OPERATING INSTRUCTIONS

TiM781

2D LiDAR sensor

SICK
Sensor Intelligence.

Product described

TiM781 (part number 1096807)

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

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

Prerequisites for safe work are:

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

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

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

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

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

1.2 Explanation of symbols

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

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

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

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

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

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

1.3 Further information

More information can be found on the product page.

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

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

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

The following information is available depending on the product:

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

2 Safety information

2.1 Intended use

The TiM781 2D LiDAR sensor features a scan plane and is designed for the following applications:

- 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.

It is suitable for applications which demand precise, non-contact optical measuring contours and dimensioning. Typical fields of application are, for example, stationary field protection, area monitoring, access control, mobile applications (navigation and anti-collision of mobile platforms).

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

2.2 Improper use

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

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

WARNING
Danger due to improper use!

Any improper use can result in dangerous situations.

Therefore, observe the following information:

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

2.3 Cybersecurity

Overview

To protect against cybersecurity threats, 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 Modifications and conversions

NOTICE

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

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

2.6 Requirements for skilled persons and operating personnel

WARNING

Risk of injury due to insufficient training.

Improper handling of the device may result in considerable personal injury and material damage.

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

The following qualifications are required for various activities:

Table 1: Activities and technical requirements

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

2.7 Operational safety and specific hazards

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.

CAUTION
Optical radiation: Class 1 Laser Product

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

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

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

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

WARNING
Electrical voltage!

Electrical voltage can cause severe injury or death.

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

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

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

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

3 Product description

3.1 Scope of delivery

The delivery of the device includes the following components:

Table 2: Scope of delivery

No. of units	Component	Notes
1	Device in the version ordered 2x fastening clips, 2x M3 x 5 mm screws	Without connecting cables
1	Printed safety notes, multilingual	Quick guide and general safety notes

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

3.2 Setup and dimensions

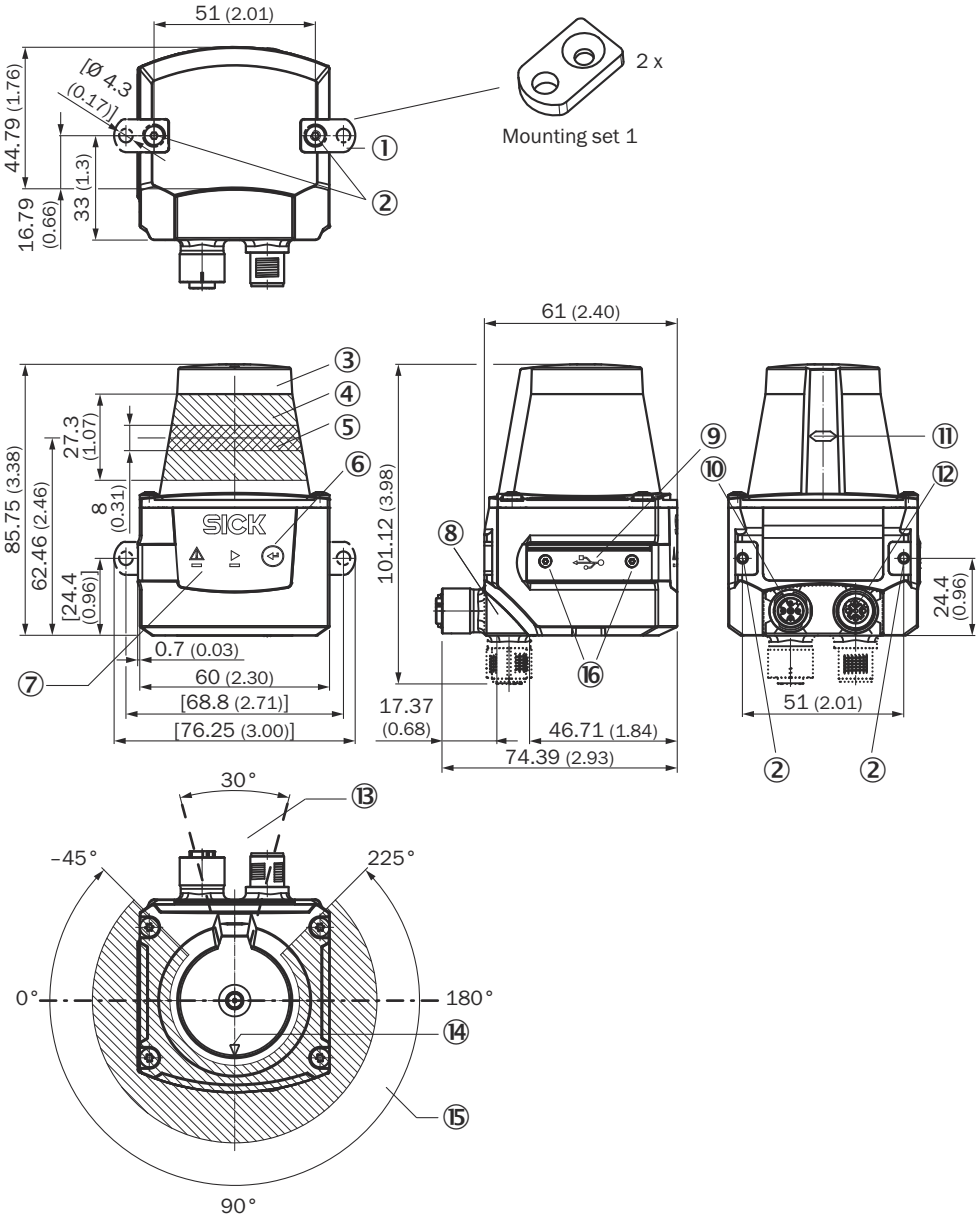


Figure 1: Structure and dimensions, unit: mm (inch), decimal separator: period

① 2x fastening clip with M3 x 5 mm countersunk screw, self-locking (included in scope of delivery)

② M3 threaded mounting hole, 2.8 mm deep (blind hole thread), max. tightening torque 0.8 Nm

③ Optics cover

④ Receiving range (light inlet)

⑤ Transmission range (light emission)

⑥ Function button for teach-in

⑦ Red and green LED (status indicators)

⑧ Swivel connector unit with electrical connections

⑨ Micro USB port, behind the black plastic cover (“Aux interface” connection for configuration with computer)

⑩ Connection for power supply, digital inputs/outputs, 12-pin M12 female connector.

⑪ Marking for the position of the light emission level
⑫ Ethernet connection, 4-pin M12 female connector
⑬ Area in which no reflective surfaces are permitted when the device is mounted
⑭ Bearing marking to support alignment (90° axis)
⑮ 270° aperture angle (visual range)
⑯ 2 x countersunk screw (Torx TX 6) M2 x 4 mm

3.3 Display and operating elements

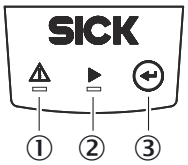


Figure 2: Display and operating elements

① Red LED (status indicator)
② Green LED (status indicator)
③ Teach-in pushbutton

Status indicators

LED (red)	LED (green)	Description
-	●	Device ready/monitoring mode
●	●	Object detection within field/fields (if there are lots of object detections in a short space of time, the red LED lighting up might look like flashing)
-	✳	Teach-in: start
●	●	Teach-in: End of pre-warning stage, 60 seconds teach-in stage
-	✳	Teach-in: End of teach-in phase
✳	-	Error, start up, firmware update, parameterization
-	-	Device without supply voltage

● = illuminated; ✳ = flashing

3.4 Type code

The devices of the product family are arranged according to the following type code:

TIM	x	y	z		–	aa	bb	c	dd
1	2	3	4	5		6	7	8	9

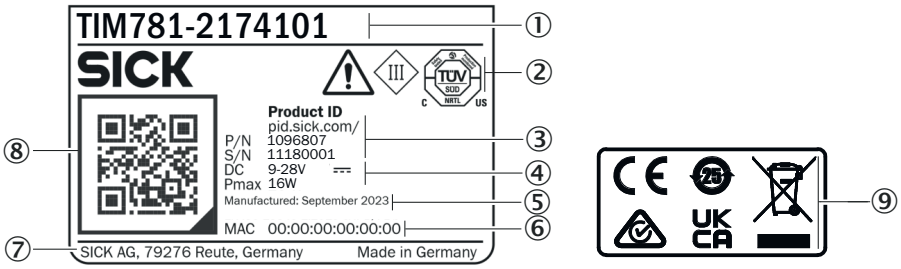
Table 3: Type code

Position	Description	Characteristic
1	Device name	TIM: Short range 2D-LiDAR sensor
2	Device type	7: Data supplier + field evaluation
3	Performed by	7: 25 m measuring range, 0.33° angular resolution, performance professional, HDDM 8: 25 m measuring range, 0.33° angular resolution, performance professional, HDDM+
4	Housing	1: Housing IP67 without heating

Position	Description	Characteristic
5		“Empty”: standard
6	Connection	21: Rotating connection unit, 1 x M12 male connector, 12-pin, D-coded (Power + I/O); 1x M12 female connector, 4-pin, A-coded (Ethernet)
7	Application	74: Measurement data supplier and flexible field evaluation with contour fields and particle filters
8	Laser type	1: Pulse power up to 880 mW, pulse width up to 5 ns, pulse rate 1,500 kHz
9	Color	01: Gray

3.5 Product identification

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



① Type code
② Conformity mark/certification mark, symbol: Observe the operating instructions!
③ Product ID with part number (P/N) and serial number (S/N)
④ Voltage supply, maximum power consumption
⑤ Production date
⑥ MAC address
⑦ Manufacturer/production location
⑧ QR code with product data (part number, serial number), example: pid.sick.com/1090608/23000000
⑨ Conformity mark/certification mark

3.6 Principle of operation

3.6.1 Measurement principle

The device is an opto-electronic LiDAR sensor (laser scanner) that uses laser beams for non-contact scanning of the outline of its surroundings on a plane. The device measures its surroundings in two-dimensional polar coordinates, relative to its measurement origin. This is marked by a circular indentation in the center of the optics cover. If a laser beam strikes an object, the position of that object is determined in terms of distance and direction.

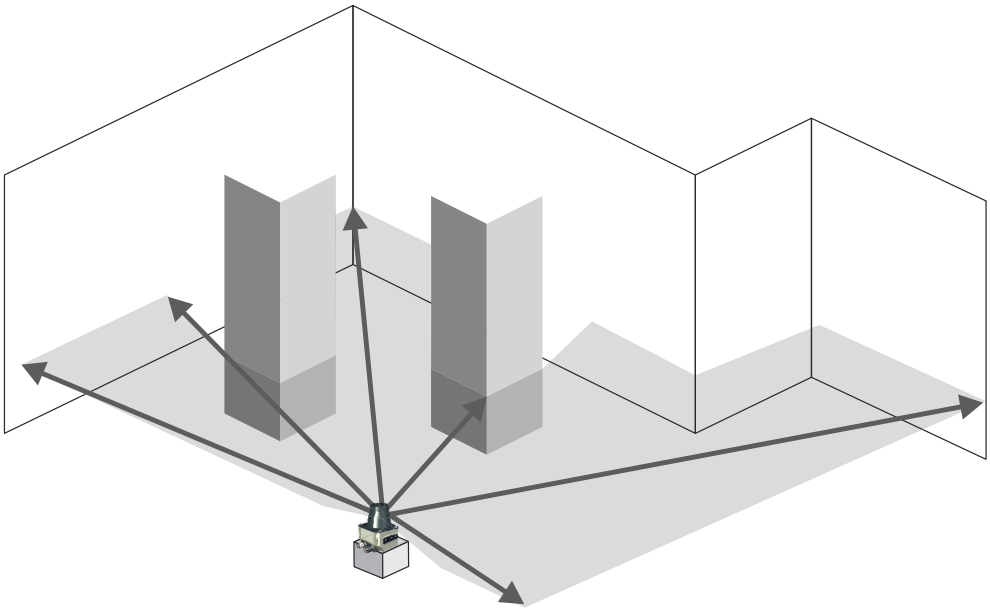


Figure 3: The 2D LiDAR sensor measurement principle

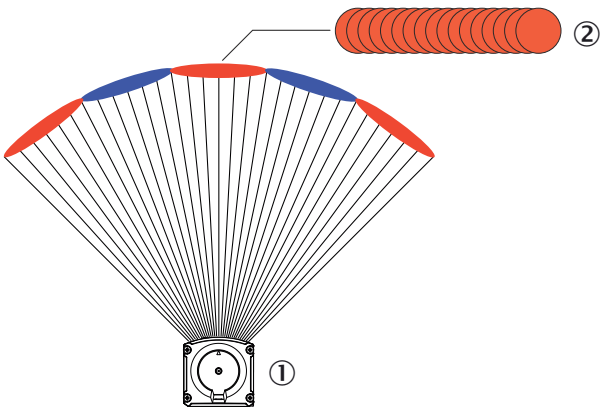
3.6.2 Distance measurement

The device emits beams pulsed by a laser diode. If a laser pulse hits an object or person, it is reflected on the surface of the object or person in question. The reflection is registered by a photosensitive element in the device receiver. The device uses SICK's own HDDM/HDDM+ (High Definition Distance Measurement) technology. With this measurement process, a measured value is formed by adding together multiple single pulses. The device calculates the distance from the object based on the elapsed time that the light requires between emitting the beam and receiving the reflection. Radar systems apply this "pulse time-of-flight measurement" principle in a similar way.

