IMU data streaming ethernet settings	String	18		ImuDataEthSettings	49 6D 75 44 61 74 61 45 74 68 53 65 74 74 69 6E 67 73

Table 82: Example: sWA ImuDataEthSettings

CoLa A	<STX>sWA{SPC}ImuDataEthSettings<ETX>
	02 73 57 41 20 49 6D 75 44 61 74 61 45 74 68 53 65 74 74 69 6E 67 73 03
CoLa B	02 02 02 02 00 00 00 17 73 57 41 20 49 6D 75 44 61 74 61 45 74 68 53 65 74 74 69 6E 67 73 20 78

13.4.1.4.3.7 Read scan data format [sRN ScanDataFormat]

Return of the scan data format

Table 83: Telegram structure: sRN ScanDataFormat

Telegram structure: sRN ScanDataFormat (User Level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E
Command	Data serialization format	String	14		ScanDataFormat	53 63 61 6E 44 61 74 61 46 6F 72 6D 61 74

Example: sRN ScanDataFormat

Read of the data serialization format

Table 84: Example: sRN ScanDataFormat

CoLa A	<STX>sRN{SPC}ScanDataFormat<ETX>	
	<STX>sRN ScanDataFormat<ETX>	
	sRN ScanDataFormat	
	02 73 52 4E 20 53 63 61 6E 44 61 74 61 46 6F 72 6D 61 74 03	
CoLa B	02 02 02 02 00 00 00 12 73 52 4E 20 53 63 61 6E 44 61 74 61 46 6F 72 6D 61 74 63	
	73 52 4E 20 53 63 61 6E 44 61 74 61 46 6F 72 6D 61 74	

Table 85: Telegram structure: sRA ScanDataFormat

Telegram structure: sRA ScanDataFormat						
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 ScanDataFormat						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command	Data serialization format	String	14		ScanDataFor-mat	53 63 61 6E 44 61 74 61 46 6F 72 6D 61 74
Data		Enum_8	1	MSGPACK Compact	1h 2h	01 02

Example: sRA ScanDataFormat

Scan data format is set to Compact = 2

Table 86: Example: sRA ScanDataFormat

CoLa A	<STX>sRA{SPC}ScanDataFormat{SPC}2<ETX>
	02 73 57 41 20 53 63 61 6E 44 61 74 61 46 6F 72 6D 61 74 20 32 03
CoLa B	02 02 02 02 00 00 00 14 73 57 41 20 53 63 61 6E 44 61 74 61 46 6F 72 6D 61 74 20 02 4E

13.4.1.4.3.8 Set Scan data format [sWN ScanDataFormat]

Set the data serialization format

Table 87: Telegram structure: sWN ScanDataFormat

Telegram structure: sWN ScanDataFormat (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	Data serialization for-mat	String	14		ScanDataFor-mat	53 63 61 6E 44 61 74 61 46 6F 72 6D 61 74
Variable Data	Data	Enum_8	1	MSGPACK: Compact:	+1d (1h) +2d (2h)	01 02

Example: sWN ScanDataFormat

Scan data format set to **Compact** format

Table 88: Example: sWN ScanDataFormat

CoLa A	<STX>sWN{SPC}ScanDataFormat{SPC}2<ETX>	
	<STX>sWN ScanDataFormat 2<ETX>	
	sWN ScanDataFormat 2	
	02 73 57 4E 20 53 63 61 6E 44 61 74 61 46 6F 72 6D 61 74 20 32 03	
CoLa B	02 02 02 02 00 00 14 73 57 4E 20 53 63 61 6E 44 61 74 61 46 6F 72 6D 61 74 20 02 44	
	73 57 4E 20 53 63 61 6E 44 61 74 61 46 6F 72 6D 61 74 20 02	

Table 89: Telegram structure: sWA ScanDataFormat

Telegram structure: sWA ScanDataFormat						
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	Data serialization format	String	14		ScanDataFormat	53 63 61 6E 44 61 74 61 46 6F 72 6D 61 74

Table 90: Example: sWA ScanDataFormat

CoLa A	<STX>sWA{SPC}ScanDataFormat<ETX>
	02 73 57 41 20 53 63 61 6E 44 61 74 61 46 6F 72 6D 61 74 03
CoLa B	02 02 02 02 00 00 00 13 73 57 41 20 53 63 61 6E 44 61 74 61 46 6F 72 6D 61 74 20 49

13.4.1.4.4 Time stamp

13.4.1.4.4.1 Set time stamp [sMN LSPsetdatetime]

NOTE
Does only work if NTP- respectively PTP-Client is not active.

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.

- Therefore, it is recommended to use a NTP or a PTP setting.

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.

Table 91: 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

Telegram structure: sMN LSPsetdatetime (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
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 92: Example 1: sMN LSPsetdatetime

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 A3	
	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 93: Example 2: sMN LSPsetdatetime

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 A3	
	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 94: 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 95: 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”.

13.4.1.4.4.2 Set NTP (Network Time Protocol) parameters

13.4.1.4.4.2.1 Set time synchronization [sWN TSCRole]

Table 96: 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 NTP role	String	7		TSCRole	54 53 43 52 6F 6C 65
Variable data	NTP role	Uint_8	1	None: Client: Server:	0 1 2	00 01 02

Table 97: 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 98: 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 99: 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

13.4.1.4.4.2.2 Set time server IP address [sWN TSCTCSrvAddr]

Table 100: 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 101: 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 102: 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 103: 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

13.4.1.4.4.2.3 Set time zone [sWN TSCTCtimezone]

Table 104: 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	Select the time zone of the client	Enum_8	1	List of time zones see table 105, page 106	+0d ... +104d (00h ... 68h)	00 ... 68

Table 105: Time zone data Values CoLa (ASCII)

0	DATE_LINE_STANDARD	35	MONROVIA_REYKJAVIK	70	MUMBAI_NEUDELHI
1	COORD_WORLD_TIME_11	36	AMSTERDAM_BERLIN_ROM	71	SRI_JAYAWARDENEPURA
2	HAWAII	37	BELGRAD_BUDAPEST_PRAG	72	KATMANDU
3	ALASKA	38	BRUESSEL_MADRID_PARIS	73	ASTANA
4	CALIFORNIA	39	SARAJEVO_WARSCHAU	74	DAKKA
5	USA_CANADA	40	WEST_CENTRAL_AFRICA	75	NOWOSIBIRSK
6	ARIZONA	41	WINDHUK	76	YANGON
7	LA_PAZ	42	AMMAN	77	BANGKOK_HANOI_JAKARTA
8	MOUNTAIN_TIME_USA	43	ATHEN_BUKAREST	78	KRASNOJARSK
9	CENTRAL_TIME_USA	44	BEIRUT	79	IRKUTSK
10	MEXICO_CITY	45	DAMASCUS	80	KUALA_LUMPUR_SINGAPUR
11	MIDDLE_AMERICA	46	HARARE_PRETORIA	81	PEKING_HONGKONG
12	SASKATCHEWAN	47	HELSINKI_KIEW_RIGA	82	PERTH
13	BOGOTA_LIMA	48	ISTANBUL	83	TAIPEH
14	EASTERN_TIME_USA	49	JERUSALEM	84	ULAN_BATOR
15	INDIANA	50	KAIRO	85	JAKUTSK
16	CARACAS	51	KALININGRAD	86	OSAKA_TOKIO
17	ASUNCION	52	EASTERN_EUROPE	87	SEOUL
18	ATLANTIC_KANADA	53	TRIPOLIS	88	ADELAIDE
19	CUIABA	54	BAGDAD	89	DARWIN
20	LAPAZ_SANJUAN	55	KUWAIT_RIAD	90	BRISBANE
21	SANTIAGO	56	MINSK	91	CANBERRA_SYDNEY
22	NEUFUNDLAND	57	MOSKAU_PETERSBURG	92	GUAM_PORT_MORESBY
23	BRASILIA	58	NAIROBI	93	HOBART
24	BUENOS_AIRES	59	TEHERAN	94	MAGADAN
25	CAYENNE_FORTALEZA	60	ABU_DHABI	95	WLADIWOSTOK
26	GROENLAND	61	BAKU	96	SALOMONEN_KALEDONIEN
27	MONTEVIDEO	62	ERIWAN	97	TSCHOKURDACH
28	SALVADOR	63	ISCHEWSK_SAMARA	98	ANADYR
29	COORD_WORLD_TIME_02	64	PORT_LOUIS	99	AUCKLAND_WELLINGTON
30	AZOREN	65	TIFLIS	100	FIDSCHI

31	KAP_VERDE	66	KABUL	101	COORD_WORLD_TIME_12
32	CASABLANCA	67	ASCHGABET_TASCHKENT	102	NAKUALOFA
33	DUBLIN_LISSABON_LONDON	68	ISLAMABAD_KARATSCHI	103	SAMOA
34	COORD_WORLD_TIME	69	JEKATERINBURG	104	KIRITIMATI

Table 106: Example: sWN TSCTCtimezone Amsterdam, Berlin, Rom

CoLa A	<STX>sWN{SPC}TSCTCtimezone{SPC}+36<ETX>
	<STX>sWN TSCTCtimezone +36<ETX>
	sWN TSCTCtimezone +36
	02 73 57 4E 20 54 53 43 54 43 74 69 6D 65 7A 6F 6E 65 20 24 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 24 16
	73 57 4E 20 54 53 43 54 43 74 69 6D 65 7A 6F 6E 65 20 24

Table 107: Telegram structure: sWA TSCTCtimezone

Telegram structure: sWA TSCTCtimezone						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sWA	73 57 41
Command	Set time zone	String	13		TSCTCtimezone	54 53 43 54 43 74 69 6D 65 7A 6F 6E 65

Table 108: 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

13.4.1.4.4.2.4 Set update time [sWN TSCTCupdatetime]

Table 109: 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 110: 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 111: 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 112: 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

13.4.1.4.5 Filter

13.4.1.4.5.1 Set particle filter [sWN LFPparticle]

Table 113: 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 dvice to handle the particles.

1) 1)

Table 114: 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 115: Telegram structure: sWA LFPparticle

Telegram structure: sWA LFPparticle						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sWA	73 57 41
Command	Set particle filter	String	11		LFPparticle	4C 46 50 70 61 72 74 69 63 6C 65

Table 116: 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

13.4.1.4.5.2 Set echo filter [sWN FREchoFilter]

NOTE
Only available with firmware versions > V1.10.

Table 117: 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 118: 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 119: 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 120: 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

13.4.1.4.5.3 Set sensitivity fog filter [sWN MCSenseLevel]

Table 121: 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 122: 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 123: 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

Telegram structure: sWA MCSenseLevel						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command	Sense level	String	12		MCSenseLevel	4D 43 53 65 6E 73 65 4C 65 76 65 6C

Table 124: 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

13.4.1.4.5.4 Set cubic area filter [sWN LFPcubicareafilter]

The cubic area filter limits a polar scan to a axisparallel cube defined by its extension in x-, y- and z-range.

Table 125: Telegram structure: sWN LFPcubicareafilter

Telegram structure: sWN LFPcubicareafilter (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	CubicAreaFilter limits a polar scan to a axisparallel cube	String	18		LFPcubicareafilter	4C 46 50 63 75 62 69 63 61 72 65 61 66 69 6C 74 65 72
Variable data 1	Enables/Disables the filter.	Bool	1	Off: On:	0d 1d	00 01
Variable Data 2	X min 1/10 mm	Int_32	4		-200000d... +200000d	FF FF B1 E0 ... 00 00 4E 20
Variable Data 3	X max 1/10 mm	Int_32	4		-200000d... +200000d	FF FF B1 E0 ... 00 00 4E 20
Variable Data 4	Y min 1/10 mm	Int_32	4		-200000d... +200000d	FF FF B1 E0 ... 00 00 4E 20
Variable Data 5	Y max 1/10 mm	Int_32	4		-200000d... +200000d	FF FF B1 E0 ... 00 00 4E 20
Variable Data 6	Z min 1/10 mm	Int_32	4		-200000d... +200000d	FF FF B1 E0 ... 00 00 4E 20
Variable Data 7	Z max 1/10 mm	Int_32	4		-200000d... +200000d	FF FF B1 E0 ... 00 00 4E 20

Disables the cubic area filter and set up to the -20000mm...+20000mm in x,y,z direction.

Table 126: Example: sWN LFPcubicAreafilter

CoLa A	<STX>sWN{SPC}LFPcubicfreafilter{SPC}0{SPC}FFFFB1E0{SPC}E420{SPC}FFFFB1E0{SPC}E420{SPC}FFFFB1E0{SPC}E420<ETX>	
	<STX>sWN LFPcubicareafilter 0 FFFFB1E0 E420 FFFFB1E0 E420 FFFFB1E0 E420<ETX>	
	sWN LFPcubicareafilter 0 FFFFB1E0 E420 FFFFB1E0 E420 FFFFB1E0 E420	
	02 73 57 4E 20 4C 46 50 63 75 62 69 63 61 72 65 61 66 69 6C 74 65 72 20 30 20 46 46 46 46 42 31 45 30 20 34 45 32 30 20 46 46 46 46 42 31 45 30 20 34 45 32 30 20 46 46 46 46 42 31 45 30 20 34 45 32 30 03	

CoLa B	02 02 02 02 00 00 00 30 73 57 4E 20 4C 46 50 63 75 62 69 63 61 72 65 61 66 69 6C 74 65 72 20 00 FF FF B1 E0 00 00 4E 20 FF FF B1 E0 00 00 4E 20 FF FF B1 E0 00 00 4E 20 66
	73 57 4E 20 4C 46 50 63 75 62 69 63 61 72 65 61 66 69 6C 74 65 72 20 00 FF FF B1 E0 00 00 4E 20 FF FF B1 E0 00 00 4E 20 FF FF B1 E0 00 00 4E 20

Table 127: Telegram structure: sWA LFPcubicareafilter

Telegram structure: sWALFPcubicareafilter						
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	CubicAreaFilter limits a polar scan to a axisparallel cube	String	18		LFPcubicareafilter	4C 46 50 63 75 62 69 63 61 72 65 61 66 69 6C 74 65 72

Table 128: Example: sWA LFPcubicareafilter

CoLa A	<STX>sWA{SPC}LFPcubicareafilter<ETX>
	02 73 57 41 20 4C 46 50 63 75 62 69 63 61 72 65 61 66 69 6C 74 65 72 03
CoLa B	02 02 02 02 00 00 00 17 73 57 41 20 4C 46 50 63 75 62 69 63 61 72 65 61 66 69 6C 74 65 72 20 56

13.4.1.4.5.5 Activate long range mode [sWN EnableLongRangeMode]

Extends the maximum scanning distance.