3.6.3 Direction measurement

The device uses a rotating mirror to deflect the emitted laser beams, thereby scanning its surroundings in a circular pattern. The measurements are triggered internally by an encoder in regular angle increments.

The measuring procedure uses the averaging from multiple pulses to determine individual measured values. A measuring point is the average value of several measurements combined.



① Device
② Laser pulse

3.6.4 Object sizes

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

Figure 4: Beam expansion

① Expanded laser beam

② Optical axis

Required values for calculating the light spot size and minimum object size:

- Light spot size on the device cover: 7 mm (rounded up)
- Light spot divergence per single pulse: 0.49 deg (8.6 mrad)
- supplement HDDM/HDDM+ (1 measured value consists of several overlapping single pulses): 5.8 mrad

Formula for calculating the light spot width:

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

Calculation example of light spot width at a distance of 4 m, with supplement 5.8 mrad:

(8.6 mrad +5.8 mrad) * 4,000 mm +7 mm = 64.6 mm

Formula for calculating the height of the light spot:

Light spot divergence [mrad] * Distance [mm] + Light spot size at the device cover [mm] = Light spot width [mm]

Example calculation of the light spot height at a distance of 4 m:

8.6 mrad * 4,000 mm +7 mm = 41.4 mm

Formula for calculating the minimum object size:

2 * supplement [mrad] * distance [mm] + light spot height [mm] = minimum object size [mm]

Calculation example of minimum object size at a distance of 4 m, with supplement 5.8 mrad:

2 * 5.8 mrad * 4,000 mm + 41.4 mm = 87.8 mm

NOTE

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

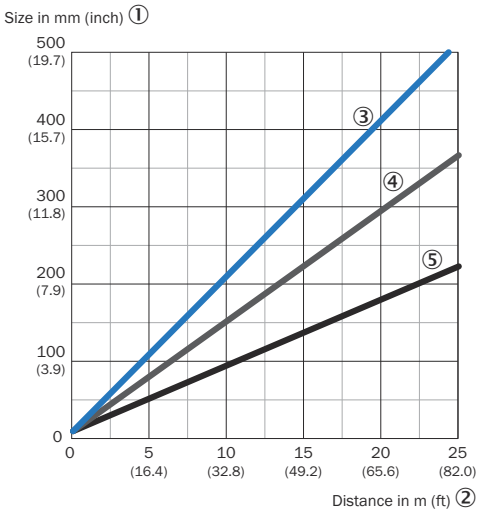


Figure 5: Minimum object size

① Size in millimeters (inc)
② Distance in meters (feet)
③ Minimum object size
④ Light spot width
⑤ Light spot height

3.6.5 Scan field flatness (conical error and tilt)

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 viewing range. Tilt errors can be compensated for in many cases by mounting the sensor at a compensating angle. An alignment aid (e.g. item no.: 2086761) can help to position the sensor precisely and avoid alignment errors.

3.6.6 Impact of object surfaces on the measurement

Remission value

Remission is the ability of a material to reflect light. The remission correlates with the amount of laser light emitted by the LiDAR sensor which is reflected by an object (see Lambert's law).

Glossy surfaces have different remissions at the same distance with different angles of impact. In the case of shiny surfaces, maximum remission is achieved when the beam makes vertical impact.

Matt and dull surfaces have diffuse remission. They therefore exhibit similar relative remissions with the same angle of impact regardless of the distance from the zero point.

Table 4: Typical remissions of frequently used materials

Material	Typ. relative remission
Rubber tires (vulcanized, black)	2%
Foam rubber (black)	2.4%
Photographic board (black, matte)	10%
Cardboard (gray)	20%
Wood (untreated fir, soiled)	40%

Material	Typ. relative remission
PVC (gray)	50%
Paper (white, matte)	80%
Plaster (white)	100%
Aluminum (black anodized)	110 ... 150%
Steel (stainless, shiny)	120 ... 150%
Steel (high gloss)	140 ... 200%

Reflection

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

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

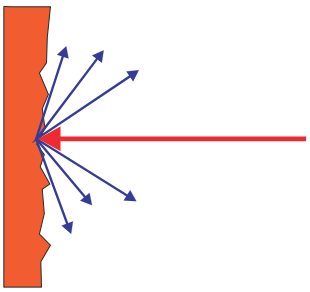


Figure 6: Reflection of light on the surface of the object

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.

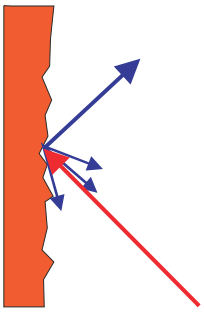


Figure 7: Angle of reflection

Retroreflection

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

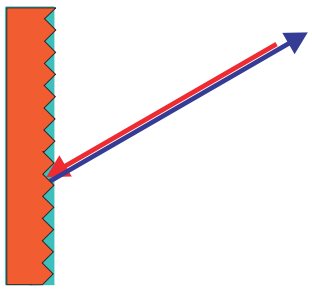


Figure 8: Retroreflection

Reflective surfaces

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

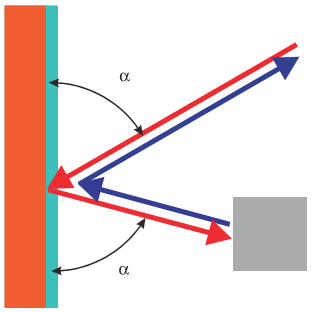


Figure 9: Specular surfaces

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.

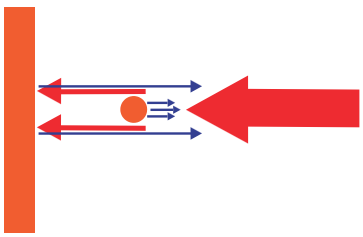


Figure 10: Object smaller than the laser beam diameter

3.6.7 Scanning range

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

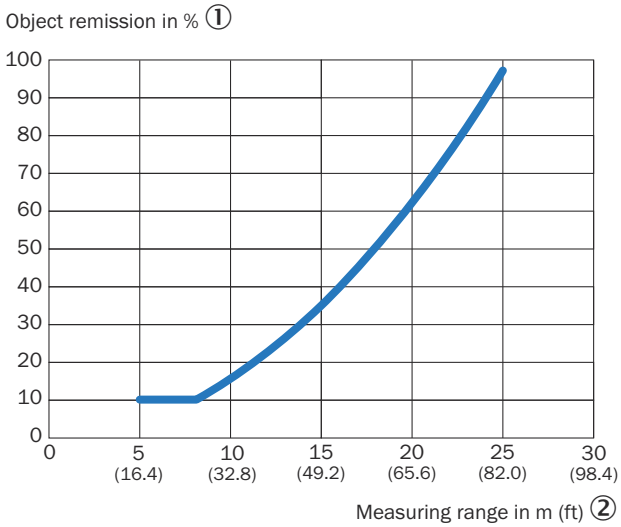


Figure 11: Scanning range rate as a function of object remission

① Object remission in percent

② Measuring range in meters (feet)

3.6.8 RSSI values

RSSI (Received Signal Strength Indicator) is an indication of the received signal strength for each measuring point. The RSSI values have a resolution of 8 bits with integer values between 0 and 254, where 0 represents no received signal and 254 the strongest signal.

The value changes non-linearly and is not calibrated. Low signal strengths are represented by a larger value range than high signal strengths.

Besides the remission of the measuring object, the angle of incidence of the measurement beam, ambient light, and temperature, this unitless value also depends on the distance.

The RSSI values can vary slightly among different devices and during the service life of the sensor.

3.6.9 Filter

The device has digital filters for pre-processing and optimizing the measured distance values. They enable the device to be adapted to meet the specific requirements of the respective application.

The filters can be combined without restrictions. If several filters are active, then the filters are applied sequentially to the results of the preceding filter. The processing sequence is as follows:

- Edge filter
- Particle filter
- Median filter
- Average filter

The active filter functions affect the output measured values. It is not possible to recalculate the original measured values from the filtered output values. For this reason, certain combinations of filters might not be advisable.

A particularly effective way to reduce the data in a scan (reduction of measurement points) is to restrict the scan range (“Data output” > “Output range”) or the media filter.

3.6.9.1 Edge filter

The edge filter eliminates erroneous or extreme distance values at edges. These arise as a result of laser light that partially hits an object in the front and partially hits an object farther away, or due to too little light remission at the object itself, see "Impact of object surfaces on the measurement", page 16. The filter evaluates the difference in distance between adjacent points.

The previous and next measurement point are taken into consideration in this. As soon as the configured maximum adjacent distance to the previous **and** next measurement point is exceeded, the device sets the distance value to 0 (value highlighted in see table 5, page 20 **bold**). If the maximum adjacent distance is not exceeded, or only exceeded to one of the adjacent measurement points, the measurement point is not filtered. If the measurement point or one of the adjacent measurement points is 0, then the filter is not applied to this measurement point.

Table 5: Example: Measured values with and without edge filter

	Measurement points (distance values in mm)							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**...**
not filtered	1800	1750	1145	1150	1147	800	871	...
filtered	1800	1750	1145	1150	1147	**0**	871	...

The edge filter enables points to be entirely suppressed at the outer edges of the object.

Maximum edge filter range (range limitation)

The maximum distance can be set to 8,000 mm [1]. The measurement points within the configured range are taken into account.

3.6.9.2 Particle filter

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

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

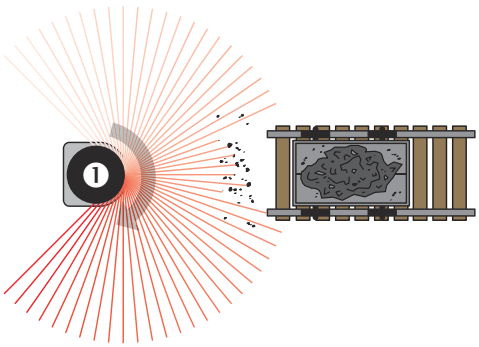


Figure 12: Without the particle filter: Violation of the contour due to dust particles in the vicinity of the object.

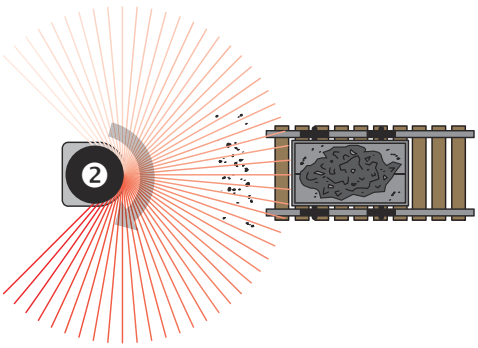


Figure 13: Using the particle filter: The response to dust particles in the detection field is delayed by one scan. Particles can thereby be blanked.

1) For the variants TiM310 / TiM320, the maximum distance can be set to 4,000 mm.

3.6.9.3 Median filter

The median filter filters out unwanted outliers in the measured data. When filtering, a distance value is compared with its two neighboring measuring points in a scan. The distance values are sorted by size and the distance value in the middle is output with the corresponding RSSI value.

There are two different types of median filter:

- The **moving median filter**, in which the viewing window of the original scan shifts by one after each filtered measuring point; this does not result in any data reduction.
- The **non-moving median filter** first looks at the first three measuring points of the original scan and filters this viewing window. The viewing window moves forward by the number of previously filtered measuring points and thus reduces by a third the number of measuring points that are output. The two outer measuring points at the start and end of a scan are not filtered.

Table 6: Example: Measured values without median filter

	Angle (distance values in mm)					
Scan	**1**	**2**	**3**	**4**	**5**	**...**
1	850	852	851	849	850	...
2	849	851	850	853	852	...
3	850	849	853	851	850	
4	851	852	854	851	849	
...	...	...	...	...	...	...

Table 7: Example: Measured values with moving median filter

	Angle (distance values in mm)					
Scan	**1**	**2**	**3**	**4**	**5**	**...**
1	850	851	851	850	...	...
2	849	850	851	852	...	...
3	850	850	851	851	...	...
4	851	852	852	851	...	...
...	...	...	...	...	...	...

Table 8: Example: Measured values with non-moving median filter

	Angle (distance values in mm)					
Scan	**1**	**2**	**3**	**4**	**5**	**...**
1	851			...		
2	851			...		
3	850			...		
4	852			...		
...	...			...		

3.6.9.4 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 can be configured (maximum 4 scans).

Table 9: Example: Average filter over 5 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	1225	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.