Table 129: Telegram structure: sWN EnableLongRangeMode

Telegram structure: sWN EnableLongRangeMode (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	Enable long range mode	String	19		EnableLongRangeMode	45 6E 61 62 6C 65 4C 6F 6E 67 52 61 6E 67 65 4D 6F 64 65
Status code	Activate / deactivate long range mode	Bool_1	1	Inactive: Active:	0 1	00 01

Enable EnableLongRangeMode

Table 130: Example: sWN EnableLongRangeMode 1

CoLa A	<STX>sWN{SPC}EnableLongRangeMode{SPC}1<ETX>
	<STX>sWN EnableLongRangeMode 1<ETX>
	sWN EnableLongRangeMode 1
	02 73 57 4E 20 45 6E 61 62 6C 65 4C 6F 6E 67 52 61 6E 67 65 4D 6F 64 65 20 31 03
CoLa B	02 02 02 02 00 00 00 19 73 57 4E 20 45 6E 61 62 6C 65 4C 6F 6E 67 52 61 6E 67 65 4D 6F 64 65 20 01 1C
	73 57 4E 20 45 6E 61 62 6C 65 4C 6F 6E 67 52 61 6E 67 65 4D 6F 64 65 20 01

Table 131: Telegram structure: sWA EnableLongeRangeModer

Telegram structure: sWA EnableLongRangeMode						
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	Enable long range mode	String	19		EnableLon-gRangeMode	45 6E 61 62 6C 65 4C 6F 6E 67 52 61 6E 67 65 4D 6F 64 65

Table 132: Example: sWA EnableLongeRangeMode

CoLa A	<STX>sWA{SPC}EnableLongRangeMode<ETX>
	02 73 57 41 20 45 6E 61 62 6C 65 4C 6F 6E 67 52 61 6E 67 65 4D 6F 64 65 03
CoLa B	02 02 02 02 00 00 00 18 73 57 41 20 45 6E 61 62 6C 65 4C 6F 6E 67 52 61 6E 67 65 4D 6F 64 65 20 12

13.4.1.4.5.6 Read status of long range mode [sRN EnableLongRangeMode]

Table 133: Telegram structure: sRN EnableLongRangeMode

Telegram structure: sRN EnableLongRangeMode						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 57 4E
Command	Enable long range mode	String	19		EnableLon-gRangeMode	45 6E 61 62 6C 65 4C 6F 6E 67 52 61 6E 67 65 4D 6F 64 65

Request status Long range mode

Table 134: Example: sRN EnableLongRangeMode

CoLa A	<STX>sRN{SPC}EnableLongRangeMode<ETX>	
	<STX>sRN EnableLongRangeMode<ETX>	
	sRN EnableLongRangeMode	
	02 73 57 4E 20 45 6E 61 62 6C 65 4C 6F 6E 67 52 61 6E 67 65 4D 6F 64 03	
CoLa B	02 02 02 02 00 00 00 17 73 57 4E 20 45 6E 61 62 6C 65 4C 6F 6E 67 52 61 6E 67 65 4D 6F 64 65 38	
	73 57 4E 20 45 6E 61 62 6C 65 4C 6F 6E 67 52 61 6E 67 65 4D 6F 64 65	

Table 135: Telegram structure: sRA EnableLongRangeMode

Telegram structure: sRA EnableLongRangeMode						
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	Sector function	String	19		EnableLon-gRangeMode	45 6E 61 62 6C 65 4C 6F 6E 67 52 61 6E 67 65 4D 6F 64 65

Telegram structure: sRA EnableLongRangeMode						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Status code	Activate / deactivate long range mode	Bool_1	1	Inactive: Active:	0 1	00 01

Table 136: Example: sWA EnableLongrangeMode

CoLa A	<STX>sRA{SPC}EnableLongRangeMode{SPC}1<ETX>
	02 73 52 41 20 45 6E 61 62 6C 65 4C 6F 6E 67 52 61 6E 67 65 4D 6F 64 65 20 31 03
CoLa B	02 02 02 02 00 00 00 13 73 52 41 20 45 6E 61 62 6C 65 4C 6F 6E 67 52 61 6E 67 65 4D 6F 64 65 20 01 16

13.4.1.4.5.7 Set angle range filter [sWN LFPangleRangeFilter]

The angle range filter set up the horizontal (theta) and vertical (phi) start- and stop angle in rad.

With multiScan only the horizontal (theta) angle is adjustable. To adjust the vertical limits use the layer filter (LFPlayerFilter)

BeamIncrement = the 'beamIncrement' which is used to subsample the beams within the selected angle range. With a 'beamIncrement' of n only every nth beam from the selected angle range is copied to the output scan, i.e. the angle resolution is reduced by factor n. If the beamIncrement is zero it is set to one.

Table 137: Telegram structure: sWN LFPangleRangeFilter

Telegram structure: sWN LFPangleRangeFilter (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	filter set up the horizontal (theta) and vertical (phi) start- and stop angle in rad	String	19		LFPangleRange-Filter	4C 46 50 61 6E 67 6C 65 52 61 6E 67 65 46 69 6C 74 65 72
Variable Data 1	Enables/Disables the filter	Bool_1	1	Off: On:	+0d +1d	00 01
Variable Data 2	ThetaStart	Real	4	No impact on sensor setting but needs to be filled (see example)	-1,800,000d ... +1,800,000d (FFE488C0 ... 001B7740h)	FF E4 88 C0 ... 00 1B 77 40
Variable Data 3	ThetaStop	Real	4	No impact on sensor setting but needs to be filled (see example)	-1,800,000d ... +1,800,000d (FFE488C0 ... 001B7740h)	FF E4 88 C0 ... 00 1B 77 40
Variable Data 4	PhiStart	Real	4	No impact on sensor setting but needs to be filled (see example)	-900,000d ... +900,000d (FFF24460 ... 000DBBA0h)	FF F2 44 60 ... 00 0D BB A0
Variable Data 5	PhiStop	Real	4	No impact on sensor setting but needs to be filled (see example)	-900,000d ... +900,000d (FFF24460 ... 000DBBA0h)	FF F2 44 60 ... 00 0D BB A0
Variable Data 6	BeamIncrement	UInt_16	2	No impact on sensor setting but needs to be filled (see example)	1d...+20d	00 01 ... 00 14

Explanation: Enable the angle range filter and set up theta (horizontal) start -90°, theta stop +90°, phi (vertical) start -90°, phi stop +90°, beam increment 1

Table 138: Example: sWN LFPanlgeRangeFilter

CoLa A	<STX>sWN{SPC}LFPangleRangeFilter{SPC}1{SPC}-900000{SPC}+900000{SPC}-900000{SPC}+900000{SPC}1<ETX>
	<STX>sWN LFPangleRangeFilter 1 -900000 +900000 -900000 +900000 1<ETX>
	sWN LFPangleRangeFilter 1 -900000 +900000 -900000 +900000 1
	02 73 57 4E 4C 46 50 61 6E 67 6C 65 52 61 6E 67 65 46 69 6C 74 65 72 20 31 20 2D 39 30 30 30 30 20 2B 39 30 30 30 30 30 20 2D 39 30 30 30 30 30 20 2B 39 30 30 30 30 30 20 31 03
CoLa B	02 02 02 02 00 00 00 2B 73 57 4E 20 4C 46 50 61 6E 67 6C 65 52 61 6E 67 65 46 69 6C 74 65 72 20 01 FF F2 44 60 00 0D BB A0 FF F2 44 60 00 0D BB A0 00 01 2E
	73 57 4E 20 4C 46 50 61 6E 67 6C 65 52 61 6E 67 65 46 69 6C 74 65 72 20 01 FF F2 44 60 00 0D BB A0 FF F2 44 60 00 0D BB A0 00 01

Table 139: Telegram structure: sWA LFP AngleRangeFilter

Telegram structure: sWA LFPangleRangeFilter						
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	filter set up the horizontal (theta) and vertical (phi) start- and stop angle in rad	String	19		LFPangleRange-Filter	4C 46 50 61 6E 67 6C 65 52 61 6E 67 65 46 69 6C 74 65 72

Table 140: Example: sWA LFPangleRangeFilter

CoLa A	<STX>sWA{SPC}LFPangleRangeFilter<ETX>
	02 73 57 41 20 4C 46 50 61 6E 67 6C 65 52 61 6E 67 65 46 69 6C 74 65 72 03
CoLa B	02 02 02 02 00 00 00 18 73 57 41 20 4C 46 50 61 6E 67 6C 65 52 61 6E 67 65 46 69 6C 74 65 72 20 21

13.4.1.4.5.8 Set interval filter [sWN LFPintervalFilter]

Enables and set up the interval filter. The interval filter reduce the scan output rate by a given factor.

Table 141: Telegram structure: sWN LFPintervalFilter

Telegram structure: sWN LFPintervalFilter (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	Reduce the scan output rate by a given factor	String	17		LFPintervalFilter	4C 46 50 69 6E 74 65 72 76 61 6C 46 69 6C 74 65 72
Variable data 1	Enables/Disables the filter.	Bool	1	Off: On:	0d (00h) +1d (01h)	00 01
Variable Data 2	Only every nth scan is output where n is given by the value of uiReductionFactor.	Uint_32	4		1d...+50d (00 00 00 01h ... 00 00 00 32h)	00 00 00 01 ... 00 00 00 32

Enables the interval filter an set up to the 3rd scan

Table 142: Example: sWN LFPintervalFilter

CoLa A	<STX>sWN{SPC}LFPintervalFilter{SPC}1{SPC}3<ETX>
	<STX>sWN LFPintervalFilter 1 3<ETX>
	sWN LFPintervalFilter 1 3
	02 73 57 4E 20 4C 46 50 69 6E 74 65 72 76 61 6C 46 69 6C 74 65 72 20 31 20 33 03
CoLa B	02 02 02 02 00 00 00 1B 73 57 4E 20 4C 46 50 69 6E 74 65 72 76 61 6C 46 69 6C 74 65 72 20 01 00 00 00 03 0E
	73 57 4E 20 4C 46 50 69 6E 74 65 72 76 61 6C 46 69 6C 74 65 72 20 01 00 00 00 03

Table 143: Telegram structure: sWA LFPintervalFilter

Telegram structure: sWA LFPintervalFilter						
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	Reduce the scan output rate by a given factor	String	17		LFPintervalFilter	4C 46 50 69 6E 74 65 72 76 61 6C 46 69 6C 74 65 72

Table 144: Example: sWA LFPintervalFilter

CoLa A	<STX>sWA{SPC}LFPintervalFilter<ETX>
	02 73 57 41 20 46 50 69 6E 74 65 72 76 61 6C 46 69 6C 74 65 72 03
CoLa B	02 02 02 02 00 00 00 16 73 57 41 20 4C 46 50 69 6E 74 65 72 76 61 6C 46 69 6C 74 65 72 20 00

13.4.1.4.5.9 Set moving averaging filter [sWN LFPmovingAveragingFilter]

Enables the moving average filter

Table 145: Telegram structure: sWN LFPmovingAveragingFilter

Telegram structure: sWN LFPmovingAveragingFilter (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	String	24		LFPmovingAveragingFilter	4C 46 50 6D 6F 76 69 6E 67 41 76 65 72 61 67 69 6E 67 46 69 6C 74 65 72
Variable Data 1	Moving averaging is enabled	Bool	1	Off: On:	0d (00h) +1d (01h)	00 01
Variable Data 2	averaging depth	UInt	2	Minimum: Maximum:	+2d (02h) +4d (0Ah)	00 02 00 04

Disable the moving average filter and set averaging depth to 3

Table 146: Example: sWN LFPmovingAveragingFilter +0 +3

CoLa A	<STX>sWN{SPC}LFPmovingAveragingFilter{SPC}+0{SPC}+3<ETX>
	<STX>sWN LFPmovingAveragingFilter +0 +3<ETX>
	sWN LFPmovingAveragingFilter +0 +3
	02 73 57 4E 20 4C 46 50 6D 6F 76 69 6E 67 41 76 65 72 61 67 69 6E 67 46 69 6C 74 65 72 20 2B 30 20 2B 33 03
CoLa B	02 02 02 02 00 00 00 20 73 57 4E 20 4C 46 50 6D 6F 76 69 6E 67 41 76 65 72 61 67 69 6E 67 46 69 6C 74 65 72 20 01 00 03 41
	73 57 4E 20 4C 46 50 6D 6F 76 69 6E 67 41 76 65 72 61 67 69 6E 67 46 69 6C 74 65 72 20 01 00 03

Table 147: Telegram structure: sWA LFPmovingAveragingFilter

Telegram structure: sWN LFPmovingAveragingFilter						
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	String	24		LFPmovingAveragingFilter	4C 46 50 6D 6F 76 69 6E 67 41 76 65 72 61 67 69 6E 67 46 69 6C 74 65 72

Table 148: Example: sWA LFPmovingAveragingFilter

CoLa A	<STX>sWA{SPC}LFPmovingAveragingFilter<ETX>
	02 73 57 41 20 4C 46 50 6D 6F 76 69 6E 67 41 76 65 72 61 67 69 6E 67 46 69 6C 74 65 72 03
CoLa B	02 02 02 02 00 00 00 1D 73 57 41 20 4C 46 50 6D 6F 76 69 6E 67 41 76 65 72 61 67 69 6E 67 46 69 6C 74 65 72 20 4D

13.4.1.4.5.10 Set radial distance range filter [sWN LFPradialDistanceRangeFilter]

Restriction of the scan(s) to a specified distance range.

Table 149: Telegram structure: sWN LFPradialDistanceRangeFilter

Telegram structure: sWN LFPradialDistanceRangeFilter (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	Restriction of the scan(s) to a specified distance range.	String	28		LFPradialDistanceRangeFilter	4C 46 50 72 61 64 69 61 6C 44 69 73 74 61 6E 63 65 52 61 6E 67 65 46 69 6C 74 65 72
Variable Data 1	Enables/Disables the filter.	Bool_1	1	Off: On:	0d (0h) 1d (1h)	00 01
Variable Data 2	DistMin: Lower boundary of the distance range.	Int_32	4	Minimum: Maximum:	+0d (0h) +200000d (30D40h)	00 00 00 00 00 03 0D 40

Telegram structure: sWN LFPradialDistanceRangeFilter (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Variable Data 3	DistMax: Upper boundary of the distance range.	Int_32	4	Minimum: Maximum:	+0d (0h) +200000d (30D40h)	00 00 00 00 00 03 0D 40

Disable the radial distance range filter and set up the bounderies to min 0mm and max 200000mm.

Table 150: Example: sWN LFPradialDistanceRangeFilter

CoLa A	<STX>sWN{SPC}LFPradialDistanceRangeFilter{SPC}0{SPC}0{SPC}30D40<ETX>	
	<STX>sWN LFPradialDistanceRangeFilter 0 0 30D40<ETX>	
	sWN LFPradialDistanceRangeFilter 0 0 30D40	
	02 73 57 4E 20 4C 46 50 72 61 64 69 61 6C 44 69 73 74 61 6E 63 65 52 61 6E 67 65 46 69 6C 74 65 72 20 00 20 00 20 30D40 03	
CoLa B	02 02 02 02 00 00 00 2A 73 57 4E 20 4C 46 50 72 61 64 69 61 6C 44 69 73 74 61 6E 63 65 52 61 6E 6765 46 69 6C 74 65 72 20 00 00 00 00 00 00 03 0D 40 31	
	73 57 4E 20 4C 46 50 72 61 64 69 61 6C 44 69 73 74 61 6E 63 65 52 61 6E 67 65 46 69 6C 74 65 72 20 00 00 00 00 00 00 03 0D 40	

Table 151: Telegram structure: sWA LFPradialDistanceRangeFilter

Telegram structure: sWA LFPradialDistanceRangeFilter						
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	Restriction of the scan(s) to a specified distance range.	String	28		LFPradialDis-tanceRangeFil-ter	4C 46 50 72 61 64 69 61 6C 44 69 73 74 61 6E 63 65 52 61 6E 67 65 46 69 6C 74 65 72

Table 152: Example: sWA LFPradialDistanceRangeFilter

CoLa A	<STX>sWA{SPC}LFPradialDistanceRangeFilter<ETX>
	02 73 57 41 20 4C 46 50 72 61 64 69 61 6C 44 69 73 74 61 6E 63 65 52 61 6E 67 65 46 69 6C 74 65 72 03
CoLa B	02 02 02 02 00 00 00 21 73 57 41 20 4C 46 50 72 61 64 69 61 6C 44 69 73 74 61 6E 63 65 52 61 6E 67 65 46 69 6C 74 65 72 20 70

13.4.1.4.6 Inputs and Outputs

13.4.1.4.6.1 Read state of the ports [sRN LIDportstate]

LIDportstate has to be available additionally or as successor of the LIDoutputstate telegram.

Valid for all sensors with Ethernet and ports (inputs / outputs).

Table 153: Telegram structure: sRN LIDportstate

Telegram structure: sRN LIDportstate						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E
Command	Ask for port configuration	String	12		LIDportstate	4C 49 44 70 6F 72 74 73 74 61 74 65

Table 154: Example: sRN LIDportstate

CoLa A	<STX>sRN{SPC}LIDportstate<ETX>	
	<STX>sRN LIDportstate<ETX>	
	sRN LIDportstate	
	02 73 52 4E 20 4C 49 44 70 6F 72 74 73 74 61 74 65 03	
CoLa B	02 02 02 02 00 00 00 10 73 52 4E 20 4C 49 44 70 6F 72 74 73 74 61 74 65 60	
	73 52 4E 20 4C 49 44 70 6F 72 74 73 74 61 74 65	

Table 155: Telegram structure: sRA LIDportstate

Telegram structure: sRA LIDportstate						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sRA	73 52 41
Command	Port state	String	12		LIDportstate	4C 49 44 70 6F 72 74 73 74 61 74 65
Status code	Version number	Uint_16	2	Current version:	0 ... FFFFh 0	00 01 ... FF FF
	System counter (time in µs since power up max. 71min then starting from 0 again)	Uint_32	4		0 ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
ARRAY which defines the number of internal ports*	0...n	Uint_16	2	Hex: Not available: Number of ports:	0000 - FFFF 00 01 ... n	00 00 - FF FF

Telegram structure: sRA LIDportstate						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
State of the ports and count value in hex	Internal port state	Enum_8	1	Output voltage low: (Relays open) Output voltage high: (Relays closed) Tri-state: Input voltage high (level): Input voltage from low to high (edge) Input voltage low (level): Input voltage high to low (edge)	00 01 02 03 04	00 01 02 03 04
	Internal port counter	Uint_32	4		0 ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
	.....					
ARRAY which defines the number of external or **virtual** ports*	0...n	Uint_16	1	Hex: Not available: Numer of ports:	00 00 - FF FF 00 01 ... n	00 00 - FF FF
State of the ports and count value in hex	External port state	Enum_8	1	Output voltage low: (Relays open) Output voltage high: (Relays closed) Tri-state: Input voltage high (level): Input voltage from low to high (edge) Input voltage low (level): Input voltage high to low (edge)	00 01 02 03 04	00 01 02 03 04
	External port counter	Uint_32	4		0 ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
Time	States code	Enum_16	1	No time data: Time data:	00 00 00 01	00 00 00 01
Time Block (sensor time from the last change of min. one of the outputs)	Year	Array	2	E.g.	1970	07 B2
	Month		1		1 ... 12	01 ... 0C
	Day		1		1 ... 31	01 ... 1F
	Hour		1		0 ... 23	00 ... 17
	Minute		1		0 ... 59	00 ... 3B
	Second		1		0 ... 59	00 ... 3B
	Microsecond		4		0 ... 999999	00 00 00 00 ... 00 0F 42 3F

Inputs/outputs: If the device has separate inputs and outputs (instead of general purpose ports) the ARRAY shall start with inputs followed by the outputs.

Virtual ports are ports that can be used to expand the number of ports but are not physically available. They just show up in the corresponding ethernet telegrams (like LIDportstate).

Tri-State: Port is neither input nor output; the port is set inactive in SOPAS

Table 156: Example: sRA LIDportstate

CoLa A	<STX>sRA{SPC}LIDport-state{SPC}0{SPC}3340BBDA{SPC}1{SPC}0{SPC}1{SPC}0{SPC}4{SPC}0{SPC}2{SPC}0{SPC}2{SPC}0{SPC}2{SPC}0{SPC}2{SPC}0{SPC}2{SPC}0{SPC}2{SPC}0{SPC}2{SPC}0{SPC}2{SPC}0{SPC}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}7B2{SPC}1{SPC}1{SPC}2{SPC}25{SPC}1D{SPC}C6F98<ETX>
	02 73 52 41 20 4C 49 44 70 6F 72 74 73 74 61 74 65 20 30 20 33 33 34 30 42 42 44 41 20 31 20 30 20 31 20 30 20 34 20 30 20 32 20 30 20 32 20 30 20 32 20 30 20 32 20 30 20 32 20 30 20 32 20 30 20 32 20 30 20 32 20 30 20 32 20 30 20 32 20 30 20 32 20 30 20 32 20 30 20 32 20 30 20 32 20 30 20 32 20 30 20 31 20 37 42 32 20 31 20 31 20 32 20 32 35 20 31 44 20 43 36 46 39 38 03
CoLa B	02 02 02 02 00 00 00 74 73 52 41 20 4C 49 44 70 6F 72 74 73 74 61 74 65 20 00 00 33 40 BB DA 01 00 00 00 00 01 00 00 00 00 04 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 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 02 00 00 00 00 02 00 00 00 00 02 00 00 00 00 02 00 00 00 00 01 07 B2 01 01 02 25 1D 00 0C 6F 98 2E

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

Table 157: 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
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 158: 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 159: 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

Telegram structure: sRA PortConfiguration						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
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
Sensitivity	Status change at Edge or Level	Enum_8	1	Edge: Level:	0 1	00 01
Reserved	Reserved value 1	Uint_16	2		0	00 00
Reserved	Reserved value 2	Uint_16	2		0	00 00
Output Settings						
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: Input:	0 1 2	00 01 02
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
Restart input	[Only with restart type = Input], input for restart	Uint_16	2		1 ... 8	00 01 ... 00 08
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

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

Table 160: Example: sRA ProtConfiguration

CoLa A	<STX>sRA{SPC}PortConfiguration{SPC}1{SPC}6{SPC}InOut1{SPC}0{SPC}A{SPC}1{SPC}0{SPC}0{SPC}1{SPC}0{SPC}0{SPC}C8{SPC}1{SPC}1{SPC}0{SPC}0{SPC}1{SPC}DRDY{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}1{SPC}6{SPC}InOut2{SPC}0{SPC}A{SPC}1{SPC}0{SPC}0{SPC}1{SPC}0{SPC}0{SPC}C8{SPC}1{SPC}1{SPC}0{SPC}0{SPC}1{SPC}DRDY{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}6{SPC}InOut3{SPC}0{SPC}A{SPC}1{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}C8{SPC}1{SPC}1{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}6{SPC}InOut4{SPC}0{SPC}A{SPC}1{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}C8{SPC}1{SPC}1{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}6{SPC}InOut5{SPC}0{SPC}A{SPC}1{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}C8{SPC}1{SPC}1{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}6{SPC}InOut6{SPC}0{SPC}A{SPC}1{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}C8{SPC}1{SPC}1{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}1{SPC}6{SPC}InOut7{SPC}0{SPC}A{SPC}1{SPC}0{SPC}0{SPC}1{SPC}0{SPC}C8{SPC}1{SPC}1{SPC}0{SPC}0{SPC}1{SPC}DRDY{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}1{SPC}6{SPC}InOut8{SPC}0{SPC}A{SPC}1{SPC}0{SPC}0{SPC}1{SPC}0{SPC}0{SPC}0{SPC}C8{SPC}1{SPC}1{SPC}0{SPC}0{SPC}1{SPC}DRDY{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0<ETX>
	<STX>sRA PortConfiguration 1 6 InOut1 0 A 1 0 0 1 0 0 C8 1 1 0 0 1 DRDY 0 0 0 0 0 0 0 1 6 InOut2 0 A 1 0 0 1 0 0 C8 1 1 0 0 1 DRDY 0 0 0 0 0 0 0 0 6 InOut3 0 A 1 0 0 0 0 0 C8 1 1 0 0 0 0 0 0 0 0 6 InOut4 0 A 1 0 0 0 0 0 C8 1 1 0 0 0 0 0 0 0 0 6 InOut5 0 A 1 0 0 0 0 0 C8 1 1 0 0 0 0 0 0 0 0 6 InOut6 0 A 1 0 0 0 0 0 C8 1 1 0 0 0 0 0 0 0 1 6 InOut7 0 A 1 0 0 1 0 0 C8 1 1 0 0 1 DRDY 0 0 0 0 0 0 0 1 6 InOut8 0 A 1 0 0 1 0 0 C8 1 1 0 0 1 DRDY 0 0 0 0 0 0 0<ETX>
	sRA PortConfiguration 1 6 InOut1 0 A 1 0 0 1 0 0 C8 1 1 0 0 1 DRDY 0 0 0 0 0 0 0 1 6 InOut2 0 A 1 0 0 1 0 0 C8 1 1 0 0 1 DRDY 0 0 0 0 0 0 0 0 6 InOut3 0 A 1 0 0 0 0 0 C8 1 1 0 0 0 0 0 0 0 0 6 InOut4 0 A 1 0 0 0 0 0 C8 1 1 0 0 0 0 0 0 0 0 6 InOut5 0 A 1 0 0 0 0 0 C8 1 1 0 0 0 0 0 0 0 0 6 InOut6 0 A 1 0 0 0 0 0 C8 1 1 0 0 0 0 0 0 0 1 6 InOut7 0 A 1 0 0 1 0 0 C8 1 1 0 0 1 DRDY 0 0 0 0 0 0 0 1 6 InOut8 0 A 1 0 0 1 0 0 C8 1 1 0 0 1 DRDY 0 0 0 0 0 0 0

CoLa B	02 02 02 02 00 00 01 7A 73 52 41 20 50 6F 72 74 43 6F 6E 66 69 67 75 72 61 74 69 6F 6E 20 01 00 06 49 6E 4F 75 74 31 00 00 0A 01 00 00 00 00 01 00 00 00 00 00 C8 00 01 01 00 00 00 00 00 01 44 52 44 59 00 00 00 00 00 00 00 00 00 00 00 01 00 06 49 6E 4F 75 74 32 00 00 0A 01 00 00 00 00 01 00 00 00 00 00 C8 00 01 01 00 00 00 00 00 01 44 52 44 59 00 00 00 00 00 00 00 00 00 00 00 00 06 49 6E 4F 75 74 33 00 00 0A 01 00 00 00 00 00 00 00 00 00 00 C8 00 01 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 06 49 6E 4F 75 74 34 00 00 0A 01 00 00 00 00 00 00 00 00 00 00 C8 00 01 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 06 49 6E 4F 75 74 35 00 00 0A 01 00 00 00 00 00 00 00 00 00 00 C8 00 01 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 06 49 6E 4F 75 74 36 00 00 0A 01 00 00 00 00 00 00 00 00 00 00 C8 00 01 01 00 00 00 00 00 00 00 00 00 00 00 00 01 00 06 49 6E 4F 75 74 37 00 00 0A 01 00 00 00 00 01 00 00 00 00 00 C8 00 01 01 00 00 00 00 00 01 44 52 44 59 00 00 00 00 00 00 00 00 00 00 01 00 06 49 6E 4F 75 74 38 00 00 0A 01 00 00 00 00 01 00 00 00 00 00 C8 00 01 01 00 00 00 00 00 01 44 52 44 59 00 00 00 00 00 00 00 00 00 00 01
	73 52 41 20 50 6F 72 74 43 6F 6E 66 69 67 75 72 61 74 69 6F 6E 20 01 00 06 49 6E 4F 75 74 31 00 00 0A 01 00 00 00 00 01 00 00 00 00 00 C8 00 01 01 00 00 00 00 00 01 44 52 44 59 00 00 00 00 00 00 00 00 00 00 00 01 00 06 49 6E 4F 75 74 32 00 00 0A 01 00 00 00 00 01 00 00 00 00 00 C8 00 01 01 00 00 00 00 00 01 44 52 44 59 00 00 00 00 00 00 00 00 00 00 00 00 06 49 6E 4F 75 74 33 00 00 0A 01 00 00 00 00 00 00 00 00 00 00 C8 00 01 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 06 49 6E 4F 75 74 34 00 00 0A 01 00 00 00 00 00 00 00 00 00 00 C8 00 01 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 06 49 6E 4F 75 74 35 00 00 0A 01 00 00 00 00 00 00 00 00 00 00 C8 00 01 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 06 49 6E 4F 75 74 36 00 00 0A 01 00 00 00 00 00 00 00 00 00 00 C8 00 01 01 00 00 00 00 00 00 00 00 00 00 00 00 01 00 06 49 6E 4F 75 74 37 00 00 0A 01 00 00 00 00 01 00 00 00 00 00 C8 00 01 01 00 00 00 00 00 01 44 52 44 59 00 00 00 00 00 00 00 00 00 00 01 00 06 49 6E 4F 75 74 38 00 00 0A 01 00 00 00 00 01 00 00 00 00 00 C8 00 01 01 00 00 00 00 00 01 44 52 44 59 00 00 00 00 00 00 00 00 00 00

13.4.1.4.6.3 Set port configuration [sWN PortConfiguration]

Configuration of the given ports. Telegram structure represents the configuration of 1 port. If the device has multiple ports, use the same structure.

Table 161: Telegram structure: sWN PortConfiguration

Telegram structure: sWN PortConfiguration (Required User Level: authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3		sWN	73 57 4E
Command	Configuration of the given ports	String	17		PortConfigura-tion	50 6F 72 74 43 6F 6E 66 69 67 75 72 61 74 69 6F 6E
Port Type	Input or Output	Enum_8	1	Input: Output:	+0d (0h) +1d (1h)	00 01
Name	Name of the port	FlexString	9 (0..32)	Default:	InOut1	00 06 49 6E 4F 75 74 31 00
Input Settings						
Logic	Logic of the input	Enum_8	1	Active high: Active low:	+0d (0h) +1d (1h)	00 01
Debounce	Select debouncing time in ms	Uint_8	1		+0d ... +255d (0h ... FFh)	00 ... FF
Sensitivity	Status change at Edge or Level	Enum_8	1	Edge: Level:	+0d (0h) +1d (1h)	00 01
Reserved1		Uint_16	2		+0d (0h)	00 00
Reserved2		Uint_16	2		+0d (0h)	00 00
Output Settings						

Telegram structure: sWN PortConfiguration (Required User Level: authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Logic	Definition of the output logic	Enum_8	1	Active high: Active low:	+0d (0h) +1d (1h)	00 01
Output Mode	Set kind of mode for output pin	Enum_8	1	PNP: NPN: Push-Pull:	0 1 2	00 01 02
Restart Type	Defines type of restart to be used	Enum_8	1	Immediately: Time: :	+0d (0h) +1d (1h) +2d (2h)	00 01 02
Restart Time	[Only with restart type = Time], time in ms	Uint_32	4	(20 ms ... 600,000 ms)	+20d ... +600000d (C8h ... 927C0h)	00 00 00 00 ... 00 09 27 C0
Restart Input	[Only with restart type = Input], input for restart	Uint_16	2		+1d ... +8d (1h ... 8h)	00 00 ... 00 08
Combina-tion	Combining multiple Events and/or Inputs	Enum_8	1	AND: OR: XOR:	+0d (0h) +1d (1h) +2d (2h)	
Reserved	Reserved value 3	Uint_16	2		+0d (0h)	00 00
Reserved	Reserved value 4	Uint_16	2		+0d (0h)	00 00
	Source					
Source	The source parameter are only existing if the port is set to OUTPUT!		2		+1d (1h)	00 01
Source Name	Name of the source option	String	4	DeviceNotReady: Input1: Input2: SopasCommand:	DRDY IN01 IN02 SC01	44 52 44 59 49 4E 30 31 49 4E 30 32 53 43 30 31
Invert	Invert the source signal	Bool_1	1	Not inverted: Inverted:	+0d (0h) +1d (1h)	00 01
Reserved	Reserved value 5	Uint_8	1		+0d (0h)	00
Reserved	Reserved value 6	Uint_8	1		+0d (0h)	00
				Reserved		
Reserved	Reserved value 7	Uint_16	2		+0d (0h)	00 00
Reserved	Reserved value 8	Uint_16	2		+0d (0h)	00 00
Reserved	Reserved value 9	Uint_16	2		+0d (0h)	00 00
Reserved	Reserved value 10	Uint_16	2		+0d (0h)	00 00