Once the measured value telegram has been confirmed, 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 digit of the first scan included in the averaging calculation is always output in the scan counter. 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.

Based on the scanning frequency of 15 Hz, a measured value is generated every 67 s. The time delay affecting data output results from this base value multiplied by the number of averaging operations (e.g., 2 averaging operations = 134 ms, 10 averaging operations = 670 ms).

3.6.10 Calculation of the field size for mobile applications

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

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

- the braking distance, which can be found in the vehicle documentation
- the distance covered during the vehicle control's response time, which can be found in the vehicle documentation
- The distance covered during the response time of the LiDAR sensor, "Technical data", page 51.

NOTE

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

Mounting height

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

3.6.10.1 Switching field length

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

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

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

SL = SA + ZG + ZR + ZB

SA = Stopping distance

ZG = General supplement of the LiDAR sensor = 100 mm

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

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

Stopping distance SA

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

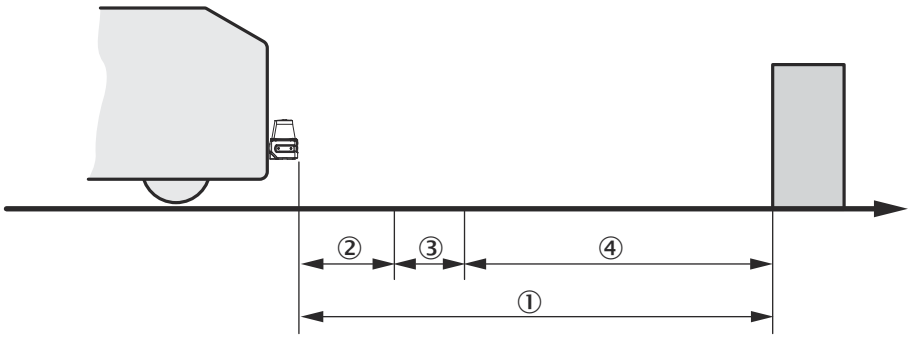


Figure 14: Stopping distance

① S_A

② S_{AnF}

③ S_{AnS}

④ S_{Br}

NOTE

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

How to calculate the stopping distance SA:

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

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

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

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

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

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

For more information on the response time T_S of the LiDAR sensor, see "Technical data", page 51.

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

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

T_S = Response time of the LiDAR sensor

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

The response time T_S of the LiDAR sensor depends on:

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

ZR supplement

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

3.6.10.2 Switching field width

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

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

SB = FB + 2 × (ZG + ZR)

FB = Vehicle width

ZG = General supplement of the LiDAR sensor = 100 mm

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

4 Transport and storage

4.1 Transport

NOTICE
Damage due to improper transport!

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

4.2 Unpacking

- To protect the device against condensation, allow it to equilibrate with the ambient temperature before unpacking if necessary.
- Handle the device with care and protect it from mechanical damage.
- To avoid ingress of dust and water, only remove the protective elements, e.g. protective caps of the electrical connections just before attaching the connecting cable.

4.3 Transport inspection

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

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

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

4.4 Storage

- Electrical connections are provided with a protective cap.
- Do not store outdoors.
- Store in a place protected from moisture and dust.
- Recommendation: Use the original packaging.
- To allow any residual dampness to evaporate, do not package in airtight containers.
- Do not expose to any aggressive substances.
- Protect from sunlight.
- Avoid mechanical shocks.
- Storage temperature: see "Technical data", page 51.
- Relative humidity: see "Technical data", page 51.
- For storage periods of longer than 3 months, check the general condition of all components and packaging on a regular basis.

5 Mounting

5.1 Mounting instructions

- Observe the technical data.
- Protect the sensor from direct and indirect sunlight.
- To prevent condensation, avoid exposing the device to rapid changes in temperature.
- The mounting site has to be designed for the weight of the device.
- The device can be mounted in any position.
- It should be mounted so that it is exposed to as little shock and vibration as possible. Optional mounting accessories are available, see "Accessories", page 55.
- In application areas 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 (see "Accessories", page 55). Mount the device in a freely suspended manner.
- When mounting the device, make sure there are no reflective surfaces behind the reference target, see "Setup and dimensions", page 11.
- When mounting the device, make sure the swivel connector area is recessed so it does not lie on the mounting surface see "Setup and dimensions", page 11.
- 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, see "Mutual interference", page 27.
- Avoid having shiny or reflective surfaces in the scanning range, e.g., stainless steel, aluminum, glass, reflectors, or surfaces with these types of coatings.
- Protect the device from moisture, contamination, and damage.
- Make sure that the status indicator is clearly visible.
- Do not subject the device to excessive shock or vibrations. In systems subjected to heavy vibrations, secure the fixing screws with screw-locking devices.
- The M3 x 5 screws included with delivery are intended for mounting the fastening clips via the blind hole threads on the rear or underside of the device, see "Setup and dimensions", page 11. If the mounting clamps are not used or if other screws are used, the screws must not be screwed into the thread by more than 2.8 mm. The maximum tightening torque is 0.8 Nm.

5.2 Mounting device

1. Mount the LiDAR sensor using the designated fixing holes, see "Setup and dimensions", page 11.

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.

2. Make the electrical connection. Attach and tighten a voltage-free cable, see "Connecting the device electrically", page 33 .
3. Switch on the supply voltage.

✓ The green operating LED lights up.

4. Align the vertical center line of the field of view of the device with the center of the area to be monitored. The marking (90° axis) on the upper side of the optics cover serves as a alignment aid.

5.3 Mutual interference

NOTE
Optical sensors and other IR light sources can influence the measurement and detection capabilities of the device.

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

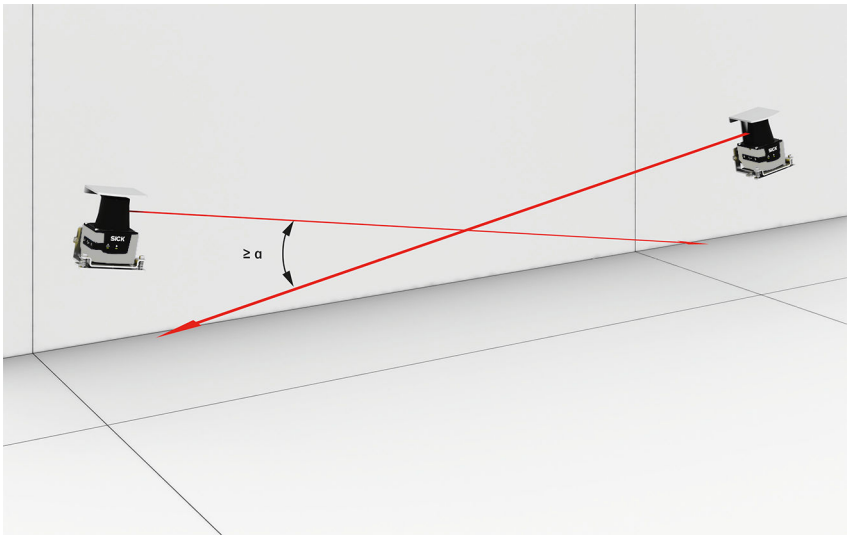


Figure 15: Angle ≥ 6 °

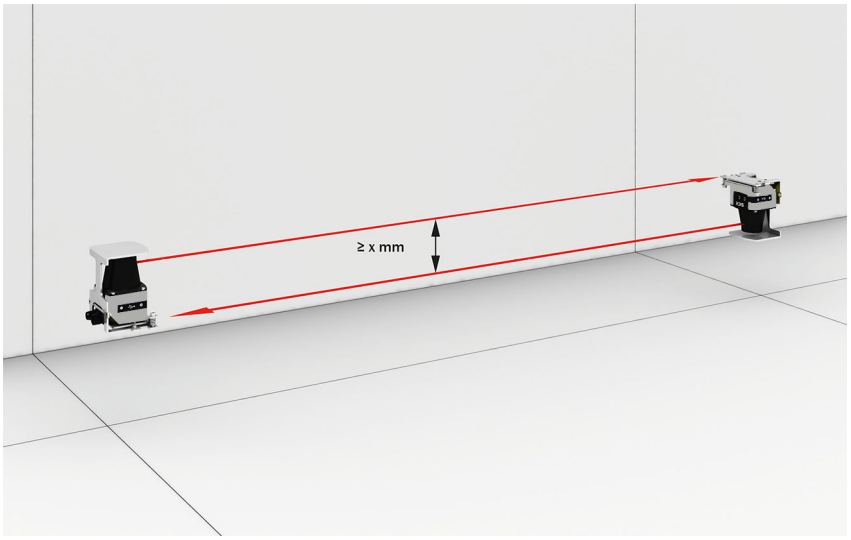


Figure 16: Distance ≥ 200 mm

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

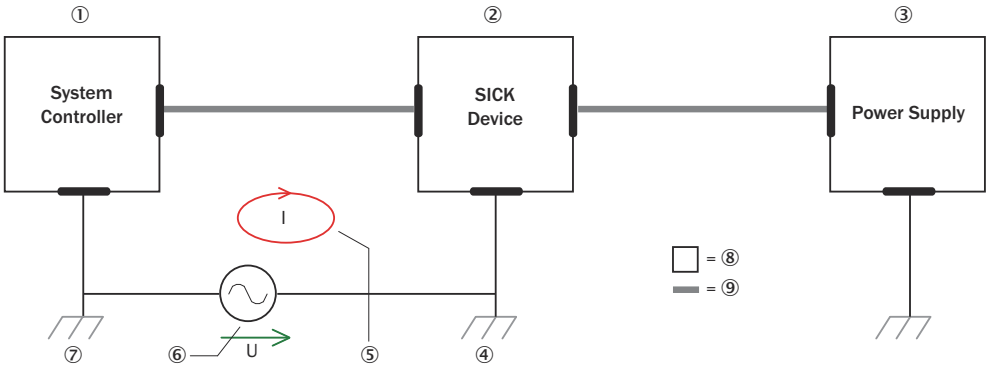


Figure 17: Example: Occurrence of equipotential bonding currents in the system configuration

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.

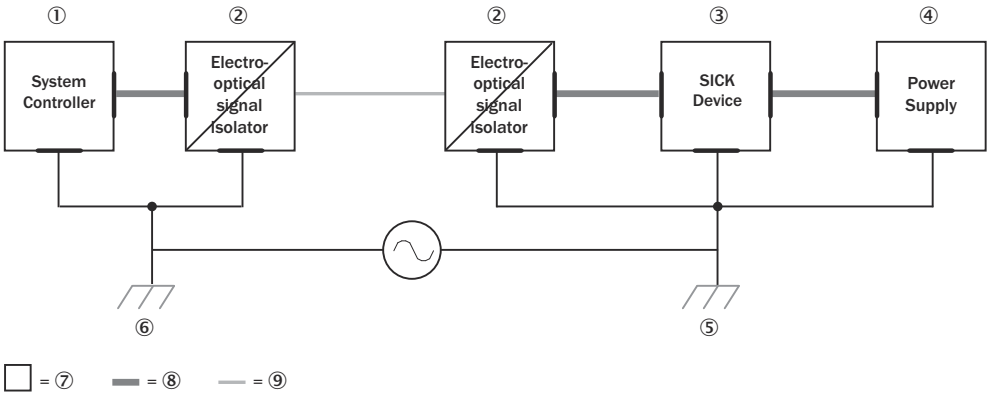


Figure 18: Example: Prevention of equipotential bonding currents in the system configuration by the use of electro-optical signal isolators

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.

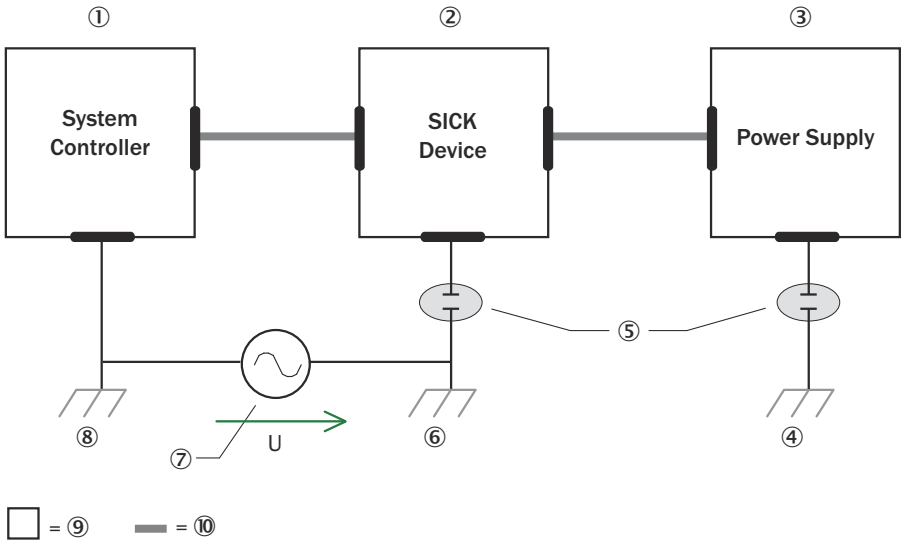


Figure 19: Example: Prevention of equipotential bonding currents in the system configuration by the insulated mounting of the device

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 Electrical block diagram for commissioning

TiMxxx-21xxxxx

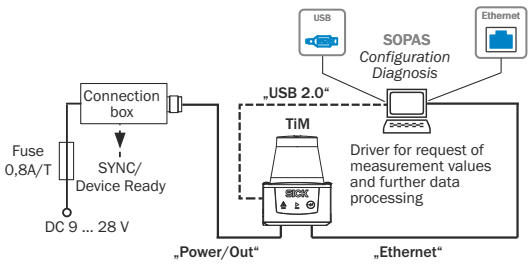


Figure 20: "Power, I/O" connection: With M12 male connector, 12-pin, A-coded; "Ethernet" connection: With M12 female connector, 4-pin, D-coded

6.3 Wiring instructions

NOTE

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

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

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

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

NOTICE

Faults during operation and defects in the device or the system

Incorrect wiring may result in operational faults and defects.