Table 162: Example: sWN ProtConfiguration

CoLa A	<STX>sWN{SPC}PortConfiguration{SPC}1{SPC}6{SPC}InOut1{SPC}0{SPC}A{SPC}1{SPC}0{SPC}0{SPC}1{SPC}0{SPC}0{SPC}C8{SPC}1{SPC}1{SPC}0{SPC}0{SPC}1{SPC}DRDY{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}1{SPC}6{SPC}InOut2{SPC}0{SPC}A{SPC}1{SPC}0{SPC}0{SPC}1{SPC}0{SPC}0{SPC}C8{SPC}1{SPC}1{SPC}0{SPC}0{SPC}1{SPC}DRDY{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}6{SPC}InOut3v0{SPC}A{SPC}1{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}C8{SPC}1{SPC}1{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0<ETX>	
	<STX>sWN PortConfiguration 1 6 InOut1 0 A 1 0 0 1 0 0 C8 1 1 0 0 1 DRDY 0 0 0 0 0 0 0 1 6 InOut2 0 A 1 0 0 1 0 0 C8 1 1 0 0 1 DRDY 0 0 0 0 0 0 0 0 6 InOut3 0 A 1 0 0 0 0 0 C8 1 1 0 0 0 0 0 0 0<ETX>	
	sWN PortConfiguration 1 6 InOut1 0 A 1 0 0 1 0 0 C8 1 1 0 0 1 DRDY 0 0 0 0 0 0 0 1 6 InOut2 0 A 1 0 0 1 0 0 C8 1 1 0 0 1 DRDY 0 0 0 0 0 0 0 0 6 InOut3 0 A 1 0 0 0 0 0 C8 1 1 0 0 0 0 0 0 0	
	02 73 57 4E 20 50 6F 72 74 43 6F 6E 66 69 67 75 72 61 74 69 6F 6E 20 31 20 36 20 49 6E 4F 75 74 31 20 30 20 41 20 31 20 30 20 30 20 31 20 30 20 30 20 43 38 20 31 20 31 20 30 20 30 20 31 20 44 52 44 59 20 30 20 30 20 30 20 30 20 30 20 30 20 30 20 30 20 31 20 36 20 49 6E 4F 75 74 32 20 30 20 41 20 31 20 30 20 30 20 31 20 30 20 30 20 43 38 20 31 20 31 20 30 20 30 20 31 20 44 52 44 59 20 30 20 30 20 30 20 30 20 30 20 30 20 30 20 30 20 30 20 36 20 49 6E 4F 75 74 33 20 30 20 41 20 31 20 30 20 30 20 30 20 30 20 30 20 30 20 43 38 20 31 20 31 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 9F 73 57 4E 20 50 6F 72 74 43 6F 6E 66 69 67 75 72 61 74 69 6F 6E 20 01 00 06 49 6E 4F 75 74 31 00 00 0A 01 00 00 00 00 01 00 00 00 00 00 C8 00 01 01 00 00 00 00 00 01 44 52 44 59 00 00 00 00 00 00 00 00 00 00 01 00 06 49 6E 4F 75 74 32 00 00 0A 01 00 00 00 00 01 00 00 00 00 00 C8 00 01 01 00 00 00 00 00 01 44 52 44 59 00 00 00 00 00 00 00 00 00 00 00 00 06 49 6E 4F 75 74 33 00 00 0A 01 00 00 00 00 00 00 00 00 00 00 C8 00 01 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 9F	
	73 57 4E 20 50 6F 72 74 43 6F 6E 66 69 67 75 72 61 74 69 6F 6E 20 01 00 06 49 6E 4F 75 74 31 00 00 0A 01 00 00 00 00 01 00 00 00 00 00 C8 00 01 01 00 00 00 00 00 01 44 52 44 59 00 00 00 00 00 00 00 00 00 00 01 00 06 49 6E 4F 75 74 32 00 00 0A 01 00 00 00 00 01 00 00 00 00 00 C8 00 01 01 00 00 00 00 00 01 44 52 44 59 00 00 00 00 00 00 00 00 00 00 00 00 06 49 6E 4F 75 74 33 00 00 0A 01 00 00 00 00 00 00 00 00 00 00 C8 00 01 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00	

Table 163: Telegram structure: sWA PortConfiguration

Telegram structure: sWA PortConfiguration						
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	Configuration of the given ports	String	17		PortConfigura-tion	50 6F 72 74 43 6F 6E 66 69 67 75 72 61 74 69 6F 6E

Table 164: Example: sWA PortConfiguration

CoLa A	<STX>sWA{SPC}PortConfiguration<ETX>
	02 73 57 41 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 16 73 57 41 20 50 6F 72 74 43 6F 6E 66 69 67 75 72 61 74 69 6F 6E 20 0C

13.4.1.4.6.4 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 128.

Table 165: 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 166: 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 167: 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 168: Example: sRA LIDinputstate In1 not used, In2 inactive, In3 inactive, In4 not used, In5 not used, In6 not used, In7 not used, In8 not used, time: 1970-01-01 0:13 58.443 sec

CoLa A	<STX>sRA{SPC}LIDinput-state{SPC}1{SPC}31F99C10{SPC}2{SPC}0{SPC}0{SPC}2{SPC}2{SPC}2{SPC}2{SPC}1{SPC}7B2{SPC}1{SPC}1{SPC}0{SPC}D{SPC}3A{SPC}6C278<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 32 20 30 20 30 20 32 20 32 20 32 20 32 20 32 20 31 20 37 42 32 20 31 20 31 20 30 20 44 20 33 41 20 36 43 32 37 38 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 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 31 F9 9C 10 02 00 00 02 02 02 02 00 01 07 B2 01 01 00 0D 3A 00 06 C2 78 5A

13.4.1.4.6.5 Read state of the outputs [sRN LIDoutputstate]

Status of all outputs

Table 169: 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 170: 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 171: 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 128.						

13.4.1.4.6.6 Receive outputstate by event [sEN LIDoutputstate]

Output telegram is sent every time an output state changes.

Table 172: Telegram structure: sEN LIDoutputstate

Telegram structure: sEN LIDoutputstate						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Event	String	3		sEN	73 45 4E
Command	Output state	String	14		LIDoutputstate	4C 49 44 6F 75 74 70 75 74 73 74 61 74 65
	Start/stop	Enum_8	1	Start: Stop:	1 0	01 00

Table 173: 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 174: 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 175: Example: sRA LIDoutputstate Out1 active Count 0, Out2 active Count 0, all other Outputs not used, time: 1970-01-01 0:31 38.191 sec

CoLa A	<STX>sRA{SPC}LIDoutput-state{SPC}1{SPC}712414B0{SPC}1{SPC}0{SPC}1{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}0SPC}1{SPC}7B2{SPC}1{SPC}1{SPC}0{SPC}1F{SPC}26{SPC}2EA18<ETX>
	02 73 52 41 20 4C 49 44 6F 75 74 70 75 74 73 74 61 74 65 20 31 20 37 31 32 34 31 34 42 30 20 31 20 30 20 31 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 42 32 20 31 20 31 20 30 20 31 46 20 32 36 20 32 45 41 31 38 03
CoLa B	02 02 02 02 00 00 00 4E 73 52 41 20 4C 49 44 6F 75 74 70 75 74 73 74 61 74 65 20 00 01 71 24 14 B0 01 00 00 00 00 01 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 B2 01 01 00 1F 26 00 02 EA 18 C4

13.4.1.4.6.7 Set output state [sMN mDOSetOutput]

NOTE
Output source needs to be set to "SOPAS command" and the port configured as Output (in case of I/O).

Table 176: 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 number		Uint_8	1	Depends on system plug:	1 ... 3 1 ... 6	01 ... 03 01 ... 06
Output state		Enum_8	1	Not active: Active:	0 1	00 01

Table 177: 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 178: 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 179: 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

13.4.1.4.7 Status

13.4.1.4.7.1 Read firmware version [sRN DeviceIdent]

Table 180: 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 181: 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 182: Telegram structure: sRA DeviceIdent

Telegram structure: sRA DeviceIdent						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sRA	73 52 41
Command		String	11		DeviceIdent	44 65 76 69 63 65 49 64 65 6E 74
Value	Length of firmware designation	Enum_16	2		0 ... 22h	0 ... 22h
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 183: Example: sRA DeviceIdent

CoLa A	<STX>sRA{SPC}DeviceIdent{SPC}8{SPC}picoScan{SPC}8{SPC}1.2.0.0B<ETX>
	02 73 52 41 20 44 65 76 69 63 65 49 64 65 6E 74 20 38 20 70 69 63 6F 53 63 61 6E 20 38 20 31 2E 32 2E 30 2E 30 42 03
CoLa B	02 02 02 02 00 00 00 24 73 52 41 20 44 65 76 69 63 65 49 64 65 6E 74 20 00 08 70 69 63 6F 53 63 61 6E 00 08 31 2E 32 2E 30 2E 30 42 4F

13.4.1.4.7.2 Read version of the application software [sRN FirmwareVersion]

Table 184: Telegram structure: sRN FirmwareVersion

Telegram structure: sRN FirmwareVersion						
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 version of the application software	String	15		FirmwareVersion	46 69 72 6D 77 61 72 65 56 65 72 73 69 6F 6E

Table 185: Example: sRN FirmwareVersion

CoLa A	<STX>sRN{SPC}FirmwareVersion<ETX>	
	<STX>sRN FirmwareVersion<ETX>	
	sRN FirmwareVersion	
	02 73 52 4E 20 46 69 72 6D 77 61 72 65 56 65 72 73 69 6F 6E 03	
CoLa B	02 02 02 02 00 00 00 13 73 52 4E 20 46 69 72 6D 77 61 72 65 56 65 72 73 69 6F 6E 24	
	73 52 4E 20 46 69 72 6D 77 61 72 65 56 65 72 73 69 6F 6E	

Table 186: Telegram structure: sRA FirmwareVersion

Telegram structure: sRA FirmwareVersion						
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 version of the application software	String	15		FirmwareVersion	46 69 72 6D 77 61 72 65 56 65 72 73 69 6F 6E
Value	Length of version	Uint_16	2		0 ... 28h	0 ... 28h
Value	Version	String	16		(See example)	(See example)

Table 187: Example: sRA FirmwareVersion

CoLa A	<STX>sRA{SPC}FirmwareVersion{SPC}14{SPC}1.2.0-b.0+1225.523ef<ETX>
	02 73 52 41 20 46 69 72 6D 77 61 72 65 56 65 72 73 69 6F 6E 20 31 34 20 31 2E 32 2E 30 2D 62 2E 30 2B 31 32 32 35 2E 35 32 33 65 66 03
CoLa B	02 02 02 02 00 00 00 2A 73 52 41 20 46 69 72 6D 77 61 72 65 56 65 72 73 69 6F 6E 20 00 14 31 2E 32 2E 30 2D 62 2E 30 2B 31 32 32 35 2E 35 32 33 65 66 4B

13.4.1.4.7.3 Read the device state [sRN SCdevicestate]

This telegram reads the general device state.

Table 188: 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 189: 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 190: 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 191: 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

13.4.1.4.7.4 Read device order number [sRN OrdNum]

This telegram reads the device order number which corresponds to the SICK part number of the device.

Table 192: Telegram structure: sRN OrdNum

Telegram structure: sRN OrdNum						
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		OrdNum	4F 72 64 4E 75 6D

Table 193: Example: sRN OrdNum

CoLa A	<STX>sRN{SPC}OrdNum<ETX>	
	<STX>sRN OrdNum<ETX>	
	sRN OrdNum	
	02 73 52 4E 20 4F 72 64 4E 75 6D 03	
CoLa B	02 02 02 02 00 00 00 0A 73 52 4E 20 4F 72 64 4E 75 6D 40	
	73 52 4E 20 4F 72 64 4E 75 6D	

Table 194: Telegram structure: sRA OrdNum

Telegram structure: sRA OrdNum						
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		OrdNum	4F 72 64 4E 75 6D
Length	Number of characters of the following order number	Uint_16	2		0h ... 20h	00 00 ... 00 20

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

Example: sRA OrdNum 1134610 (Order Number for picoScan150 Pro-1)

Table 195: Example for picoScan150 Pro-1: sRA OrdNum

CoLa A	<STX>sRA{SPC}OrdNum{SPC}7{SPC}1134610<ETX>
	02 73 52 41 20 4F 72 64 4E 75 6D 20 37 20 31 31 33 34 36 31 30 03
CoLa B	02 02 02 02 00 00 00 14 73 52 41 20 4F 72 64 4E 75 6D 20 00 07 31 31 33 34 36 31 30 58

13.4.1.4.7.5 Read serial number [sRN SerialNumber]

Table 196: Telegram structure: sRN SerialNumber

Telegram structure: sRN SerialNumber						
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 serial number of the device	String	12		SerialNumber	53 65 72 69 61 6C 4E 75 6D 62 65 72

Table 197: Example: sRN SerialNumber

CoLa A	<STX>sRN{SPC}SerialNumber<ETX>
	<STX>sRN SerialNumber<ETX>
	sRN SerialNumber
	02 73 52 4E 20 53 65 72 69 61 6C 4E 75 6D 62 65 72 03
CoLa B	02 02 02 02 00 00 00 10 73 52 4E 20 53 65 72 69 61 6C 4E 75 6D 62 65 72 4C
	73 52 4E 20 53 65 72 69 61 6C 4E 75 6D 62 65 72

Table 198: Telegram structure: sRA SerialNumber

Telegram structure: sRA SerialNumber						
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 serial number of the device	String	12		SerialNumber	53 65 72 69 61 6C 4E 75 6D 62 65 72
Length of serial number	Number of characters of the serial number	Uint_16	2		0 ... 8h	00 00 ... 00 08h

Telegram structure: sRA SerialNumber						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Serial number	Production period (year, calendar week, number): YYWWxxxx	String	8		(See example)	(See example)

Table 199: Example: sRA SerialNumber

CoLa A	<STX>sRA{SPC}SerialNumber{SPC}8{SPC}23360024<ETX>
	02 73 52 41 20 53 65 72 69 61 6C 4E 75 6D 62 65 72 20 38 20 32 33 33 36 30 30 32 34 03
CoLa B	02 02 02 02 00 00 00 1B 73 52 41 20 53 65 72 69 61 6C 4E 75 6D 62 65 72 20 00 08 32 33 33 36 30 30 32 34 69

13.4.1.4.7.6 Read device type [sRN DItype]

This telegram asks for the device type.

Table 200: 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 201: 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 202: 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 ... FF)	00 ... FF
Device type	Type code of the device	String	(var.)		(Device type)	(Device type)

Table 203: sRA DItype Example for picoScan150:

CoLa A	<STX>sRA{SPC}DItype{SPC}B{SPC}picoScan150<ETX>
	02 73 52 41 20 44 49 74 79 70 65 20 42 20 70 69 63 6F 53 63 61 6E 31 35 30 03
CoLa B	02 02 02 02 00 00 00 18 73 52 41 20 44 49 74 79 70 65 20 00 0B 70 69 63 6F 53 63 61 6E 31 35 30 60

13.4.1.4.7.7 Read operating hours [sRN ODoprh]

Views the total number of operating hours during the lifetime of the device.

Table 204: 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 205: 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 206: 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 207: Example: sRA ODoprh

CoLa A	<STX>sRA{SPC}ODoprh{SPC}1B50B<ETX>
	02 73 52 41 20 4F 44 6F 70 72 68 20 31 42 35 30 42 03
CoLa B	02 02 02 02 00 00 00 0F 73 52 41 20 4F 44 6F 70 72 68 20 00 01 B5 0B D1

Calculation of the value: 1B50B (hex) → 111883 (dez) × 1/10 h = 11188.3 h

13.4.1.4.7.8 Read operating hours since last power on [sRN ODopdaily]

Views the runtime duration since the last power on of the device.

Table 208: Telegram structure: sRN ODopdaily

Telegram structure: sRN ODopdaily						
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	9		ODopdaily	4F 44 6F 70 64 61 69 6C 79

Table 209: Example: sRN ODopdaily

CoLa A	<STX>sRN{SPC}ODopdaily<ETX>	
	<STX>sRN ODopdaily<ETX>	
	sRN ODopdaily	
	02 73 52 4E 20 4F 44 6F 70 64 61 69 6C 79 03	
CoLa B	02 02 02 02 00 00 00 0D 73 52 4E 20 4F 44 6F 70 64 61 69 6C 79 22	
	73 52 4E 20 4F 44 6F 70 64 61 69 6C 79	

Table 210: Telegram structure: sRA ODopdaily

Telegram structure: sRA ODopdaily						
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 since last power on	String	9		ODopdaily	4F 44 6F 70 64 61 69 6C 79
Value	Operating hours in 1/10 h	Uint_32	4		0h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF

Table 211: Example: sRA ODopdaily

CoLa A	<STX>sRA{SPC}ODopdaily{SPC}424772B8<ETX>
	02 73 52 41 20 4F 44 6F 70 72 68 20 34 32 34 37 37 32 42 38 03
CoLa B	02 02 02 02 00 00 00 12 73 52 41 20 4F 44 6F 70 72 68 20 42 47 72 B8 D7

13.4.1.4.7.9 Read power on counter [sRN ODpwrc]

Table 212: 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 213: 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 214: 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 215: Example: sRA ODpwrc

CoLa A	<STX>sRA{SPC}ODpwrc{SPC}9A<ETX>
	02 73 52 41 20 4F 44 70 77 72 63 20 39 41 03
CoLa B	02 02 02 02 00 00 00 0F 73 52 41 20 4F 44 70 77 72 63 20 00 00 00 9A E7

13.4.1.4.7.10 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 216: 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 217: 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 218: 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
Tempera-ture 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)

The result is float and IEEE-754 coded

Table 219: 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

13.4.1.4.7.11 Set device name [sWN LocationName]

Table 220: 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 221: Example: sWN LocationName +9 LongRange

CoLa A	<STX>sWN{SPC}LocationName{SPC}+9{SPC}LongRange<ETX>	
	<STX>sWN LocationName +9 LongRange<ETX>	
	sWN LocationName +9 LongRange	
	02 73 57 4E 20 4C 6F 63 61 74 69 6F 6E 4E 61 6D 65 20 39 20 4C 6F 6E 67 52 61 6E 67 65 03	
CoLa B	02 02 02 02 00 00 00 1D 73 57 4E 20 4C 6F 63 61 74 69 6F 6E 4E 61 6D 65 20 00 09 4C 6F 6E 67 52 61 6E 67 65 2C	
	73 57 4E 20 4C 6F 63 61 74 69 6F 6E 4E 61 6D 65 20 00 09 4C 6F 6E 67 52 61 6E 67 65	

Table 222: Telegram structure: sWA LocationName

Telegram structure: sWA LocationName						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sWA	73 57 41
Command	Set device name	String	12		LocationName	4C 6F 63 61 74 69 6F 6E 4E 61 6D 65

Table 223: 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

13.4.1.4.7.12 Read device name [sRN LocationName]

Table 224: 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 225: 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 226: 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 220, page 140						

13.4.1.4.7.13 Reset output counter [sMN LIDrstoutpcnt]

Table 227: 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 228: 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 229: 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 230: 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

13.4.1.4.7.14 Initiate an acoustic or visual signal for a defined period of time [sMN FindMe]

This command can be used to make the device easier to find.

Table 231: Telegram structure: sMN FindMe

Telegram structure: sMN FindMe						
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	Initiate an acoustic or visual signal	String	11		FindMe	46 69 6E 64 4D 65

Telegram structure: sMN FindMe						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Period of time	Duration in seconds	Uint_16	2		0d ... 65535d (0h .. FF FF)	00 ... FF FF

Table 232: Example: sMN FindMe

CoLa A	<STX>sMN{SPC}FindMe{SPC}1<ETX>
	<STX>sMN FindMe 1<ETX>
	sMN FindMe 1
	02 73 4D 4E 20 46 69 6E 64 4D 65 20 31 03
CoLa B	02 02 02 02 00 00 00 0D 73 4D 4E 20 46 69 6E 64 4D 65 20 01 7C
	73 4D 4E 20 46 69 6E 64 4D 65 20 01

Table 233: Telegram structure: sAN FindMe

Telegram structure: sAN FindMe						
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	Initiate an acoustic or visual signal	String	11		FindMe	46 69 6E 64 4D 65

Table 234: Example: sAN FindMe

CoLa A	<STX>sAN{SPC}Findme<ETX>
	02 73 41 4E 20 46 69 6E 64 4D 65 03
CoLa B	02 02 02 02 00 00 00 0B 73 41 4E 20 46 69 6E 64 4D 65 20 71

13.4.1.4.7.15 Read date of last permanent save [sRN DIpara]

This command reads the date at which the last permanent save (see "Save parameters permanently [sMN mEEwriteall]", page 85) was executed.

Table 235: Telegram structure: sRN DIpara

Telegram structure: sRN DIpara						
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 date of last permanent save	String	6		DIpara	44 49 70 61 72 61

Table 236: Example: sRN DIpara

CoLa A	<STX>sRN{SPC}DIpara<ETX>
	<STX>sRN DIpara<ETX>
	sRN DIpara
	02 73 52 4E 20 44 49 70 61 72 61 03

CoLa B	02 02 02 02 00 00 00 4E 73 52 4E 20 44 49 70 61 72 61 40
	73 52 4E 20 44 49 70 61 72 61

Table 237: Telegram structure: sRA DIpara

Telegram structure: sRA DIpara						
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 date of last permanent save	String	6		DIpara	44 49 70 61 72 61
Value	Number of characters of the following date	Uint_16	2		0d ... 10d (0h ... 0Ah)	0 ... 0A
Date of last permanent save	DD.MM.YYYY	FlexString	10		(see example)	(see example)

Table 238: Example: sRA DIpara

CoLa A	<STX>sRA{SPC}DIpara{SPC}A{SPC}09.01.2024<ETX>
	02 73 52 41 20 44 49 70 61 72 61 20 41 20 30 39 2E 30 31 2E 32 30 32 34 03
CoLa B	02 02 02 02 00 00 00 17 73 52 41 20 44 49 70 61 72 61 20 00 0A 30 39 2E 30 31 2E 32 30 32 34 69

13.4.1.4.7.16 Read time of last permanent save [sRN DIparatm]

This command reads the time at which the last permanent save was executed.

Table 239: Telegram structure: sRN DIparatm

Telegram structure: sRN DIparatm						
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 time of last permanent save	String	8		DIparatm	44 49 70 61 72 61 74 6D

Table 240: Example: sRN DIparatm

CoLa A	<STX>sRN{SPC}DIparatm<ETX>
	<STX>sRN DIparatm<ETX>
	sRN DIparatm
	02 73 52 4E 20 44 49 70 61 72 61 74 6D 03
CoLa B	02 02 02 02 00 00 00 4E 73 52 4E 20 44 49 70 61 72 61 74 6D 59
	73 52 4E 20 44 49 70 61 72 61 74 6D

Table 241: Telegram structure: sRA DIparatm

Telegram structure: sRA DIparatm						
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 time of last permanent save	String	8		DIparatm	44 49 70 61 72 61 74 6D
Value	Number of characters of the following time	Uint_16	2		0d ... 5d (0h ... 5h)	00 00 ... 00 05
Time of last permanent save	-	FlexString	5		(see example)	(see example)

Table 242: Example: sRA DIparatm

CoLa A	<STX>sRA{SPC}DIparatm{SPC}5{SPC}12:28<ETX>
	02 73 52 41 20 44 49 70 61 72 61 74 6D 20 35 20 31 32 3A 32 38 03
CoLa B	02 02 02 02 00 00 00 14 73 52 41 20 44 49 70 61 72 61 74 6D 20 00 05 31 32 3A 32 38 40

13.4.1.4.8 Interfaces

13.4.1.4.8.1 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

13.4.1.4.8.2 Read IP address [sRN EIIpAddr]

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
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

13.4.1.4.8.3 Read IP address assigned by DHCP [sRN EIIpAddrDHCP]

NOTE
DHCP needs to be set as mode for ethernet assignment.

Table 251: Telegram structure: sRN EIIpAddrDHCP

Telegram structure: sRN EIIpAddrDHCP						
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 assigned by DHCP	String	12		EIIpAddrDHCP	45 49 49 70 41 64 64 72 44 48 43 50

Table 252: Example: srN EIIpAddrDHCP

CoLa A	<STX>sRN{SPC}EIIpAddrDHCP<ETX>	
	<STX>sRN EIIpAddrDHCP<ETX>	
	sRN EIIpAddrDHCP	
	02 73 57 4E 20 45 49 49 70 41 64 64 72 44 48 43 50 03	
CoLa B	02 02 02 02 00 00 00 10 73 52 4E 20 45 49 49 70 41 64 64 72 44 48 43 50 56	
	73 52 4E 20 45 49 49 70 41 64 64 72 44 48 43 50	

Table 253: Telegram structure: sRA EIIpAddrDHCP

Telegram structure: sRA EIIpAddrDHCP						
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 assigned by DHCP	String	12		EIIpAddrDHCP	45 49 49 70 41 64 64 72 44 48 43 50
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 254: Example: sRA EIIpAddrDHCP 192.168.0.1

CoLa A	<STX>sRA{SPC}EIIpAddrDHCP{SPC}C0{SPC}A8{SPC}0{SPC}1<ETX>
	02 73 52 41 20 45 49 49 70 41 64 64 72 44 48 43 50 20 43 30 20 41 38 20 30 20 31 03
CoLa B	02 02 02 02 00 00 00 15 73 52 41 20 45 49 49 70 41 64 64 72 44 48 43 50 20 C0 A8 00 01 10

13.4.1.4.8.4 Set mode for ethernet adress assignment [sWN EIAddrMode]

This Command determins the mode for the ethernet adress assignment.

Table 255: Telegram structure: sWN EIAddrMode

Telegram structure: sWN EIAddrMode (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 mode for ethernet adress assignment	String	10		EIAddrMode	45 49 41 64 64 72 4D 6F 64 65
Ethernet adress assign-ment	Static IP adress / DHCP	Enum_8	1	Static: DHCP:	0 1	00 01

Table 256: Example: sWN EIAddrMode

CoLa A	<STX>sWN{SPC}EIAddrMode{SPC}1<ETX>	
	<STX>sWN EIAddrMode 1<ETX>	
	sWN EIAddrMode 1	
	02 73 57 4E 20 45 49 41 64 64 72 4D 6F 64 65 20 31 03	
CoLa B	02 02 02 02 00 00 00 4E 73 57 4E 20 45 49 41 64 64 72 4D 6F 64 65 20 01 76	
	73 57 4E 20 45 49 41 64 64 72 4D 6F 64 65 20 01	

Table 257: Telegram structure: sWA EIAddrMode

Telegram structure: sWA EIAddrMode						
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 mode for ethernet adress assignment	String	10		EIAddrMode	45 49 41 64 64 72 4D 6F 64 65

Table 258: Example: sWA EIAddrMode

CoLa A	<STX>sWA{SPC}EIAddrMode<ETX>
	02 73 57 41 20 45 49 41 64 64 72 4D 6F 64 65 03
CoLa B	02 02 02 02 00 00 00 0F 73 57 41 20 45 49 41 64 64 72 4D 6F 64 65 20 79

13.4.1.4.8.5 Set fallback for DHCP [sWN EIDHCPFallback]

This Command determins the fallback when DHCP is not successful.

Table 259: Telegram structure: sWN EIDHCPFallback

Telegram structure: sWN EIDHCPFallback (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 fallback for DHCP	String	14		EIDHCPFallback	45 49 44 48 43 50 46 61 6C 6C 62 61 63 6B
Fallback ethernet adress assign-ment	Use Static IP adress / Retry DHCP	Enum_8	1	Static IP adress: DHCP retry:	0 1	00 01

Table 260: Example: sWN EIDHCPFallback

CoLa A	<STX>sWN{SPC}EIDHCPFallback{SPC}1<ETX>	
	<STX>sWN EIDHCPFallback 1<ETX>	
	sWN EIDHCPFallback 1	
	02 73 57 4E 20 45 49 44 48 43 50 46 61 6C 6C 62 61 63 6B 20 31 03	
CoLa B	02 02 02 02 00 00 00 14 73 57 4E 20 45 49 44 48 43 50 46 61 6C 6C 62 61 63 6B 20 01 54	
	73 57 4E 20 45 49 44 48 43 50 46 61 6C 6C 62 61 63 6B 20 01	

Table 261: Telegram structure: sWA EIDHCPFallback

Telegram structure: sWA EIDHCPFallback						
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 EIDHCPFallback						
Telegram part	Description	Variable	Length	Additional details	Values CoLa A (ASCII)	Values CoLa B (Binary)
Command	Set fallback for DHCP	String	14		EIDHCPFallback	45 49 44 48 43 50 46 61 6C 6C 62 61 63 6B

Table 262: Example: sWA EIDHCPFallback

CoLa A	<STX>sWA{SPC}EIDHCPFallback<ETX>
	02 73 57 41 20 45 49 44 48 43 50 46 61 6C 6C 62 61 63 6B 03
CoLa B	02 02 02 02 00 00 00 13 73 57 41 20 45 49 44 48 43 50 46 61 6C 6C 62 61 63 6B 20 5A

13.4.1.4.8.6 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 263: 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 264: 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 265: Telegram structure: sWA Elgate

Telegram structure: sWA Elgate						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sWA	73 57 41
Command	Set gateway adress	String	6		Elgate	45 49 67 61 74 65

Table 266: Example: sWA Elgate

CoLa A	<STX>sWA{SPC}Elgate<ETX>
	02 73 57 41 20 45 49 67 61 74 65 03
CoLa B	02 02 02 02 00 00 00 0B 73 57 41 20 45 49 67 61 74 65 20 7E

13.4.1.4.8.7 Read Ethernet gateway [sRN Elgate]

Read for the Ethernet gateway (TCP/IP)

Table 267: Telegram structure: sRN Elgate

Telegram structure: sRN Elgate						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E
Command	Read gateway address	String	6		Elgate	45 49 67 61 74 65

Table 268: Example: sRN Elgate

CoLa A	<STX>sRN{SPC}Elgate<ETX>	
	<STX>sRN Elgate<ETX>	
	sRN Elgate	
	02 73 52 4E 20 45 49 67 61 74 65 03	
CoLa B	02 02 02 02 00 00 00 0A 73 52 4E 20 45 49 67 61 74 65 54	
	73 52 4E 20 45 49 67 61 74 65	

Table 269: Telegram structure: sRA Elgate

Telegram structure: sRA Elgate						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sRA	73 52 41
Command	Read gateway address	String	6		Elgate	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 270: 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

13.4.1.4.8.8 Read ethernet gateway IP adress assigned by DHCP [sRN EIgateDHCP]

NOTE
DHCP needs to be set as mode for ethernet assignment.

Read for the ethernet gateway IP adress which was assigned by DHCP.