- Follow the wiring notes precisely.

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

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

Protect the device from dust and moisture when the plastic USB cover is open.

The USB interface is only for parameterization. Remove the USB cable for problem-free operation of the device.

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

Protect the device with an external 0.8 A slow-blow fuse at the beginning of the supply cable.

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

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

6.4 Connection diagram

NOTE

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

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

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

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

6.4.1 TiMxxx-21xxxxx

"Power/I/O" connection

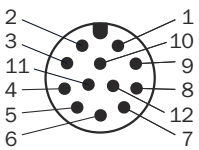


Table 10: Male connector, M12, 12-pin, A-coded

Pin	Signal	Function	Wire colors of connecting cable with flying leads [1]
1	GND	Ground	Blue
2	V_S	Supply voltage	Brown
3	IN 1	Digital input 1 (field set selection)	Red
4	IN 2	Digital input 2 (field set selection)	Green
5	OUT 1	Digital output 1 (object detection)	Pink
6	OUT 2	Digital output 2 (object detection)	Yellow
7	OUT 3	Digital output 3 (object detection)	Black
8	OUT 4	Digital output 4 (index/error)	Gray
9	IN_{GND}	Common ground for all inputs	White
10	IN 3	Digital input 3 (field set selection)	Violet
11	IN 4	Digital input 4 (field set selection)	Gray and pink
12	–	Reserved, do not wire this PIN!	Rot and blue
–	–	Shielding	–

[1] Example values when using the connecting cables part numbers 6054974 (5 m), 6054973 (10 m), 6054972 (20 m). Signal assignment and wire colors can vary when using other connecting cables.

"Ethernet" connection

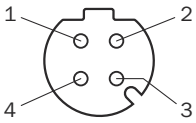


Table 11: Female connector, 4-pin, D-coded

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

6.4.2 USB interface

The Ethernet interface is recommended as a communication interface.

If using the USB interface, please note:

- Use a high-speed USB cable, maximum length of cable: 3 m.
- The connection may be interrupted due to ESD/EMC influences. If necessary: Disconnect USB cable from the device, then reconnect it. Restart communication via SOPAS ET software (“Online” button).

6.5 Connecting the device electrically

NOTICE

Observe the wiring instructions, see "Wiring instructions", page 32.

1. Ensure the voltage supply is not connected.
2. Connect the device according to the connection diagram, see "Connection diagram", page 32.

6.6 Wiring digital inputs / digital outputs

Digital inputs

The 4 switching digital inputs activate one of the 16 field sets in a binary combination as an evaluation case.

The digital inputs are decoupled from the supply voltage of the device. They have a common reference point (IN_{GND}), meaning they are not decoupled from one another.

The structure and wiring principle of the digital inputs are shown below.

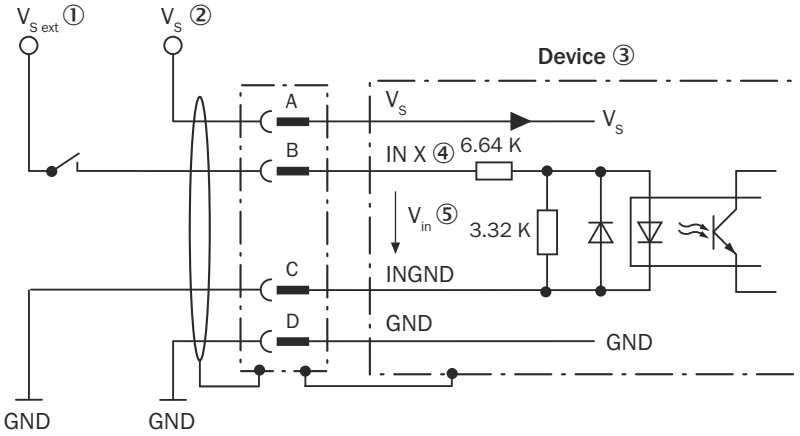


Figure 21: Wiring of digital input (PNP version)

① External supply voltage $V_{S\ ext}$

② Supply voltage V_s for device

③ Device

④ Signal IN X

⑤ Input voltage V_{in}

Position	Signal	TiM7xx-21xxxx (male connector M12, 12-pin): PIN
A	V_S	2
D	GND	1
C	IN_{GND}	9
B1	IN 1	3
B2	IN 2	4
B3	IN 3	10
B4	IN 4	11

Switching behavior	Current to the input starts the assigned function in the device. Default: Active high level, debounce 10 ms
Properties	Opto-decoupled Switchable with an electronic switch (PNP output) or mechanical switch
Electrical values	see "Mechanics/electronics", page 53

Digital outputs

The device has 4 switching digital outputs.

In combination, the digital outputs OUT 1 to OUT 3 signal the breach of the individual fields of a field set.

The digital output OUT 4 is used to issue an error and a regular index pulse.

The structure and wiring principle of the digital outputs are shown below.

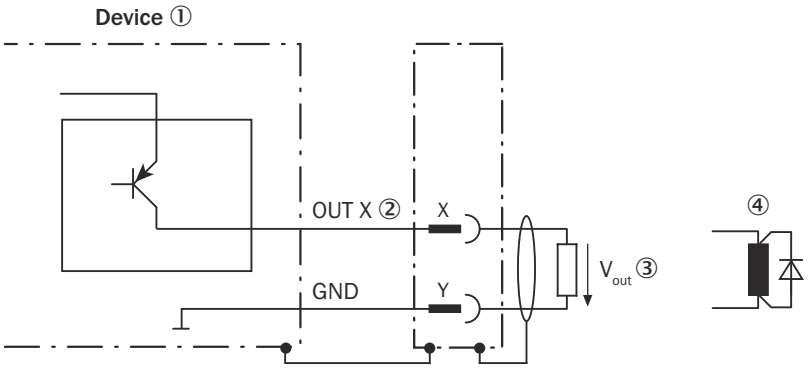


Figure 22: Wiring of digital output (PNP version)

① Device

② Signal OUT X

③ Output voltage V_{out}

④ If inductive load is present: Provide an arc-suppression switch at the digital output. Attach a freewheeling diode directly to the load for this purpose.

Position	Signal	TiM7xx-21xxxx (male connector M12, 12-pin): PIN
X1	OUT 1	5
X2	OUT 2	6
X3	OUT 3	7
X4	OUT 4	8
Y	GND	1

Switching behavior	PNP switching against supply voltage UV **OUT 1 ... OUT 3:** Resting level: High (no object detection), working level: Low (object detection) Response time: 134 ms ... 30 s (configurable via SOPAS ET), holding time: 0 ms ... 10 s (configurable via SOPAS ET) **OUT 4:** Resting level: High (device ready) working level: Low (error), low pulse (15 Hz, index, corresponds to measurement at 90°)
Properties	Short-circuit protected and temperature protected Not electrically isolated from supply voltage U_v
Electrical values	see "Mechanics/electronics", page 53

Longer connecting cables at the digital outputs of the device should be avoided due to the resulting fall in voltage. This is calculated as follows:

Δ U = (2 x length x current) : (conductance value x cross-section)

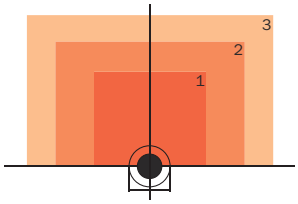
Conductance value for copper: 56 m/Ω mm^2.

Assignment of infringed fields – digital outputs

Output level: The level of the digital outputs OUT 1 ... OUT 3 is active low (in resting position: high, in working position: low (field infringed)).

All fields of a field set are considered infringed upon switching on, booting, in the event of an error and when the device is switched off.

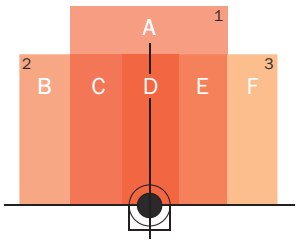
Example 1: 3 fields, rectangular, nested inside each other



Infringed fields	Digital outputs		
	OUT 1	OUT 2	OUT 3
Fields 1, 2, and 3 infringed	Active	Active	Active
Fields 2 and 3 infringed	Deactivated	Active	Active
Field 3 infringed	Deactivated	Deactivated	Active
No field infringed:	Deactivated	Deactivated	Deactivated

Example 2: 3 fields, rectangular, overlapping

The overlapping fields create areas that can be used as additional fields.



Infringed areas (fields)	Digital outputs		
	OUT 1	OUT 2	OUT 3
A (field 1 infringed)	Active	Deactivated	Deactivated
B (field 2 infringed)	Deactivated	Active	Deactivated
C (fields 1 and 2 infringed)	Active	Active	Deactivated
D (fields 1, 2, and 3 infringed)	Active	Active	Active
E (fields 1 and 3 infringed)	Active	Deactivated	Active
F (field 3 infringed)	Deactivated	Deactivated	Active
- (no field infringed)	Deactivated	Deactivated	Deactivated

7 Operation

7.1 SOPAS ET

The following activities are normally performed using the SOPAS ET configuration software:

- Measurement data output application: displaying the surrounding contour (measured line) detected by the device.
- Field evaluation application: adjusting the 3 fields of a field set.
- Parameterization: tailoring other device parameters to the application.
- Diagnosis: determining the cause of a fault.

To parameterize the device, you will require a computer with SOPAS ET installed and a free Ethernet connection (recommended) or, alternatively, a free USB interface.

NOTE

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

1. Connect the communication interface of the device to the computer.
2. Switch on and start the computer.
3. Supply the device with voltage.

✓ Following successful initialization, the green status LED lights up. The device is ready for use.

NOTE

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

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

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

Certain functions (e.g., Edit parameters) require you to be logged in to the device:

> **Device** > **Login** > Select user level and enter password:

NOTE

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

User levels	Password
Machine operator	–
Maintenance staff	`main`
Authorized client	`client`
Service	`servicelevel`

Table 12: User level and authorization

Machine operator	A **Machine operator** level user can view the basic device parameters. • No password required • Read only permissions • Not all parameters are visible
Maintenance	**Maintenance** can view the application-related device parameters. • Read only permissions • Not all parameters are visible • Can change the password for this user level
Authorized Client (Integrator)	Device parameters can be set as an **Authorized Client**. • Access to most parameters • Can change the password for this user level and the password for the **Maintenance** user level. • Can create a diagnostic report
Service	A **Service** level user can configure all device parameters. • Access to all parameters • Can change the password for this user level as well as the password for the user levels **Maintenance** and **Authorized Client** • Can create a diagnostic report • Can perform firmware updates

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

NOTE
If the password for the Service user level has been lost: see "Resetting the password for the Service user level", page 49.

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

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

NOTE
To reset the device to the factory settings, use the **Load defaults** option in SOPAS ET.
With the **Load application defaults** option, the network settings remain unchanged, all other settings are reset to the factory settings.

Tools

Depending on the selected view, different tools are available to help you perform configurations or customize the display.

The following are examples of some of the tools for customizing the display:

- Button  : Display fields in the polar coordinates system.

- Button  : Change over the view of the device/fields. Device black: View from above, device blue: View from below.
- Button  or  : Switch off display of the full measuring line or display measuring line as dotted.

7.1.1 Parameter - network

The network area has input screens for configuring the Ethernet connection, the digital inputs / digital outputs and the device.

For Ethernet configuration, note that the IP-Address / subnet-mask must correspond with the address space of the subsequent application.

Table 13: IP-Address factory setting

Parameter	Value
IP-Address	`192.168.0.1`
Subnet mask	`255.255.255.0`

NOTE

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

7.1.2 Parameter - filters

When selecting the filter, consider filter mutual interference, see "Filter", page 19.

7.2 Field evaluation

If the field shape of field set 1 has been taught in without a computer using the teach-in pushbutton, SOPAS is generally used to continue the configuration.