Table 271: Telegram structure: sRN EIgateDHCP

Telegram structure: sRN EIgateDHCP						
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 ethernet gateway IP address assigned by DHCP	String	6		EIgateDHCP	45 49 67 61 74 65 44 48 43 50

Table 272: Example: sRN EIgate

CoLa A	<STX>sRN{SPC}EIgateDHCP<ETX>
	<STX>sRN EIgateDHCP<ETX>
	sRN EIgateDHCP
	02 73 52 4E 20 45 49 67 61 74 65 44 48 43 50 03
CoLa B	02 02 02 02 00 00 00 0E 73 52 4E 20 45 49 67 61 74 65 44 48 43 50 4B
	73 52 4E 20 45 49 67 61 74 65 44 48 43 50

Table 273: Telegram structure: sRA EIgateDHCP

Telegram structure: sRA EIgateDHCP						
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 EIgateDHCP						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command	Read ethernet gateway IP address assigned by DHCP	String	6		EIgateDHCP	45 49 67 61 74 65 44 48 43 50
Gateway IP address	Default: 0.0.0.0	Uint_8	1	First part of gateway IP address	0 ...+255d (00...FF)	00 ... FF
				Second part of gateway IP address	0 ...+255d (00...FF)	00 ... FF
				Third part of gateway IP address	0 ...+255d (00...FF)	00 ... FF
				Fourth part of gateway IP address	0 ...+255d (00...FF)	00 ... FF

Table 274: Example: sRA EIgateDHCP 0.0.0.0

CoLa A	<STX>sRA{SPC}EIgateDHCP{SPC}0{SPC}0{SPC}0{SPC}0<ETX>
	02 73 52 41 20 45 49 67 61 74 65 44 48 43 50 20 30 20 30 20 30 20 30 03
CoLa B	02 02 02 02 00 00 00 13 73 52 41 20 45 49 67 61 74 65 44 48 43 50 20 00 00 00 00 64

13.4.1.4.8.9 Set IP mask [sWN EImask]

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 275: 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 276: 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 277: 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 278: 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

13.4.1.4.8.10 Read IP mask [sRN EImask]

Table 279: 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 280: 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 281: 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 282: 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

13.4.1.4.8.11 Read IP mask assigned by DHCP [sRN EImaskDHCP]

NOTE
DHCP needs to be set as mode for ethernet assignment.

Read for the IP mask which was assigned by DHCP.

Table 283: Telegram structure: sRN EImaskDHCP

Telegram structure: sRN EImaskDHCP						
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 assigned by DHCP	String	10		EImaskDHCP	45 49 6D 61 73 6B 44 48 43 50

Table 284: Example: sRN EImaskDHCP

CoLa A	<STX>sRN{SPC}EImaskDHCP<ETX>
	<STX>sRN EImaskDHCP<ETX>
	sRN EImaskDHCP
	02 73 52 4E 20 45 49 6D 61 73 6B 44 48 43 50 03
CoLa B	02 02 02 02 00 00 00 0E 73 52 4E 20 45 49 6D 61 73 6B 44 48 43 50 4B
	73 52 4E 20 45 49 6D 61 73 6B 44 48 43 50

Table 285: Telegram structure: sRA EImaskDHCP

Telegram structure: sRA EImaskDHCP						
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 EImaskDHCP						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command	Read IP mask assigned by DHCP	String	10		EImaskDHCP	45 49 6D 61 73 6B 44 48 43 50
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 286: Example: sRA EImaskDHCP 255.255.255.0

CoLa A	<STX>sRA{SPC}EIgateDHCP{SPC}FF{SPC}FF{SPC}FF{SPC}0<ETX>
	02 73 52 41 20 45 49 6D 61 73 6B 44 48 43 50 20 46 46 20 46 46 20 46 46 20 30 03
CoLa B	02 02 02 02 00 00 00 13 73 52 41 20 45 49 6D 61 73 6B 44 48 43 50 20 FF FF FF 00 98

13.4.1.4.8.12 Read MAC address [sRN EIMacAdr]

Table 287: Telegram structure: sRN EIMacAdr

Telegram structure: sRN EIMacAdr						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E
Command	Read MAC address of the device	String	8		EIMacAdr	45 49 4D 61 63 41 64 72

Table 288: Example: sRN EIMacAdr

CoLa A	<STX>sRN{SPC}EIMacAdr<ETX>	
	<STX>sRN EIMacAdr<ETX>	
	sRN EIMacAdr	
	02 73 57 4E 20 45 49 4D 61 63 41 64 72 03	
CoLa B	02 02 02 02 00 00 00 0C 73 52 4E 20 45 49 4D 61 63 41 64 72 5B	
	73 52 4E 20 45 49 4D 61 63 41 64 72	

Table 289: Telegram structure: sRA EIMacAdr

Telegram structure: sRA EIMacAdr						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sRA	73 52 41
Command	Read MAC address of the device	String	8		EIMacAdr	45 49 4D 61 63 41 64 72

Telegram structure: sRA EIMacAdr						
Telegram part	Description	Variable	Length	Additional details	Values CoLa A (ASCII)	Values CoLa B (Binary)
MAC address	Values	Uint_8	1	First part of MAC address	0 ...+255d (00...FF)	00 ... FF
		Uint_8	1	Second part of MAC address	0 ...+255d (00...FF)	00 ... FF
		Uint_8	1	Third part of MAC address	0 ...+255d (00...FF)	00 ... FF
		Uint_8	1	Fourth part of MAC address	0 ...+255d (00...FF)	00 ... FF
		Uint_8	1	Fifth part of MAC address	0 ...+255d (00...FF)	00 ... FF
		Uint_8	1	Sixth part of MAC address	0 ...+255d (00...FF)	00 ... FF

Table 290: Example: sRA EIMacAdr 00:06:77:22:40:EA

CoLa A	<STX>sRA{SPC}EIMacAdr{SPC}0{SPC}6{SPC}77{SPC}22{SPC}40{SPC}EA<ETX>
	02 73 52 41 20 45 49 4D 61 63 41 64 72 20 30 20 36 20 37 37 20 32 32 20 34 30 20 45 41 03
CoLa B	02 02 02 02 00 00 00 13 73 52 41 20 45 49 4D 61 63 41 64 72 20 00 06 77 22 40 EA 8D

13.4.1.4.8.13 Enable/ disable CoLa Scan [sWN EnableColaScan]

NOTE

Port 30178 will not be opened.

Finding the device via device search and changing the IP adress via SICK Engineering Tools is not possible anymore.

This command enables/ disables the device search by SICK Enginnering Tools.

Table 291: Telegram structure: sWN EnableColaScan

Telegram structure: sWN EnableColaScan (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 CoLa Scan / AutoIP	String	14		EnableColaS-can	45 6E 61 62 6C 65 43 6F 6C 61 53 63 61 6E
CoLa Scan / AutoIP	Enable/ disable	Bool_1	1	Disable: Enable:	0 1	00 01

Table 292: Example: sWN EnableColaScan

CoLa A	<STX>sWN{SPC}EnableColaScan{SPC}1<ETX>
	<STX>sWN EnableColaScan 1<ETX>
	sWN EnableColaScan 1
	02 73 57 4E 20 45 6E 61 62 6C 65 43 6F 6C 61 53 63 61 6E 20 31 03

CoLa B	02 02 02 02 00 00 00 14 73 57 4E 20 45 6E 61 62 6C 65 43 6F 6C 61 53 63 61 6E 20 01 54
	73 57 4E 20 45 6E 61 62 6C 65 43 6F 6C 61 53 63 61 6E 20 01

Table 293: Telegram structure: sWA EnableColaScan

Telegram structure: sWA EnableColaScan						
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 CoLa Scan / AutoIP	String	14		EnableColaS-can	45 6E 61 62 6C 65 43 6F 6C 61 53 63 61 6E

Table 294: Example: sWA EnableColaScan

CoLa A	<STX>sWA{SPC}EnableColaScan<ETX>
	02 73 57 41 20 45 6E 61 62 6C 65 43 6F 6C 61 53 63 61 6E 03
CoLa B	02 02 02 02 00 00 00 13 73 57 41 20 45 6E 61 62 6C 65 43 6F 6C 61 53 63 61 6E 20 5A

13.4.1.4.8.14 Enable/ disable CoLa1 interface [sWN ElAuxEnable]

After enabling the CoLa1 interface, use port 2111 for CoLa A and port 2112 for CoLa B.

Table 295: Telegram structure: sWN ElAuxEnable

Telegram structure: sWN ElAuxEnable (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 CoLa1 interface	String	11		ElAuxEnable	45 49 41 75 78 45 6E 61 62 6C 65
CoLa1 interface	Enable/ disable	Bool_1	1	Disable: Enable:	0 1	00 01

Table 296: Example: sWN ElAuxEnable

CoLa A	<STX>sWN{SPC}ElAuxEnable{SPC}1<ETX>
	<STX>sWN ElAuxEnable 1<ETX>
	sWN ElAuxEnable 1
	02 73 57 4E 20 45 49 41 75 78 45 6E 61 62 6C 65 20 31 03
CoLa B	02 02 02 02 00 00 00 11 73 57 4E 20 45 49 41 75 78 45 6E 61 62 6C 65 20 01 0A
	73 57 4E 20 45 49 41 75 78 45 6E 61 62 6C 65 20 01

Table 297: Telegram structure: sWA EIAuxEnable

Telegram structure: sWA EIAuxEnable						
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 CoLa1 interface	String	11		EIAuxEnable	45 49 41 75 78 45 6E 61 62 6C 65

Table 298: Example: sWA EIAuxEnable

CoLa A	<STX>sWA{SPC}EIAuxEnable<ETX>
	02 73 57 41 20 45 49 41 75 78 45 6E 61 62 6C 65 03
CoLa B	02 02 02 02 00 00 00 10 73 57 41 20 45 49 41 75 78 45 6E 61 62 6C 65 20 04

13.4.1.4.8.15 Set Webserver state [sMN SetWebserverEnabled]

This command enables/ disables the Webserver. Port 80 will not be opened after a reboot.

Table 299: Telegram structure: sMN SetWebserverEnabled

Telegram structure: sMN SetWebserverEnabled (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 Webserver state	String	19		SetWebserver-Enabled	53 65 74 57 65 62 73 65 72 76 65 72 45 6E 61 62 6C 65 64
State	Enable/ disable	Bool_1	1	Disable: Enable:	0 1	00 01

Table 300: Example: sMN SetWebserverEnabled

CoLa A	<STX>sMN{SPC}SetWebserverEnabled{SPC}1<ETX>	
	<STX>sMN SetWebserverEnabled 1<ETX>	
	sMN SetWebserverEnabled 1	
	02 73 4D 4E 20 47 65 74 57 65 62 73 65 72 76 65 72 45 6E 61 62 6C 65 64 20 31 03	
CoLa B	02 02 02 02 00 00 00 19 73 4D 4E 20 53 65 74 57 65 62 73 65 72 76 65 72 45 6E 61 62 6C 65 64 20 01 23	
	73 4D 4E 20 53 65 74 57 65 62 73 65 72 76 65 72 45 6E 61 62 6C 65 64 20 01	

Table 301: Telegram structure: sAN SetWebserverEnabled

Telegram structure: sAN SetWebserverEnabled						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sAN	73 41 4E

Telegram structure: sAN SetWebserverEnabled						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command	Set Webserver state	String	19		SetWebserver-Enabled	53 65 74 57 65 62 73 65 72 76 65 72 45 6E 61 62 6C 65 64

Table 302: Example: sAN SetWebserverEnabled

CoLa A	<STX>sAN{SPC}SetWebserverEnabled<ETX>
	02 73 41 4E 20 47 65 74 57 65 62 73 65 72 76 65 72 45 6E 61 62 6C 65 64 03
CoLa B	02 02 02 02 00 00 00 18 73 41 4E 20 47 65 74 57 65 62 73 65 72 76 65 72 45 6E 61 62 6C 65 64 2E

13.4.1.4.8.16 Read Webserver state [sMN GetWebserverEnabled]

Returns state if Webserver is enabled.

Table 303: Telegram structure: sMN GetWebserverEnabled

Telegram structure: sMN GetWebserverEnabled (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	Webserver state	String	19		GetWebserver-Enabled	47 65 74 57 65 62 73 65 72 76 65 72 45 6E 61 62 6C 65 64

Table 304: Example: sMN GetWebserverEnabled

CoLa A	<STX>sMN{SPC}GetWebserverEnabled<ETX>	
	<STX>sMN GetWebserverEnabled<ETX>	
	sMN GetWebserverEnabled	
	02 73 4D 4E 20 47 65 74 57 65 62 73 65 72 76 65 72 45 6E 61 62 6C 65 64 03	
CoLa B	02 02 02 02 00 00 00 17 73 4D 4E 20 47 65 74 57 65 62 73 65 72 76 65 72 45 6E 61 62 6C 65 64 16	
	73 4D 4E 20 47 65 74 57 65 62 73 65 72 76 65 72 45 6E 61 62 6C 65 64	

Table 305: Telegram structure: sAN GetWebserverEnabled

Telegram structure: sAN GetWebserverEnabled						
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	Webserver state	String	19		GetWebserver-Enabled	47 65 74 57 65 62 73 65 72 76 65 72 45 6E 61 62 6C 65 64

Telegram structure: sAN GetWebserverEnabled						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
State	Enabled/ disabled	Bool_1	1	Disabled: Enabled:	0 1	00 01

Table 306: Example: sAN GetWebserverEnabled

CoLa A	<STX>sAN{SPC}GetWebserverEnabled{SPC}1<ETX>
	02 73 41 4E 20 47 65 74 57 65 62 73 65 72 76 65 72 45 6E 61 62 6C 65 64 20 31 03
CoLa B	02 02 02 02 00 00 00 19 73 41 4E 20 47 65 74 57 65 62 73 65 72 76 65 72 45 6E 61 62 6C 65 64 20 01 3B

13.4.1.4.8.17 Enable/ disable LEDs [sWN LEDEnable]

This command enables/ disables the LEDs of the device.

Table 307: Telegram structure: sWN LEDEnable

Telegram structure: sWN LEDEnable						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3		sWN	73 57 4E
Command	Set LEDs	String	9		LEDEnable	4C 45 44 45 6E 61 62 6C 65
Status	Enable/ disable	Bool_1	1	Off: On:	0 1	00 01

Table 308: Example: sWN LEDEnable

CoLa A	<STX>sWN{SPC}LEDEnable{SPC}1<ETX>	
	<STX>sWN LEDEnable 1<ETX>	
	sWN LEDEnable 1	
	02 73 57 4E 20 4C 45 44 45 6E 61 62 6C 65 20 31 03	
CoLa B	02 02 02 02 00 00 00 0F 73 57 4E 20 4C 45 44 45 6E 61 62 6C 65 20 01 07	
	73 57 4E 20 4C 45 44 45 6E 61 62 6C 65 20 01	

Table 309: Telegram structure: sWA LEDEnable

Telegram structure: sWA LEDEnable						
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 LEDs	String	9		LEDEnable	4C 45 44 45 6E 61 62 6C 65

Table 310: Example: sWA LEDEnable

CoLa A	<STX>sWA{SPC}LEDEnable<ETX>
	02 73 57 41 20 4C 45 44 45 6E 61 62 6C 65 03
CoLa B	02 02 02 02 00 00 00 0E 73 57 41 20 73 57 41 20 4C 45 44 45 6E 61 62 6C 65 20 09

13.4.1.4.8.18 Read state of LEDs [sRN LEDState]

Read the current state of the LEDs.