This includes the following:

- Setting of field shapes/sizes and, if applicable, non-programmable field sets based on the default setting and the setting of response time of the fields
- Setting of the blanking size and the holding time of the assigned digital outputs OUT 1 ... OUT 3

7.2.1 Teach-in / Field set selection

For the field evaluation application, it is possible to configure the device as follows rather than configuring it via SOPAS:

- Selection of one of 16 field sets, each with 3 fields of the same size nested into one another, by means of input wiring.
- Teach-in of the surrounding contour to automatically generate the outer field with any shape, including more complex shapes, and to deduce the two inner fields.

The field sets are organized into groups in segmented field shapes. The shape can be changed at will, the default is rectangular.

In the factory setting, the 3 origin-oriented fields of a field set are nested inside one another. The size of field 2 corresponds to field 1 plus 25%. The size of field 3 corresponds to field 1 plus 52%.

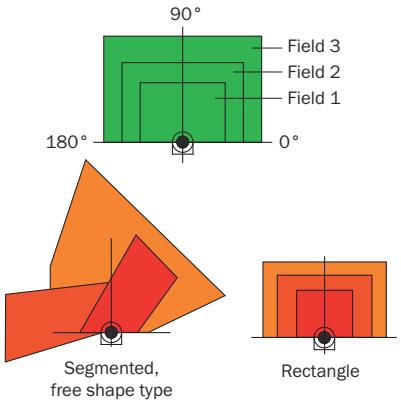


Figure 23: Structure of the fields of a field set and possible field shapes

① Segmented, free field shape

② Nested together, rectangle

7.2.1.1 Teach-in

Preparation

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

Teaching in the field contour

The device uses field set 1 (segmented, initial shape: rectangle) to adjust the field shape and size to the surrounding contour that was detected. The digital inputs are not allowed to be supplied with current during this process.

The device forms the outer field 3 from the surrounding contour with a negative offset of 100 mm, and derives from this the limits of the two inner fields: field 1 = field 3 minus 33%, and field 2 = field 3 minus 17%.

The device stores the shortest value measured during the teach-in phase as a field limit for each angle.

A parameter upload is required in order to display the newly taught-in field contour in SOPAS.

1. Press and hold the Teach-in pushbutton for 3 s to start the teach-in process.
2. The field shape to be formed can be defined by pacing out the limits during the teach-in phase. Do not wear black clothing during this process!

The LEDs display the individual phases of the teach-in process:

LED (red)	LED (green)	Phase
-		**Teach-in pre-warning:** LED flashes slowly at first (0.5 Hz) LED flashes increasingly rapidly within 15 s
		**Teach-in:** Duration 60 seconds
-		**Teach-in completion:** LED flashes increasingly rapidly within 15 s

LED (red)	LED (green)	Phase
-	●	**Monitoring mode:** Automated return after teach-in complete All fields free
●	●	**Monitoring mode:** If an object detection has occurred

● = illuminated; ☼ = flashing

The taught-in field contour is automatically and permanently saved as a new field set 1.

7.2.1.2 Field set selection via input wiring

One of the predefined field sets can be activated by wiring the digital inputs.

To select field sets via digital inputs, the **Digital inputs** field set selection method must be set in the device. This is the case by default.

Field set factory settings – digital inputs

Field set	Digital inputs				Field shape Default size of field 1
	IN 1	IN 2	IN 3	IN 4	
1	0	0	0	0	Rectangle, segmented L: 1 m, W: 2 m
2	1	0	0	0	Rectangle, segmented L: 1.25 m, W: 2 m
3	0	1	0	0	Rectangle, segmented L: 1.5 m, W: 2 m
4	1	1	0	0	Rectangle, segmented L: 1.75 m, W: 2 m
5	0	0	1	0	Rectangle, segmented L: 1 m, W: 2 m
6	1	0	1	0	Rectangle, segmented L: 1.25 m, W: 2 m
7	0	1	1	0	Rectangle, segmented L: 1.5 m, W: 2 m
8	1	1	1	0	Rectangle, segmented L: 1.75 m, W: 2 m
9	0	0	0	1	Rectangle, segmented L: 1 m, W: 2 m
10	1	0	0	1	Rectangle, segmented L: 1.25 m, W: 2 m
11	0	1	0	1	Rectangle, segmented L: 1.5 m, W: 2 m
12	1	1	0	1	Rectangle, segmented L: 1.75 m, W: 2 m
13	0	0	1	1	Rectangle, segmented L: 1 m, W: 2 m
14	1	0	1	1	Rectangle, segmented L: 1.25 m, W: 2 m
15	0	1	1	1	Rectangle, segmented L: 1.5 m, W: 2 m
16	1	1	1	1	Rectangle, segmented L: 1.75 m, W: 2 m
Output form (rectangle), can be changed as desired, L = length, W = width					

Input level: Low (in resting position): ≤ 2 V, high (in working position): ≥ 8 V

7.2.1.3 Field set selection via telegram

One of the predefined field sets can be activated by means of a telegram sent via the Ethernet interface on the device. To select field sets via telegram, the “SOPAS telegram” field set selection method must be set in the device. This can be done using a telegram or via settings in SOPAS ET. The device can then be instructed via a telegram of the predefined field set that is to be activated. Field set 1 is preselected by default.

Overview of the telegram functionality for field set selection:

- Select the field set selection method (`sWN FieldSetSelectionMethod [0;1]`)
- Read the field set selection method (`sRN FieldSetSelectionMethod`)

- Select the active field set (`sWN ActiveFieldSet [0x01…0x10]`)
- Read the active field set (`sRN ActiveFieldSet`)

NOTE
For the detailed structure of the telegrams, see "Telegram listing (EN)", page 56.

7.2.2 Monitor - field evaluation monitor

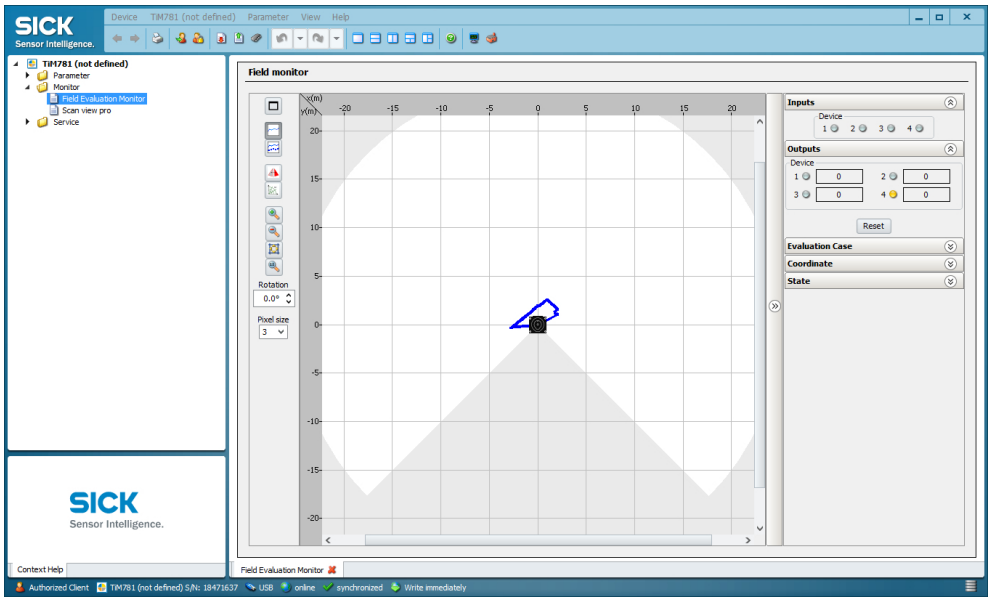


Figure 24: Device window: Monitor - field evaluation monitor

In the **Field evaluation monitor** window, SOPAS displays the field contour (scan line) currently seen by the device through ambient reflection in blue.

If the 4 digital inputs are not energized, SOPAS displays the following for field set 1 according to the default setting of the device:

- 3 evaluation fields (segmented rectangles) or the field shape generated with the aid of teach-in with its dimensions
- Status of the digital inputs/outputs
- Position of the mouse pointer

SOPAS displays the fields as green if no object detection is present.

If objects of a certain size are located for a certain duration in the part of the visual range that is covered by fields, the device will recognize this as an object detection. SOPAS displays this separately in yellow for the individual fields.

Click the “Reset” button to make SOPAS reset the digital output counters.

7.2.3 Parameter - evaluation cases

Response time, blanking size and digital output holding time

The blanking size is the smallest size from which an object can be detected in a field by the device and lead to an object detection. All objects that are smaller than the minimum size are blanked out.

Like the response time and the holding time, the blanking size applies to all field sets and their fields.

Objects that are smaller than the blanking size can cause shadows to be cast at close range. Objects that are larger than the blanking size may be located within this shadow. These may not be detected correctly.

Parameter	Factory setting
Blanking size	Cross-section 200 mm
Response time of the fields	335 ms (5 scans)
Digital output holding time	335 ms (5 scans)

When selecting the response time, note that the device internal reaction time must also be added (max. 67 ms).

To test the effects of the changed settings, change to the **Field evaluation monitor** view.

If the changed fields have been transferred to the device as described, SOPAS will also display these in the monitor, displaying the infringed fields in yellow. If you wish to observe another field set, it must first be activated accordingly using the digital inputs.

7.2.4 Parameters - detection fields

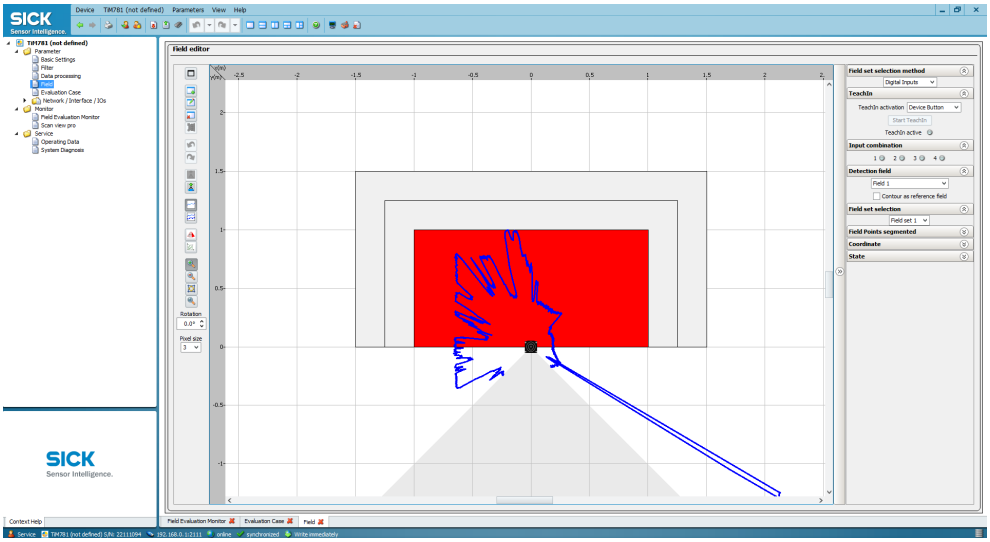


Figure 25: Device window: Parameters - detection fields

The user can change parameters in the right part of the window. SOPAS immediately transfers these changes to the device (default setting).

However, detection fields that have been changed in size and shape must always be manually transferred to the device using the button. All changed parameters are only temporarily stored in the device for the time being and are not stored on the computer.

In order to optimize the dimensions of the detection fields:

1. Under Field selection, for example, select field set 1.
2. Select the field to be configured.
3. Make the required optimizations, see description in the following sections.

Shifting field positions

1. Button .
2. Click on the green marking rectangle of the desired field position in the outer field.

✓ The color of the marker square changes to blue.

3. Re-click the rectangle and drag it to the desired position, then release the push-button.

✓ SOPAS controls the available positioning area during shifting.

Inserting additional field positions

1. Button .
2. Click on the desired position on the limits of the outer field.

✓ SOPAS inserts a new green marker square. This can also be shifted.

Deleting field points

1. Button .
2. Click on the green marking rectangle of the field position to be deleted in the outer field.

✓ The color of the marking rectangle changes to red.

3. Re-click the marking rectangle.

✓ SOPAS removes the marker square and instead connects the two nearest marker squares with a new line.

Rotating the field pair and device around the central axis

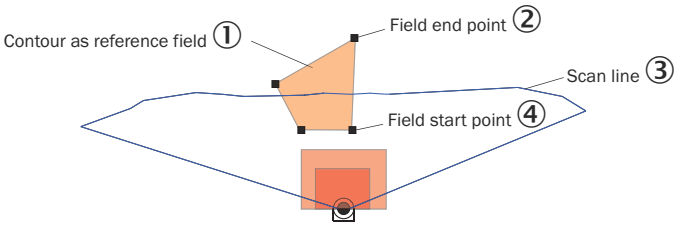
► In order to align the position of the field pair in SOPAS to the conditions on site from the user´s perspective, enter and confirm the desired angle of rotation in the 0.0° input. A negative “-” sign sets the turning direction to the right.

Setting up the contour as reference field

In each field set, any field can be selected as a contour as reference field.

The contour as reference field is used to monitor contours. This can be used to check whether the background is still recognized correctly. If this is no longer the case, possible reasons are that the device is covered or rotated.

The same evaluation strategy is used for all contour as reference fields (evaluation time and blanking size).



① Contour as Reference field
② End point of contour
③ Scanning line
④ Start point of contour

1. Select the required field in the field set.
2. Select the **contour as reference field** check mark under **Field selection**.
3. Delete the two field positions closest to the device.
4. Mark the two remaining field positions and shift them outside of the required reference contour area.
5. The button can switch to the configuration for the contour as reference field start points.
   The field starting points must lie between the scan line and the device so that the scan line passes between the start and end points of the contour as reference field.
6. Depending on the required shape of the reference contour field, create additional field points between the device and the reference contour field. The distance between the start and end points of the contour as reference field should be approximately 20 cm.

The button can be used to switch between the start and end points of the contour as reference field for editing purposes. The active points are highlighted in light green while the deactivated points are dark green.

7.3 Measurement data output

7.3.1 Monitor - scan view pro

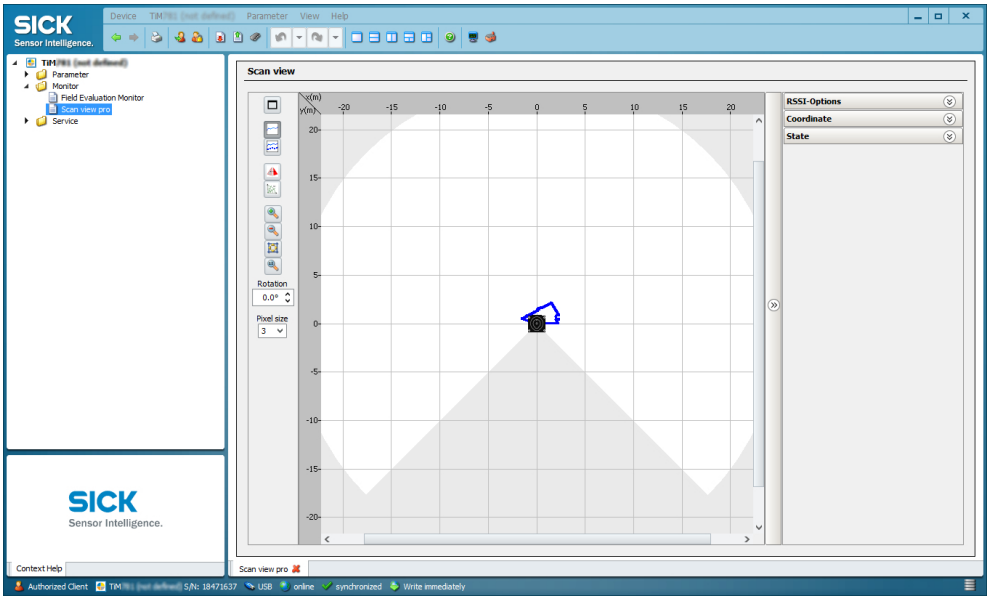


Figure 26: Device window: Monitor - scan view pro

In the Pro scan display window, SOPAS displays the field contour (scan line) currently seen by the device through ambient reflection in blue.

The user can change parameters in the right part of the program window under Settings & Device Status. SOPAS immediately transfers these changes to the device (default setting).

7.3.2 Parameter - data processing

Data output format of the measured values

The data output format per scan is comprised of the measured values (radial distance, RSSI signal level), device and status information and time stamp.

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

In order to output remission values in the telegram, select the RSSI checkbox.

To display the remission values in the scan as well, select the RSSI... checkbox under “Monitor” > “scan view pro”.

Output range of the measured values

The device scans an angular range of 270° (-45° to 225°) and outputs one measured value per angular increment (0.33° or 1° depending on the device type).

The angle range for which measured values can be output can be set using the Output range input field (resolution corresponds to the angular resolution of the device).

7.3.3 Telegram

If the device receives one of the two following commands by telegram over the Ethernet or USB interface, it will start the output of measured values in real time over these data interfaces.

The detailed construction of the output telegram as well as the flow of requests is described in see "Telegram listing (EN)", page 56.

One-off output of measured values:

Telegram structure: sRN LMDscandata

Telegram part	Description	Variable type	Length (Byte)	Value range
Command type	Request (SOPAS read by name)	String	3	sRN
Command	Request data	String	11	LMDscandata

Example:

Telegram type	Command
ASCII	<STX>sRN{SPC}LMDscandata<ETX>
HEX	02 73 52 4E 20 4C 4D 44 73 63 61 6E 64 61 74 61 03
Binary	02 02 02 02 00 00 00 0F 73 52 4E 20 4C 4D 44 73 63 61 6E 64 61 74 61 05

Continuous output of measured values:

Telegram structure: sEN LMDscandata measurement start/stop

Telegram part	Description	Variable type	Length (Byte)	Value range
Command type	Request (SOPAS event by name)	String	3	sEN
Command	Request data	String	11	LMDscandata
StartStop measure-ment	Starting/stopping the output of measured values	Enum8	1	0: Stop 1: Start

Example:

Telegram type	Command
ASCII	<STX>sEN{SPC}LMDscandata{SPC}1<ETX>
HEX	02 73 45 4E 20 4C 4D 44 73 63 61 6E 64 61 74 61 20 31 03
Binary	02 02 02 02 00 00 00 11 73 45 4E 20 4C 4D 44 73 63 61 6E 64 61 74 61 20 01 33

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

1. Start the terminal emulator with the button.
2. Select the Connect... command in the Connections menu in the dialog window and establish communication with the TiM over the Ethernet or USB interface.
3. Enter one of the two telegrams in the "Send telegram" input line as they appear (automatically framed by STX and ETX when sending in the default setting). Pay attention to blank characters in the string.
4. Use the button to transmit the telegram to the device.

✓ The device responds by providing the data as a one-off or continuously in the

display area of the terminal emulator.

7.4 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).

8 Maintenance

8.1 Maintenance plan

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

Table 14: Maintenance plan

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

8.2 Cleaning

NOTICE

Equipment damage due to improper cleaning.

Improper cleaning may result in equipment damage.

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

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

9 Troubleshooting

9.1 Detecting and displaying errors

Error memory

The device has an error memory where its internal error states are recorded. The content of the error memory is retained when the device is switched off and when the “Restore Factory Settings” function is used.

The SOPAS ET software can be used to create a diagnostics report containing error information (**Complete** view, **Configuration** tab, **Start Diagnosis** button).

9.2 Resetting the password for the Service user level

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

NOTE

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

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

Requesting an unlock code

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

✓ The **Reset password** window appears.

4. Enter the relevant data.

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

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

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

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

Entering the unlock code

Prerequisite

- SICK has sent an unlock code.

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

✓ The **Reset password** window appears.

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

6. Click **Ok**.
✓ Password has been reset to the default password `servicelevel`. Parameters are not changed.

Assigning a new password for the Service user level

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

9.3 Repairs

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

9.4 Returns

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

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

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

9.5 Disposal

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

NOTICE
Danger to the environment due to improper disposal of the device.
Disposing of devices improperly may cause damage to the environment.
Therefore, observe the following information:

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

10 Technical data

NOTE

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

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

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

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

Please note: This documentation may contain further technical data.

10.1 Features

Variant	TiM781
Measurement principle	HDDM+ (improved behavior for edge hits)
Application	Outdoor
Light source	Infrared (wavelength 850 nm, max. pulse power 880 mW, max. pulse width 5 ns, pulse frequency 1,500 kHz)
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 8 May 2019.
Aperture angle	270° (horizontal)
Scan field flatness [1]	Typ. ± 1.5°
Scanning frequency	15 Hz
Angular resolution	0.33°
Working range	0.05 m ... 25 m
Scanning range for 5% remission	5 m (typical)
Sensing range for 10% remission	8 m (typical) [2]
Distance resolution	1 mm
Spot size	Spot size on the optics cover: 7 mm Divergence: 8.6 mrad (0.49°)
Number of echoes evaluated	1
Scan/frame rate	12195 measurement point(s)

1) Reference area for base of housing

2) at ambient temperature < -15 °C: typ. 7.5 m

Working range diagram

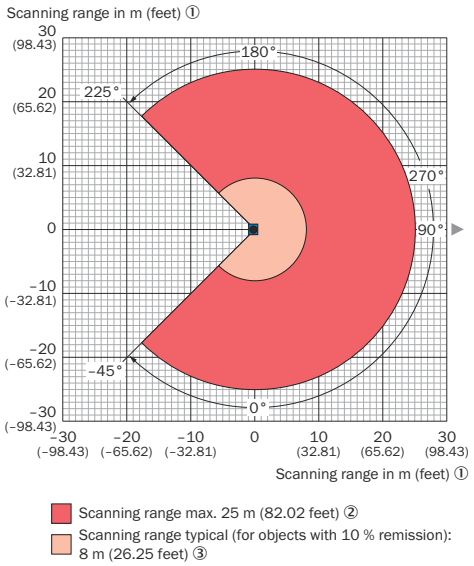


Figure 27: TiMx1x / TiMx2x working range diagram

① Sensing range in meters (feet)

② Maximum sensing range: 25 m (82.02 feet)

③ Typical sensing range for objects with 10% remission: 8 m (26.25 feet)

10.2 Performance

Variant	TiM781
Response time	Typ. 1 scan: 67 ms Max. 2 scans: 134 ms [1]
Power-up delay	Typ. 20 s
Detectable object shape	Almost any
Measurement error	Statistical (1 σ): < 20 mm [2] Systematic: ± 60 mm [2] Temperature drift: 0.5 mm/K
Integrated application	Field evaluation with flexible fields Measurement data output (radial distance, RSSI signal level, device and status information, time stamp)
Number of field sets	16 triple fields, (of 1 triple (3 fields) can be configured directly at the scanner)
Simultaneous evaluation cases	1 (3 fields) 2 (2 fields for detection and 1 field for contour as reference)
Filter	Edge filter Particle filter Median filter Average filter

1) Corresponds to max. 134 ms between +45° and +225° of the working range, max. 150 ms between -45° and +45° of the working range (see "Working range diagram", page 52).

2) Typical value at room temperature and up to a sensing range of 10 m at 90% remission, real value depends on ambient conditions, temperature drift: 0.5 mm/K

10.3 Interfaces

Variant	TiM781
Ethernet	TCP/IP Max. data transmission rate: 10 Mbit and 100 Mbit Length of cable: max. 100 m
USB	Type: Micro-USB Function: AUX, configuration
Digital inputs	4 (PNP, for field switching)
Digital outputs	3 (PNP, additional 1x “device ready”)
Delay time	67 ms ... 30,000 ms (can be configured)
Holding time	67 ms ... 10,000 ms (can be configured)
Optical indicators	2 LEDs (on, “device ready”)
Control elements	Pushbutton (teach-in)

10.4 Mechanics/electronics

Variant	TiM781
Electrical connection	1 x Ethernet connection, 4-pin M12 female connector 1 x voltage supply connection, 12-pin M12 male connector 1 x micro USB female connector, type B
Supply voltage	9 V DC ... 28 V DC SELV and PELV acc. to IEC 60364-4-41
Power consumption	4 W (typical), with unloaded digital outputs, incl. start-up current Max. 16 W with 4 loaded digital outputs
Input voltage	Low: $V_{in} \leq 2$ V; $I_{in} \leq 0.3$ mA High: 8 V $\leq V_{in} \leq$ 32 V; 0.7 mA $\leq I_{in} \leq$ 5 mA
Output voltage	Low: 0 V $\leq V_{out} \leq 2$ V High: $(V_S - 2\text{ V}) \leq V_{out} \leq V_S$; $I_{out} \leq 100$ mA
Housing	Base part: Aluminum die cast Optics cover: Polycarbonate with scratch-resistant coating
Housing color	Gray (RAL 7032)
Electrical safety	According to IEC 61010-1 (ed.3)
Enclosure rating	IP67 (IEC 60529:1989+AMD1:1999+AMD2:2013), only valid with closed “Aux interface” plastic cover
Protection class	III (IEC 61140:2016-1)
Weight	250 g, without connecting cables
Dimensions (L x W x H)	60 mm x 60 mm x 86 mm
$MTTF_D$	Mean time to dangerous failure: 100 years, at 25 °C ambient temperature (EN ISO 13849-1:2015)

10.5 Ambient data

Variant	TiM781
Object remission	4% ... 1,000% (reflectors)
Electromagnetic compatibility (EMC)	IEC 61000-6-3:2006+AMD1:2010, IEC 61000-6-2:2005

Variant	TiM781
Vibration resistance	**Sine resonance scan:** 10 Hz ...1,000 Hz (IEC 60068-2-6:2007) **Sine test:** 10 Hz ... 500 Hz; 5 g; 10 frequency cycles (IEC 60068-2-6:2007) **Noise test:** 10 ... 250 Hz; 4.24 grms, 5 h (IEC 60068-2-64:2008)
Shock resistance	50 g; 11 ms; 6 shocks/axis 25 g; 6 ms; 2,000 shocks/axis 50 g; 3 ms; 10,000 shocks/axis (IEC 60068-2-27:2008)
Impact resistance	IK05 (0.7 joule) and IK06 (1 joule) as per DIN EN 62262:2022-02
Ambient temperature	Commissioning/switching on: –10 °C ... +50 °C Operation: –25 °C ... +50 °C Storage: –40 °C ... +75 °C
Ambient humidity	Operation: ≤ 80%, non-condensing ≤ 95%, non-condensing (EN 60068-2-30:2005)
Ambient light immunity	80,000 lx (indirect exposure)
Altitude	< 5,000 m above sea level
Ambient conditions	Contamination level 3 outside the housing (EN 61010-1:2011-07)
Damp heat	+25 °C ... +55 °C, 95% r.h., 6 cycles (EN 60068-2-30:2005)
Temperature change	-25 °C... +50 °C, 10 cycles (EN 60068-2-14: 2009)

11 Accessories

NOTE

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

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

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

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

12 Annex

12.1 Declarations of conformity and certificates

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

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

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

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

12.2 Licenses

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

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

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

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

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

12.3 Telegram listing (EN)

Contents

12.3.1.1 About this document

12.3.1.1.1 Information on the telegram listing

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

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

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

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

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

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

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

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

The devices generally support automatic IP address discovery.