Table 311: Telegram structure: sRN LEDState

Telegram structure: sRN LEDState						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3		sRN	73 52 4E
Command	Read state of LEDs	String	8		LEDState	4C 45 44 53 74 61 74 65

Table 312: Example: sRN LEDState

CoLa A	<STX>sRN{SPC}LEDState<ETX>	
	<STX>sRN LEDState<ETX>	
	sRN LEDState	
	02 73 52 4E 20 4C 45 44 53 74 61 74 65 03	
CoLa B	02 02 02 02 00 00 00 0C 73 52 4E 20 4C 45 44 53 74 61 74 65 55	
	73 52 4E 20 4C 45 44 53 74 61 74 65	

Table 313: Telegram structure: sRA LEDState

Telegram structure: sRA LEDState						
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	State of LEDs	String	8		LEDState	4C 45 44 53 74 61 74 65
LED Color	-	Enum_8	1	Green: Yellow: Red:	0 1 2	00 01 02
LED behavior	-	Enum_8	1	On: Off: Blinking: Blinking fast: Blinking delayed: Find me active (see "Initiate an acoustic or visual signal for a defined period of time [sMN FindMe]", page 142):	0 1 2 3 4 5	00 01 02 03 04 05
LED ID	Name of the LED	String	8		LED1 LED2	4C 45 44 31 4C 45 44 32

Table 314: Example: sRA LEDState

CoLa A	<STX>sRA{SPC}LED-State{SPC}2{SPC}0{SPC}0{SPC}LED2{SPC}0{SPC}0{SPC}0{SPC}0{SPC}1{SPC}0{SPC}LED1{SPC}0{SPC}0{SPC}0{SPC}0<ETX>
	02 73 52 41 20 4C 45 44 53 74 61 74 65 20 32 20 30 20 30 20 4C 45 44 32 00 00 00 00 20 31 20 30 20 4C 45 44 31 00 00 00 00 03

CoLa B	02 02 02 02 00 00 00 23 73 52 41 20 4C 45 44 53 74 61 74 65 20 00 02 00 00 4C 45 44 32 00 00 00 00 01 00 4C 45 44 31 00 00 00 00 7A

13.4.1.4.9 Application

13.4.1.4.9.1 Set activation of evaluation group [sMN ActivateEvaluationGroup]

The telegram is intended to activate or deactivate groups via telegram. The group activation needs to be changed from always to telegram in advance.

Table 315: Telegram structure: sMN ActivateEvaluationGroup

Telegram structure: sMN ActivateEvaluationGroup						
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 / deactivate evaluation group	String	23		ActivateEvaluationGroup	41 63 74 69 76 61 74 65 45 76 61 6C 75 61 74 69 6F 6E 47 72 6F 75 70
Amount of evaluation groups to activate/ deactivate	Array	Uint_16	2		1d ... 48d (1 ... 30h)	00 01 ... 00 30
Evalution group number	Evaluation group 1	UInt_16	2		1	00 01
Activate/ deactivate evaluation group	Activate / deactivate evaluation group 1	Bool_1	1	Deactivate: Activate:	0 1	00 01
Evalution group number	Evaluation group 2	UInt_16	2		2	00 02
Activate/ deactivate evaluation group	Activate / deactivate evaluation group 2	Bool_1	1	Deactivate: Activate:	0 1	00 01
...						
Evalution group number	Evaluation group 48	UInt_16	2		48d (30h)	00 30
Activate/ deactivate evaluation group	Activate / deactivate evaluation group 48	Bool_1	1	Deactivate: Activate:	0 1	00 01

Table 316: Example1: sMN ActivateEvaluationGroup – Deactivate evaluation group 1

CoLa A	<STX>sMN{SPC}ActivateEvaluationGroup{SPC}1{SPC}1{SPC}0<ETX>
	<STX>sMN ActivateEvaluationGroup 1 1 0<ETX>
	sMN ActivateEvaluationGroup 1 1 0
	02 73 4D 4E 20 41 63 74 69 76 61 74 65 45 76 61 6C 75 61 74 69 6F 6E 47 72 6F 75 70 20 31 20 31 20 30 03
CoLa B	02 02 02 02 00 00 00 21 73 4D 4E 20 41 63 74 69 76 61 74 65 45 76 61 6C 75 61 74 69 6F 6E 47 72 6F 75 70 20 00 01 00 01 00 20
	73 4D 4E 20 41 63 74 69 76 61 74 65 45 76 61 6C 75 61 74 69 6F 6E 47 72 6F 75 70 20 00 01 00 01 00

Table 317: Example2: sMN ActivateEvaluationGroup – Deactivate evaluation group 1, activate evaluation group 2 and 3

CoLa A	<STX>sMN{SPC}ActivateEvaluationGroup{SPC}3{SPC}1{SPC}0{SPC}2{SPC}1{SPC}3{SPC}1<ETX>
	<STX>sMN ActivateEvaluationGroup 3 1 0 2 1 3 1<ETX>
	sMN ActivateEvaluationGroup 3 1 0 2 1 3 1
	02 73 4D 4E 20 41 63 74 69 76 61 74 65 45 76 61 6C 75 61 74 69 6F 6E 47 72 6F 75 70 20 33 20 31 20 30 20 32 20 31 20 33 20 31 03
CoLa B	02 02 02 02 00 00 00 21 73 4D 4E 20 41 63 74 69 76 61 74 65 45 76 61 6C 75 61 74 69 6F 6E 47 72 6F 75 70 20 00 03 00 01 00 00 02 01 00 03 01 20
	73 4D 4E 20 41 63 74 69 76 61 74 65 45 76 61 6C 75 61 74 69 6F 6E 47 72 6F 75 70 20 00 03 00 01 00 00 02 01 00 03 01

Table 318: Telegram structure: sAN ActivateEvaluation

Telegram structure: sAN ActivateEvaluation						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sAN	73 52 41
Command	Activate / deactivate evaluation group	String	23		ActivateEvaluationGroup	41 63 74 69 76 61 74 65 45 76 61 6C 75 61 74 69 6F 6E 47 72 6F 75 70
Amount of activated/ deactivated evaluation groups	-	UInt_16	2		1h... 30h	00 01 ... 00 30
State of activation / deactivation	-	Bool_1	1	Activation/Deactivation failed: Successfully activated/deactivated:	0 1	00 01

Table 319: Example 1: sAN ActivateEvaluation – Successfully deactivated evaluation group 1

CoLa A	<STX>sAN{SPC}ActivateEvaluation{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 47 72 6F 75 70 20 31 20 31 03
CoLa B	02 02 02 02 00 00 00 1F 73 41 4E 20 41 63 74 69 76 61 74 65 45 76 61 6C 75 61 74 69 6F 6E 47 72 6F 75 70 20 00 01 01 2C

Table 320: Example 2: sAN ActivateEvaluation – Successfully deactivated evaluation group 1, successfully activated evaluation group 2, failed activation of evaluation group 3

CoLa A	<STX>sAN{SPC}ActivateEvaluation{SPC}3{SPC}1{SPC}1{SPC}0<ETX>
	02 73 41 4E 20 41 63 74 69 76 61 74 65 45 76 61 6C 75 61 74 69 6F 6E 47 72 6F 75 70 20 31 20 31 20 31 20 30 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 47 72 6F 75 70 20 00 03 01 01 00 73

13.4.1.4.9.2 Set field evaluation contour [sMN SetFieldEvaluationContour]

This telegram can be used to change or reshape the contour/ polygon of an already defined field. It´s possible to add additional points.

Table 321: Telegram structure: sMN SetFieldEvaluationContour

Telegram structure: sMN SetFieldEvaluationContour						
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 the region of interest of an object detection evaluation	String	25		SetFieldEvaluationContour	53 65 74 46 69 65 6C 64 45 76 61 6C 75 61 74 69 6F 6E 43 6F 6E 74 6F 75 72
Evalution ID	-	UInt_16	2		+1d ... +48d (1 ... 30h)	00 01 ... 00 30
Amount of polygon verticies	E. g. triangle = 3, square = 4	Array	... 800		+3d ... +800d (3 ... 320h)	00 03... 03 20
Polygon vertex	Coordinates of first polygon vertex in mm	Int_32	4	X coordinate:	-60,000d ... +60,000d (FFFF15A0 ... EA60h)	FF FF 15 A0 ... 00 00 EA 60
				Y coordinate:	-60,000d ... +60,000d (FFFF15A0 ... EA60h)	FF FF 15 A0 ... 00 00 EA 60
Polygon vertex	Coordinates of second polygon vertex in mm	Int_32	4	X coordinate:	-60,000d ... +60,000d (FFFF15A0 ... EA60h)	FF FF 15 A0 ... 00 00 EA 60
				Y coordinate:	-60,000d ... +60,000d (FFFF15A0 ... EA60h)	FF FF 15 A0 ... 00 00 EA 60
...						
Polygon vertex	Coordinates of last polygon vertex in mm	Int_32	4	X coordinate:	-60,000d ... +60,000d (FFFF15A0 ... EA60h)	FF FF 15 A0 ... 00 00 EA 60
				Y coordinate:	-60,000d ... +60,000d (FFFF15A0 ... EA60h)	FF FF 15 A0 ... 00 00 EA 60

Telegram structure: sMN SetFieldEvaluationContour						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Lower Z limit	Lower bound of extrusion in Z direction	Int_32	4			
Upper Z limit	Upper bound of extrusion in Z direction	Int_32	4			

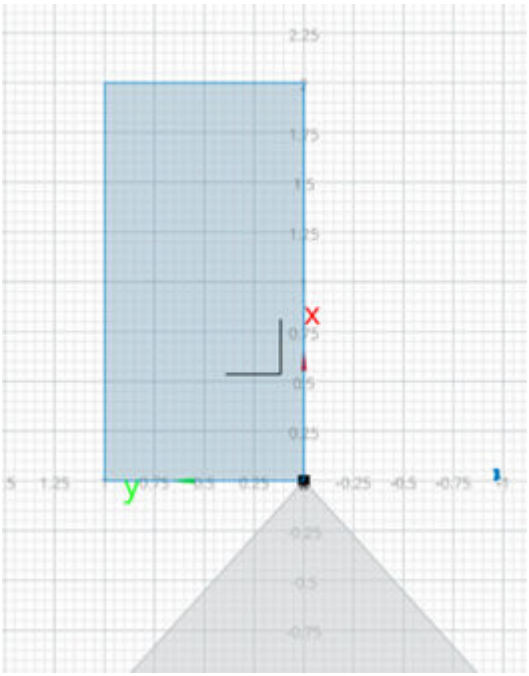


Table 322: Example: sMN SetFieldEvaluationContour

CoLa A	<STX>sMN{SPC}SetFieldEvaluationContour{SPC}1{SPC}4{SPC}+2000{SPC}+1000{SPC}+0{SPC}+1000{SPC}+0{SPC}+0{SPC}+2000{SPC}+0{SPC}+0{SPC}+1<ETX>
	<STX>sMN SetFieldEvaluationContour 1 4 +2000 +1000 +0 +1000 +0 +0 +2000 +0 +0 +1<ETX>
	sMN SetFieldEvaluationContour 1 4 +2000 +1000 +0 +1000 +0 +0 +2000 +0 +0 +1
	02 73 4D 4E 20 53 65 74 46 69 65 6C 64 45 76 61 6C 75 61 74 69 6F 6E 43 6F 6E 74 6F 75 72 20 31 20 34 20 2B 32 30 30 30 20 2B 31 30 30 30 20 2B 30 20 2B 31 30 30 30 20 2B 30 20 2B 30 20 2B 32 30 30 30 20 2B 30 20 2B 30 20 2B 31 03
CoLa B	02 02 02 02 00 00 00 4A 73 4D 4E 20 53 65 74 46 69 65 6C 64 45 76 61 6C 75 61 74 69 6F 6E 43 6F 6E 74 6F 75 72 20 00 01 00 04 00 00 07 D0 00 00 03 E8 00 00 00 00 00 00 03 E8 00 00 00 00 00 00 00 00 00 00 07 D0 00 00 00 00 00 00 00 00 00 00 00 01 1C
	73 4D 4E 20 53 65 74 46 69 65 6C 64 45 76 61 6C 75 61 74 69 6F 6E 43 6F 6E 74 6F 75 72 20 00 01 00 04 00 00 07 D0 00 00 03 E8 00 00 00 00 00 00 03 E8 00 00 00 00 00 00 00 00 00 00 07 D0 00 00 00 00 00 00 00 00 00 00 00 01

Table 323: Telegram structure: sAN SetFieldEvaluationContour

Telegram structure: sAN SetFieldEvaluationContour						
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 SetFieldEvaluationContour						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command	Set the region of interest of an object detection evaluation	String	25		SetFieldEvaluationContour	53 65 74 46 69 65 6C 64 45 76 61 6C 75 61 74 69 6F 6E 43 6F 6E 74 6F 75 72
State of evaluation conversion	-	Enum_8	1	Conversion successful: Invalid Evaluation: Invalid Polygon: Invalid Z limit:	0 1 2 3	00 01 02 03

Table 324: Example: sAN SetFieldEvaluationContour – Evaluation conversion successful

CoLa A	<STX>sAN{SPC}SetFieldEvaluationContour{SPC}0<ETX>
	02 73 41 4E 20 53 65 74 46 69 65 6C 64 45 76 61 6C 75 61 74 69 6F 6E 43 6F 6E 74 6F 75 72 20 30 03
CoLa B	02 02 02 02 00 00 00 1F 73 41 4E 20 53 65 74 46 69 65 6C 64 45 76 61 6C 75 61 74 69 6F 6E 43 6F 6E 74 6F 75 72 20 00 14

13.4.1.4.9.3 Read the current field evaluation application state [sRN FieldEvaluationApplicationState]

This telegram is intended to read the field evaluation application state. It returns the information whether the field evaluation application is active, configuring or deactivated.

Further information on the different states:

- Deactivated - no evaluation is set up
- Active - at least one active evaluation (group activation has no influence on that)
- Configuring - new calculation of the e.g. field geometries

Table 325: Telegram structure: sRN FieldEvaluationApplicationState

Telegram structure: sRN FieldEvaluationApplicationState						
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 on current field evaluation application state	String	31		FieldEvaluationApplicationState	46 69 65 6C 64 45 76 61 6C 75 61 74 69 6F 6E 41 70 70 6C 69 63 61 74 69 6F 6E 53 74 61 74 65

Table 326: Example: sRN FieldEvaluationApplicationState

CoLa A	<STX>sRN{SPC}FieldEvaluationApplicationState<ETX>	
	<STX>sRN FieldEvaluationApplicationState<ETX>	
	sRN FieldEvaluationApplicationState	
	02 73 52 4E 20 46 69 65 6C 64 45 76 61 6C 75 61 74 69 6F 6E 41 70 70 6C 69 63 61 74 69 6F 6E 53 74 61 74 65 03	

CoLa B	02 02 02 02 00 00 00 23 73 52 4E 20 46 69 65 6C 64 45 76 61 6C 75 61 74 69 6F 6E 41 70 70 6C 69 63 61 74 69 6F 6E 53 74 61 74 65 36
	73 52 4E 20 46 69 65 6C 64 45 76 61 6C 75 61 74 69 6F 6E 41 70 70 6C 69 63 61 74 69 6F 6E 53 74 61 74 65

Table 327: Telegram structure: sRA FieldEvaluationApplicationState

Telegram structure: sRA FieldEvaluationApplicationState						
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 on current field evaluation application state	String	31		FieldEvaluationApplicationState	46 69 65 6C 64 45 76 61 6C 75 61 74 69 6F 6E 41 70 70 6C 69 63 61 74 69 6F 6E 53 74 61 74 65
Field evaluation application state	-	Enum_8	1	Deactivated: Active: Configuring: Error:	0 1 2 3	00 01 02 03

Table 328: Example: sRA FieldEvaluationApplicationState

CoLa A	<STX>sRA{SPC}FieldEvaluationApplicationState{SPC}1<ETX>
	02 73 52 41 20 46 69 65 6C 64 45 76 61 6C 75 61 74 69 6F 6E 41 70 70 6C 69 63 61 74 69 6F 6E 53 74 61 74 65 20 31 03
CoLa B	02 02 02 02 00 00 00 25 73 52 41 20 46 69 65 6C 64 45 76 61 6C 75 61 74 69 6F 6E 41 70 70 6C 69 63 61 74 69 6F 6E 53 74 61 74 65 20 01 18

13.4.1.4.9.4 Read field evaluation result [sRN FieldEvaluationResult]

This telegram returns the status information of all evaluations.