Default IP address is:
- 192.168.0.1

Subnet mask is 255.255.255.0.

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

The telegram listing does not or only in a few exceptional cases differentiate between individual device versions or sub product families such as LMS5xx Lite and LMS5xx PRO. Most parameter changes also require certain user levels. Additionally, commands may change during the product lifecycle and development process with a new firmware.

This document is based on the following firmware version (or newer):
- V2.51

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

12.3.1.1.2 Explanation of symbols

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

Telegram to device

Telegram from device

Unformatted example to copy and paste

12.3.1.2 Communication format

12.3.1.2.1 Binary telegram (CoLa B)

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

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

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

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

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

Table 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

12.3.1.2.2 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)

12.3.1.2.3 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

Variable type	Length (byte)	Value range	Sign
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-depend-ent	Strings are not terminated in zeroes	
FlexString	array of visible characters with preceeding cur-rent 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 Basic-Type, Structs again or an Array.
Array	An Array is a repetition of a type. The length of the array is defined with each Array. The types can be of a BasicType, a Struct or an Array again (n-dimensional).
Flex Array	A FlexArray is a repetition of a type with a variable length. The maximum length of the array is defined with each FlexArray. The current length of the FlexArray is transferred as a UInt preceeding the Array itself. The types can be of a BasicType, a Struct or an Array again (n- dimensional).

12.3.1.2.4 Command basics

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

Every response telegram starts with a separat framed string:

<STX>sSI 2 1<ETX><STX>“Answer”<ETX>

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

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

NOTE
Every write command (sWN) has a read (sRN) counter part even if it may not be described in the Telegram Listing. You can verify the current parameters this way, e.g. read the current IP address via `sRN EIIpAddr` see "Read IP address [sRN EIIpAddr]", page 104.

12.3.1.2.5 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 64. 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 71.

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

12.3.1.3 Workflows

12.3.1.3.1 Parameterize the scan

Log in: `sMN SetAccessMode` see "Log in [sMN SetAccessMode] ", page 64

Configure scandata content: `sWN LMDscandatacfg` see "Configure the data content for the scan [sWN LMDscandatacfg]", page 73

Configure scandata output: `sWN LMPoutputRange` see "Configure aperture angle of the scandata for output [sWN LMPoutputRange]", page 74

Store parameters: `sMN mEEwriteall` see "Save parameters permanently [sMN mEEwriteall]", page 70

Log out: `sMN Run` see "Set to run [sMN Run]", page 71

Request scan:

`sRN LMDscandata` see "Poll one telegram [sRN LMDscandata]", page 76

`sEN LMDscandata` see "Send data permanently [sEN LMDscandata]", page 77

(Device output ...)

12.3.1.3.2 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.

"Log in [sMN SetAccessMode] ", page 64
"Load factory defaults [sMN mSCloadfacdef]", page 66
"Load application defaults [sMN mSCloadappdef]", page 67
"Change password [sMN SetPassword]", page 68
"Check password [sMN CheckPassword]", page 70
"Reboot device [sMN mSCreboot]", page 72

12.3.1.4 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"

12.3.1.4.1 Log in [sMN SetAccessMode]

NOTE

- Please note that the laser is shut off after a succesful log in and no measurement data is created any more. The laser is turned on again after log out (see "Set to run [sMN Run]", page 71).
- An automated hash-value calculator can be found in SOPAS ET under menu “password”. Required userlevel 'Service' (see "Change password [sMN SetPassword]", page 68).

A log in to the device is necessary to change parameters. In most cases, the user level 'Authorized client' is needed. Changed parameters will be reset to the previous state via a reboot unless the are saved. To save parameter changes the command "sMN mEEwriteall" (see "Save parameters permanently [sMN mEEwriteall]", page 70) 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

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

12.3.1.4.2 Basic Settings

12.3.1.4.2.1 Read for frequency and angular resolution [sRN LMPscancfg]

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

Table 25: Telegram structure: sRN LMPscancfg

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

Table 26: Example: sRN LMPscancfg

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

Table 27: Telegram structure: sRA LMPscancfg

Telegram structure: sRA LMPscancfg						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sRA	73 52 41
Command	Info of scan frequency and angular resolution	String	10		LMPscancfg	4C 4D 50 73 63 61 6E 63 66 67
Scan frequency	[1/100 Hz]	Uint_32	4	15 Hz:	5DCh 7D0h	00 00 05 DC
Reserved	-	Int_16	2	Always:	1	00 01
Angular resolution	[1/10000°]	Uint_32	4	0.333°:	D05h	00 00 0D 05
Start angle	[1/10000°]	Int_32	4	-45°	FFF92230h	FF F9 22 30
Stop angle	[1/10000°]	Int_32	4	+225°	225510h	00 22 55 10

Table 28: Example: sRA LMPscancfg - 15 Hz scan frequency, 0.33° angular resolution, -45° start angle, +225° stop angle

CoLa A	<STX>sRA{SPC}LMPscancfg{SPC}5DC{SPC}1{SPC}D05{SPC}FFF92230{SPC}225510<ETX>
	02 73 52 41 20 4C 4D 50 73 63 61 6E 63 66 67 20 35 44 43 20 31 20 44 30 35 20 46 46 46 39 32 32 33 30 20 32 32 35 35 31 30 03
CoLa B	02 02 02 02 00 00 00 21 73 52 41 20 4C 4D 50 73 63 61 6E 63 66 67 20 00 00 05 DC 00 01 00 00 0D 05 FF F9 22 30 00 22 55 10 EF

12.3.1.4.2.2 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 29: 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 30: 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 31: 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 32: Example: sAN mSCloadfacdef

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

12.3.1.4.2.3 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 33: 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

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	Load application defaults	String	13		mSCloadappdef	6D 53 43 6C 6F 61 64 61 70 70 64 65 66

Table 34: 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 35: Telegram structure: sAN mSCloadappdef

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

Table 36: Example: sAN mSCloadappdef

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

12.3.1.4.2.4 Change password [sMN SetPassword]

Changing the log in password for a specific user level.

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

Table 37: Telegram structure: sMN SetPassword

Telegram structure: sMN SetPassword (Same user level or higher required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3		sMN	73 4D 4E
Command	Set password request	String	13		SetPassword	53 65 74 50 61 73 73 77 6F 72 64

Telegram structure: sMN SetPassword (Same user level or higher required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
User level	User level that the password will be applied to	Int_8	1	Maintenance: Authorized client: Service:	2 3 4	02 03 04
Password	Hash value of the new password	Uint_32	4		<Hash value>	

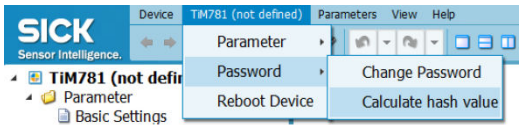
Set password for 'Authorized client' to “testtest” (hash value = 1920E4C9).

Table 38: Example: sMN SetPassword

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

Calculating the hash value of the password

- ► Login in SOPAS ET with user level “Service” to the device.
- ► Select **[Device]** > **Password** > **Calculate Hash value.**



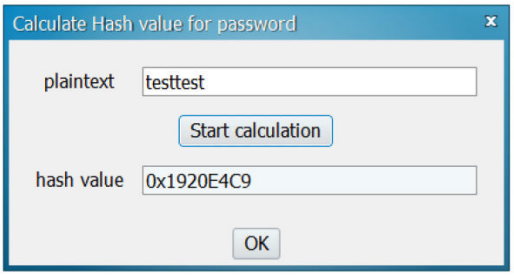


Table 39: Telegram structure: sAN SetPassword

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

Table 40: Example: sAN SetPassword

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

12.3.1.4.2.5 Check password [sMN CheckPassword]

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

Table 41: 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 42: 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 43: Telegram structure: sAN CheckPassword

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

Table 44: Example: sAN CheckPassword

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

12.3.1.4.2.6 Save parameters permanently [sMN mEEwriteall]

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

Table 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

12.3.1.4.2.7 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
Command	Start the device	String	3		Run	52 75 6E
Status code	Accepted when value is 1	Bool_1	1	Error: Success:	0 1	00 01

Table 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

12.3.1.4.2.8 Reboot device [sMN mSCreboot]

This command includes saving all parameters.

Table 53: Telegram structure: sMN mSCreboot

Telegram structure: sMN mSCreboot (User level 'Service' 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

12.3.1.4.3 Measurement output telegram

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

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

Table 57: Telegram structure: sWN LMDscandatacfg

Telegram structure: sWN LMDscandatacfg (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3		sWN	73 57 4E
Command	Configure scandata	String	14		LMDscanda-tacfg	4C 4D 44 73 63 61 6E 64 61 74 61 63 66 67
Reserved	-	Uint_8	2	Always:	1 0	01 00
RSSI	Enable or disable the output of RSSI values	Uint_8	1	RSSI disabled: RSSI enabled:	0 1	00 01
Resolution	Resolution of RSSI	Enum_8	1	8 Bit: 16 Bit:	0 1	00 01
Unit	Unit of RSSI	Enum_8	1	Digits:	0	00
Reserved	-	Uint_8	2	Always:	0 0	00 00
Reserved	-	Bool_1	1	Always:	0	00
Device name	Sends the device name	Bool_1	1	No: Yes:	0 1	00 01
Reserved	-	Bool_1	1	Always:	0	00
Reserved	-	Bool_1	1	Always:	0	00
Output rate	Sends the output rate	Uint_16	2	All scans: Each 2nd scan: Max. Each 50000th scan:	+1d (1h) +2d (2h) +50000d (C350h)	00 01 00 02 C3 50

Example: RSSI enabled, resolution 8Bit, send device name and output rate all scans

Table 58: Example: sWN LMDscandatacfg

CoLa A	<STX>sWN{SPC}LMDscandatacfg{SPC}1{SPC}0{SPC}1{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}1{SPC}0{SPC}0{SPC}+1<ETX>
	<STX>sWN LMDscandatacfg 1 0 1 0 0 0 0 0 1 0 0 +1<ETX>
	sWN LMDscandatacfg 1 0 1 0 0 0 0 0 1 0 0 +1
	02 73 57 4E 20 4C 4D 44 73 63 61 6E 64 61 74 61 63 66 67 20 31 20 30 20 31 20 30 20 30 20 30 20 30 20 30 20 30 20 31 20 30 20 30 20 30 20 2B 31 03
CoLa B	02 02 02 02 00 00 00 20 73 57 4E 20 4C 4D 44 73 63 61 6E 64 61 74 61 63 66 67 20 01 00 01 00 00 00 00 00 01 00 00 00 01 43
	73 57 4E 20 4C 4D 44 73 63 61 6E 64 61 74 61 63 66 67 20 01 00 01 00 00 00 00 00 01 00 00 00 01

Table 59: Telegram structure: sWA LMDscandatacfg

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

Table 60: Example: sWA LMDscandatacfg

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

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

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

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

Table 61: Telegram structure: sWN LMPoutputRange

Telegram structure: sWN LMPoutputRange (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional Details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3		sWN	73 57 4E
Command	Change output angle range	String	14		LMPoutputRange	4C 4D 50 6F 75 74 70 75 74 52 61 6E 67 65
Reserved	-	Int_16	2	Always:	1	00 01