Table 329: Telegram structure: sRN FieldEvaluationResult

Telegram structure: sRN FieldEvaluationResult						
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 on field evaluation result	String	21		FieldEvaluationResult	46 69 65 6C 64 45 76 61 6C 75 61 74 69 6F 6E 52 65 73 75 6C 74

Table 330: Example: sRN FieldEvaluationResult

CoLa A	<STX>sRN{SPC}FieldEvaluationResult<ETX>
	<STX>sRN FieldEvaluationResult<ETX>
	sRN FieldEvaluationResult
	02 73 52 4E 20 46 69 65 6C 64 45 76 61 6C 75 61 74 69 6F 6E 52 65 73 75 6C 74 03

CoLa B	02 02 02 02 00 00 00 19 73 52 4E 20 46 69 65 6C 64 45 76 61 6C 75 61 74 69 6F 6E 52 65 73 75 6C 74 12
	73 52 4E 20 46 69 65 6C 64 45 76 61 6C 75 61 74 69 6F 6E 52 65 73 75 6C 74

Table 331: Telegram structure: sRA FieldEvaluationResult

Telegram structure: sRA FieldEvaluationResult						
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 on field evaluation result	String	21		FieldEvaluation-Result	46 69 65 6C 64 45 76 61 6C 75 61 74 69 6F 6E 52 65 73 75 6C 74
Version	Version number	Uint_16	2		0 ... FFFFh	00 ... FF FF
Time-stamp	Current system time since power on in ms	Uint_32	4		0 ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
Evaluation Result List	Maximum possible evaluations = 48	Array	48	Not configured: Inactive: Free: Detecting infringed: Infringed: Detecting free:	0 1 2 3 4 5	00 01 02 03 04 05

Table 332: Example: sRA FieldEvaluationResult - field 1 is infringed, field 2 is free

CoLa A	<STX>sRA{SPC}FieldEvaluationRe-sult{SPC}1{SPC}1D869664{SPC}4{SPC}0{SPC}2{SPC}0 {SPC} ETX>
	02 73 52 41 20 46 69 65 6C 64 45 76 61 6C 75 61 74 69 6F 6E 52 65 73 75 6C 74 20 31 20 31 44 38 36 39 36 36 34 20 34 20 30 20 32 20 30 03
CoLa B	02 02 02 02 00 00 00 B4 73 52 41 20 46 69 65 6C 64 45 76 61 6C 75 61 74 69 6F 6E 52 65 73 75 6C 74 20 00 01 00 00 00 00 1D 86 96 64 04 00 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 53

13.4.1.5 Diagnostics

13.4.1.5.1 SOPAS error codes

sFA ErrorCode

Table 333: 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

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

Telegram structure: sFA ErrorCode			
Error code	**Description**	**Dec.**	**Hex.**
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	20

Table 334: 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

13.4.1.5.2 Additional information

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>

13.5 Integration of the product into mobile platforms

This section provides guidance for customers who want to integrate the product into a mobile application.

Different types of mobile platforms

The integration guide applies to all mobile platforms:

- Mobile platforms in logistics
- AMRs in mobile production
- Service/cleaning robots

LiDAR applications in mobile platforms

The product can perform a variety of tasks in an application. The product can be used for collision avoidance at the front and rear of the platform and for navigation. Protrusion monitoring, detection of pallet pockets or steps and volume measurement is possible when the product is mounted at an angle. The product also offers picking control.

13.5.1 Assignment of integration phases

The following phases are assigned to the listed sections to indicate in which phase the information might be important.

- **Testing**: In this phase, tests are performed to see how the device behaves in the intended target environment.
- **Design-in**: This phase covers the design-in process of the device. It provides information on what to consider when integrating the device into mobile platforms.

- **Production**: In this phase, instructions are given which are to be observed in the production process and in the series installation of the device.
- **Lifecycle**: This phase describes what needs to be considered during the lifecycle of the sensor.

Alignment aid for displaying the laser position [Testing | Design-in]

An alignment aid can help to precisely position the sensor and avoid alignment errors (e.g. part no.: 2101720).

Temperature fluctuations [Testing | Design-in]

When temperatures change rapidly, e.g. when moving from a cold storage facility to a hall at a normal temperature, the optics cover may fog up due to moisture. This can lead to a limited measuring capability. We recommend using the contamination indication see "Contamination indication", page 38.

Device Ready Status [Testing | Design-in]

The “device ready” status is required for operation of the sensor. If this status is not displayed, an error is present. The status can also be transmitted to all digital outputs, and is easily recognized via the LEDs, see "Visibility of the LEDs [Design-in]", page 173.

Error [Testing | Design-in]

So errors can be eliminated, it is necessary to display the errors and the type of error. The errors displayed to the customer are listed under **Diagnostics** in SOPASair see "Troubleshooting", page 59.

Filter types [Testing | Design-in]

By using suitable filters, it is possible to further increase the availability of the sensor in certain applications and also in outdoor area, see "Filter", page 20.

Contamination of the system plug [Testing | Design-in | Lifecycle | Production]

To avoid contamination, it is important that the system plug is always covered or connected. This also applies when a device is replaced. Furthermore, the system plug should be easily accessible so that the device can be replaced without any problems. Mounting and removal must be performed while the device is de-energized.

Permanent inductive charging [Design-in | Testing]

Inductive charging creates a strong magnetic field that can limit the functionality of the device. Particularly at risk are places where several magnetic fields overlap, for example branches in the induction strip. Additional shielding of the device is recommended.

Testing your own targets [Testing | Lifecycle]

The performance of the device also depends on the targets. It is therefore important to test the functionality on the customer's own targets see "Object sizes", page 25and see "Scanning range", page 32.

Minimum object size [Testing | Lifecycle]

It is important to know what target size can still be detected at what distance. For detailed information, see "Object sizes", page 25.

Note that the operating instructions present the worst case. We therefore recommend in-house testing to verify the performance under the customer's own conditions.

Scratch protection [Design-in]

Scratches in the optics cover can lead to incorrect values, as this can cause reflection of the beams. The optics cover has an anti-scratch coating for this reason, but the sensor should, if possible, still be mounted in such a way as to protect it from scratches.

Visibility of the LEDs [Design-in]

The LEDs indicate the operational status of the device. It is recommended to make them visible after the device is installed in order to quickly recognize whether the device is switched on or, for example, whether fields have been interrupted. For detailed information, see "Display and control elements", page 12.

The LEDs can be permanently switched off via SOPASair.

Light spot divergence [Design-in]

Due to the light spot divergence that occurs in LiDAR sensors, where the emitted beam spreads over a larger area as the distance between the sensor and the target is increased, a reduction in beam intensity with distance will occur.

The device has a low light spot divergence, but care should still be taken to ensure that the intensity of the beam is sufficient for the application in question, see "Features", page 61.

Vehicle housing [Design-in]

For the product to measure reliably, interference to the laser beam by the vehicle housing must be excluded. Any reflections that occur should also be taken into account. Using a protection hood ensures that the product is reliably protected from external influences such as ambient light and scratches.

Gap width in the housing [Design-in]

The reflected beam is significantly magnified compared to the emitted beam due to the light spot divergence of LiDAR sensors. To generate reliable measured values, this should be taken into account when dimensioning the gap width.

Restrictions in the field of view [Design-in]

In order to exclude irritations caused by reflections of the housing and a resulting object detection, it is recommended to maintain a separation between the scan range and the housing.

Scan field flatness (conical error and tilt) [Design-in]

The scan field flatness describes the production-related vertical deviation of the horizontal scan plane of the sensor. Conical errors and tilt can affect the three-dimensional measurements. This should be taken into consideration to ensure reliable measurement results. Conical errors can only be corrected for a small field of view.

Tilt errors can be compensated for in many cases by mounting the sensor at a compensating angle. The alignment bracket can be used for this (part number: 2136134).

Scanning behavior behind protective screens [Design-in]

Screens are often used to protect the sensor from, for example, dirt and moisture. It should be noted that optical elements in the transmit and receive paths can influence the measurement behavior. For recommendations on the use of protective screens, see "Measurement behavior through screens [Design-in]", page 174.

Measurement behavior through screens [Design-in]

A screen is an optical element and therefore affects the measurement. If the device needs to be protected by a screen, an anti-reflective and thin screen is recommended. It should also be scratch-resistant and have a low refractive index. The screen must be mounted in such a way that a direct reflection does not hit the receiver and no dust cannot settle on it. To this end, it is recommended to install the screen at a slightly inclined angle. We recommend using the multi-echo function and the fog filter.

Mounting on platforms with vibration [Design-in]

If the surface is uneven, the sensor is exposed to vibrations and shock, which can damage the sensor. The resilience of the device can be increased in these cases by using a shock absorber. The device was tested by means of shock and vibration tests, see "Shock and vibration [Design-in]", page 176.

Upside down mounting [Design-in]

It is possible to mount the sensor upside down. It should be noted that the laser beam always rotates in the same direction with respect to the sensor coordinate system (counterclockwise), regardless of the installed orientation.

Influence of magnetic fields on the motor of the sensor [Design-in]

Magnetic fields near the motor can cause an imbalance in the internal motor. If this exceeds a permissible limit, the sensor stops. If the sensor is to be used in the vicinity of magnets, a function check-out is recommended. Tests with permanent magnets show that they only have an effect if they are placed in the immediate vicinity of the motor.

Ventilation element [Design-in]

For detailed information, see "Ventilation element", page 41

Material components of the housing [Design-in]

For detailed information, see "Mechanics and electronics", page 62

External pressure on the optics cover [Design-in]

The motor of the sensor is located under the optics cover. Make sure that no external pressure is applied to the optics cover.

Temperature effects on the housing [Design-in]

Temperature fluctuations can result in increased tolerances. This effect should be taken into account when mounting the device, e.g., with regard to the spacing of the drill holes.

Reference target for positioning [Design-in | Testing]

Reference targets are often used, for example, for a precise approach to charging or transfer stations, and should be easily recognizable to the sensor. Material that ensures a good measuring capability of the sensor can be purchased using part number 5600079.

Access permission [Design-in]

To prevent the device from being reconfigured by unauthorized persons, it is recommend to change the default password, see "Password management", page 54.

Cybersecurity [Design-in]

Cybersecurity is used to protect against unwanted access by third parties.

A Cybersecurity Hardening Guide can be requested via psirt@sick.de.

Software driver [Design-in]

The device is supported by the following drivers:

- ROS1
- ROS2
- C++
- Python

These are available at the following link: https://github.com/SICKAG/sick_scan_xd; https://github.com/SICKAG/ScanSegmentAPI

Time stamp [Design-in | Testing]

Time stamps are helpful to evaluate when exactly each distance value was measured. This is of great importance in moving applications. The time stamps are sent to the customer with each data packet. The duration until the customer can access the time stamp depends on the transport time in the network.

Synchronization of devices [Design-in]

Since multiple devices are often used in one application, it makes sense to synchronize them. This is possible with the help of NTP or PTP.

Boot time/power-up time [Design-in]

The power-up time indicates how long it takes for the device to be ready for use from any state. The boot time, the time from the switched-off state to operation, is of importance in this regard. The device is designed for 24-hour use and therefore does not necessarily have to be switched off. For the power-up time, see "Performance", page 63.

Voltage supply [Design-in]

With battery-powered AMRs, the supply voltage at the device is not always constant. Limit values must not be exceeded. For the maximum voltage range, see "Mechanics and electronics", page 62.

Energy consumption [Design-in]

The power rating of the device must be considered when sizing the battery, see "Mechanics and electronics", page 62.

Orientation of the system plug [Design-in]

To provide a high level of flexibility during installation, it is possible to mount the system plug in different directions: horizontally to the rear or vertically downwards, see "Dimensional drawing", page 69

Memory of the system plug [Design-in]

To ensure the network configuration is retained after a system plug is replaced, the configuration is stored in the system plug see "Data backup", page 54.

Heat generation [Design-in]

To avoid overheating of the device, attention should be paid to the heat generation. To dissipate the heat in the best possible way, the device should be thermally connected to the largest possible surface so that it serves as a heat sink. It is best not to install the device in a heat-insulated location.

EMC – Short cable routing [Design-in]

To minimize the EMC effect, avoid coiling cables. Coiling cables can create a larger electromagnetic field. A simple cable routing is therefore recommended.

EMC – Shielded cables [Design-in]

Shielded cables prevent influences or interference from electromagnetic fields.

EMC – Ethernet CAT class [Design-in]

To ensure data transmission, we recommend CAT class 5 for the Ethernet cable.

EMC testing [Design-in]

For the EMC tests performed, see "Ambient data", page 64.

Horizontal and vertical edge hits [Design-in | Testing]

Edge hits can lead to incorrect distance values. An edge hit occurs when a single beam hits two different objects. This can falsify the distance calculation. The device uses HDDM+ and the multi-echo method, which provides up to three distance values per measuring beam. This applies to both vertical and horizontal edge hits.

Shock and vibration [Design-in]

It is recommended to compare the specified shock and vibration data with the vibration loads in the specific application and to provide a damping system if necessary.

Mutual interference with other light sources [Design-in]

Mutual interference between two or more devices is very unlikely due to the HDDM+ method used.

With the HDDM+ measuring method, several pulses are sent per measurement point. A single measurement point is then calculated from these subpulses. If a subpulse fails, the statistical measurement procedure still has sufficient information to calculate a measurement point. In addition, the emission of the subpulses is randomized in time. The probability of many faulty subpulses is therefore low.

External optical influence [Design-in | Testing]

Common external influences are the sun, arcs from welding, and light from photo-electric sensors. Please determine what sources of ambient light the sensor may be exposed to in the application. A protection hood that shades out ambient light could be helpful.

Ambient light due to AMR illumination [Design-in]

Lights such as LEDs, light strips or the like are often installed on AMRs. Light in the visible range does not affect the device due to a bandpass filter. The wavelength of other light pulse emitting devices must be taken into account (± 30 nm at 905 nm).

Brackets [Design-in]

There are various brackets that can be used for different applications see "Accessories", page 70. The device can be mounted using the three threads on the bottom or using the two threads on the back.

External reference target [Design-in]

External reference targets are needed if the exact position of the mobile platform is to be determined. It is an easy way to mark specific areas, such as busy hallways or areas with people. Reflectors are well suited as reference targets when RSSI information is used for evaluation. The reflectors must be mounted in the scan field. Reference geometries can also be used.

Screwdriver [Design-in]

To be able to easily tighten the M12 connections and fixing screws, the use of an appropriate tool is recommended. The mounting holes have an M4 thread. A separate torque screwdriver is available for the connections (part no. 2081618).

Update [Design-in]

Updates enable the device to be kept up to date. The manual update is done via SOPAS ET. To update semi-automatically (via batch file), use the SICK AppManager.

Operating temperature range [Design-in]

The device can only measure reliably within a defined temperature range, see "Ambient data", page 64.

It should be noted that cables may have different temperature specifications.

Runoff of liquid [Design-in | Lifecycle]

The device has an IP65/67 protection class rating. Nevertheless, water residues on the device should be avoided. In applications where wetness occurs, a water drain may be useful to keep stagnant liquid away from the device.

Ground hits due to ground unevenness [Design-in | Lifecycle | Production]

An object detection can occur when the device briefly measures the ground due to ground unevenness. If these vehicle pitching movements are unavoidable, a temporal filter can be used (field evaluation).

Water protection testing [Design-in | Lifecycle]

For the use and cleaning of the AMR, it is important to know which IP protection class the device has, see "Mechanics and electronics", page 62.

Detecting optics cover contamination [Design-in | Lifecycle]

If the optics cover is heavily contaminated, measurement inaccuracies may occur due to direct reflection of the emitted light pulse from the contamination see "Contamination indication", page 38.

Tightening torque of screws and connections [Production]

For precise mounting, it is important that the fixing screws and connections are tightened correctly. For the tightening torque of the screws (device and system plug mounting), see "Dimensional drawing", page 69.

The use of special torque screwdrivers is recommend for the cable connections (M12), see "Screwdriver [Design-in]", page 177.

Connecting cables [Production]

The pin assignment of the connecting cables can be found in the operating instructions and is printed on the system plug, see "Connections and pin assignment", page 49.

SICK's own connecting cables can be found under the following link: **https://www.sick.com/c/g274507**

The bend radii can be found in the respective specifications.

8028323/1NJW/2024-04-26/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.