Telegram structure: sWN LMPoutputRange (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional Details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Angular resolution	[1/10000°] Fixed value, not changeable	Uint_32	4	0.333°:	+3333d (D05h)	00 00 0D 05
Start angle	[1/10000°]	Int_32	4	-45°:	-450000d (FFF92230h)	FF F9 22 30
Stop angle	[1/10000°]	Int_32	4	+225°	+2250000d (225510h)	00 22 55 10

Table 62: Example: sWN LMPoutputRange - set output data for angular resolution at 0.333° and range from 0° to 90°

CoLa A	<STX>sWN{SPC}LMPoutputRange{SPC}1{SPC}D05{SPC}0{SPC}DBBA0<ETX>
	<STX>sWN LMPoutputRange 1 D05 0 DBBA0<ETX>
	sWN LMPoutputRange 1 D05 0 DBBA0
	02 73 57 4E 20 4C 4D 50 6F 75 74 70 75 74 52 61 6E 67 65 20 31 20 44 30 35 20 30 20 44 42 42 41 30 03
CoLa B	02 02 02 02 00 00 00 21 73 57 4E 20 4C 4D 50 6F 75 74 70 75 74 52 61 6E 67 65 20 00 01 00 00 0D 05 00 00 00 00 00 0D BB A0
	73 57 4E 20 4C 4D 50 6F 75 74 70 75 74 52 61 6E 67 65 20 00 01 00 00 0D 05 00 00 00 00 00 0D BB

Table 63: Telegram structure: sWA LMPoutputRange

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

Table 64: Example: sWA LMPoutputRange

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

12.3.1.4.3.3 Read for actual output range [sRN LMPoutputRange]

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

Table 65: Telegram structure: sRN LMPoutputRange

Telegram structure: sRN LMPoutputRange						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E

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

Table 66: Example: sRN LMPoutputRange

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

Table 67: Telegram structure: sRA LMPoutputRange

Telegram structure: sRA LMPoutputRange						
Telegram part	Description	Variable	Length	Additional details	Values CoLa A (ASCII)	Values CoLa B (Binary)
Command type	Answer	String	3		sRA	73 52 41
Command	Output range	String	14		LMPoutpu-tRange	4C 4D 50 6F 75 74 70 75 74 52 61 6E 67 65
Reserved		Int_16	2	Always:	1h	00 01
Angular resolution	[1/10000°]	Uint_32	4	0.333°:	D05h	00 00 0D 05
Start angle	[1/10000°]	Int_32	4	-45°	FFF92230h	FF F9 22 30
Stop angle	[1/10000°]	Int_32	4	+225°	225510h	00 22 55 10

Table 68: Example: sRA LMPoutputRange – device output set at 0.333° angular resolution and range from 0° to 90°

CoLa A	<STX>sRA{SPC}LMPoutputRange{SPC}1{SPC}D05{SPC}0{SPC}DBBA0<ETX>
	02 73 52 41 20 4C 4D 50 6F 75 74 70 75 74 52 61 6E 67 65 20 31 20 44 30 35 20 30 20 44 42 42 41 30 03
CoLa B	02 02 02 02 00 00 00 21 73 52 41 20 4C 4D 50 6F 75 74 70 75 74 52 61 6E 67 65 20 00 01 00 00 0D 05 00 00 00 00 00 0D BB A0 6E

12.3.1.4.3.4 Poll one telegram [sRN LMDscandata]

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

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

Table 69: Telegram structure: sRN LMDscandata

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

Table 70: 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 71: 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 77.						

12.3.1.4.3.5 Send data permanently [sEN LMDscandata]

Start/ stop continuous retrievement of measurement data from device. Data will be transmitted as configured in "Configure the data content for the scan [sWN LMDscandatacfg]", page 73 as soon as measurement data is generated by the device. No data is generated when there is an acitve log in, the laser is shut off (e.g. in Standby mode), the motor is stopped or in case of certain error modes (e.g. Device not ready).

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

Table 72: 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 73: 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 74: Telegram structure: sEA LMDscandata

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

Table 75: 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 76: Telegram structure: Datastream of sRA LMDscandata/sSN LMDscandata

Telegram structure: sRA LMDscandata / sSN LMDscandata						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	Answer to sRN LMDscandata: Answer to sEN LMDscandata:	sRA sSN	73 52 41 73 53 4E
Command	Data telegram	String	11		LMDscandata	4C 4D 44 73 63 61 6E 64 61 74 61
Version number	For detecting format changes by the version.	Uint_16	2		1h ... FFFFh	00 01 ... FF FF
Device						
Device number	Defined with SOPAS	Uint_16	2		0h ... FFFFh	00 00 ... FF FF
Serial number	Production period (year, calendar week, number): YYWWxxxx	Uint_32	4		0h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
Device status	(See values column)	Uint_8	2 x 1	Ok: Error:	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
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]", page 74)	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:	00000000h	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]", page 74)	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

Telegram structure: sRA LMDscandata / sSN LMDscandata						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Reserved						
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 77: 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	01 51 63 76
Device status	(See values column)	Uint_8	2 x 1	Ok:	0 0	00 00
Status info						
Telegram counter	Number of measurement telegrams finished in the scanner and given to the interface. Does not count how many telegrams were really given out; is relevant if not all scans are delivered from the scan core.	Uint_16	2		C4C6h	C4 C6
Scan counter	Number of scans which were created in the device; counts how many scans were really done.	Uint_16	2		C4E3h	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	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	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	5DCh	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	00 00 00 A2
Reserved						
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:	00000000	00 00 00 00
Start angle	[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	0D 05
Amount of data	Number of transmitted measurement points	Uint_16	2	16 Measurement points	10h	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	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 78: 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

12.3.1.4.4 Filters

12.3.1.4.4.1 Set particle filter [sWN LFPparticle]

Filter out disturbances in the meausurement data caused by particles such as dust, snow flakes or similar see "Particle filter", page 20.

Table 79: 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[2)]	Particle threshold in mm	Uint_16	2	(must be taken)	+500d (1F4h)	01 F4

1) Never change the threshold here, it is taken by the device to handle the particles.

Table 80: 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	

2) 1)

Table 81: 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 82: Example: sWA LFPparticle

CoLa A	<STX>sWA{SPC}LFPparticle<ETX>
	02 73 57 41 20 4C 46 50 70 61 72 74 69 63 6C 65 03
CoLa B	02 02 02 02 00 00 00 10 73 57 41 20 4C 46 50 70 61 72 74 69 63 6C 65 20 2B

12.3.1.4.4.2 Set average filter [sWN LFPmeanfilter]

This filter smooths the distance value by calculating the arithmetic average from the configured number of scans see "Average filter", page 21.

Table 83: Telegram structure: sWN LFPmeanfilter

Telegram structure: sWN LFPmeanfilter (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3		sWN	73 57 4E
Command	Set average filter	String	13		LFPmeanfilter	4C 46 50 6D 65 61 6E 66 69 6C 74 65 72
Status code	Code number	Bool_1	1	Inactive: Active:	0 1	00 01
Number of scans	Number	Uint_16	2		+2d ... +4d (00 02h ... 00 04h)	00 02 ... 00 04
Reserved	-	Enum_8	1	Always:	0	00

Table 84: Example: sWN LFPmeanfilter

CoLa A	<STX>sWN{SPC}LFPmeanfilter{SPC}1{SPC}+4{SPC}0<ETX>	
	<STX>sWN LFPmeanfilter 1 +4 0<ETX>	
	sWN LFPmeanfilter 1 +4 0	
	02 73 57 4E 20 4C 46 50 6D 65 61 6E 66 69 6C 74 65 72 20 31 20 2B 34 20 30 03	
CoLa B	02 02 02 02 00 00 00 16 73 57 4E 20 4C 46 50 6D 65 61 6E 66 69 6C 74 65 72 20 01 00 04 00 32	
	73 57 4E 20 4C 46 50 6D 65 61 6E 66 69 6C 74 65 72 20 01 00 04 00	

Table 85: Telegram structure: sWA LFPmeanfilter

Telegram structure: sWA LFPmeanfilter						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sWA	73 57 41
Command	Set average filter	String	13		LFPmeanfilter	4C 46 50 6D 65 61 6E 66 69 6C 74 65 72

Table 86: Example: sWA LFPmeanfilter

CoLa A	<STX>sWA{SPC}LFPmeanfilter<ETX>
	02 73 57 41 20 4C 46 50 6D 65 61 6E 66 69 6C 74 65 72 03
CoLa B	02 02 02 02 00 00 00 12 73 57 41 20 4C 46 50 6D 65 61 6E 66 69 6C 74 65 72 20 38

12.3.1.4.4.3 Activate Median filter [sWN LFPmedianfilter]

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

Table 87: Telegram structure: sWN LFPmedianfilter

Telegram structure: sWN LFPmedianfilter (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3		sWN	73 57 4E
Command	Sector function	String	15		LFPmedianfilter	4C 46 50 6D 65 64 69 61 6E 66 69 6C 74 65 72
Status code	Enable or disable Median filter	Bool_1	1	Disable: Enable median filter with data reduction: Enable moving median filter without data reduction:	0 1 2	00 01 02
Reserved	-	Uint_16	2	Always:	3	00 03

Enable median filter with data reduction

Table 88: Example: sWN LFPmedianfilter

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

Table 89: Telegram structure: sWA LFPmedianfilter

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

Table 90: Example: sWA LFPmedianfilter

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

12.3.1.4.4.4 Activate Edge filter [sWN LFPedgefilter]

Activate the Edge filter to eliminate wrong measurement points at object edges.

Table 91: Telegram structure: sWN LFPedgefilter

Telegram structure: sWN LFPedgefilter (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	Sector function	String	13		LFPedgefilter	4C 46 50 65 64 67 65 66 69 6C 74 65 72
Status code	Activate / deactivate edge filter	Bool_1	1	Inactive: Active:	0 1	00 01

Enable Edge filter

Table 92: Example: sWN LFPedgefilter 1

CoLa A	<STX>sWN{SPC}LFPedgefilter{SPC}1<ETX>	
	<STX>sWN LFPedgefilter 1<ETX>	
	sWN LFPedgefilter 1	
	02 73 57 4E 20 4C 46 50 65 64 67 65 66 69 6C 74 65 72 20 31 03	
CoLa B	02 02 02 02 00 00 00 13 73 57 4E 20 4C 46 50 65 64 67 65 66 69 6C 74 65 72 20 01 32	
	73 57 4E 20 4C 46 50 65 64 67 65 66 69 6C 74 65 72 20 01	

Table 93: Telegram structure: sWA LFPedgefilter

Telegram structure: sWA LFPedgefilter						
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 LFPedgefilter						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command	Sector function	String	13		LFPedgefilter	4C 46 50 65 64 67 65 66 69 6C 74 65 72

Table 94: Example: sWA LFPedgefilter

CoLa A	<STX>sWA{SPC}LFPedgefilter<ETX>
	02 73 57 41 20 4C 46 50 65 64 67 65 66 69 6C 74 65 72 03
CoLa B	02 02 02 02 00 00 00 12 73 57 41 20 4C 46 50 65 64 67 65 66 69 6C 74 65 72 20 3C

12.3.1.4.5 Inputs and Outputs

12.3.1.4.5.1 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 91.

Table 95: 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 96: 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 97: 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

Telegram structure: sRA LIDinputstate						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
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 98: Example: sRA LIDinputstate In1 active, In2 active, In3 inactive, In4 not used, In5 not used, In6 inactive, In7 inactive, In8 not used, time: 2022-04-25 14:10 46.946 sec

CoLa A	<STX>sRA{SPC}LIDinput-state{SPC}1{SPC}98E71BD7{SPC}1{SPC}1{SPC}0{SPC}2{SPC}2{SPC}0{SPC}0{SPC}2{SPC}1{SPC}7E6{SPC}4{SPC}19{SPC}E{SPC}A{SPC}2E{SPC}E6F50<ETX>
	02 73 52 41 20 4C 49 44 69 6E 70 75 74 73 74 61 74 65 20 31 20 39 38 45 37 31 42 44 37 20 31 20 31 20 30 20 32 20 32 20 30 20 30 20 32 20 31 20 37 45 36 20 34 20 31 39 20 45 20 41 20 32 45 20 45 36 46 35 30 03
CoLa B	02 02 02 02 00 00 00 2D 73 52 41 20 4C 49 44 69 6E 70 75 74 73 74 61 74 65 20 00 01 98 E7 1B D7 01 01 00 02 02 00 00 02 00 01 07 E6 04 19 0E 0A 2E 00 0E 6F 50 76

12.3.1.4.5.2 Read state of the outputs [sRN LIDoutputstate]

Status of all outputs

Table 99: 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 100: 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 101: 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 91.						

12.3.1.4.5.3 Receive outputstate by event [sEN LIDoutputstate]

Output telegram is sent every time an output state changes.

Table 102: 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 103: 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 104: 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

Telegram structure: sRA/sSN LIDoutputstate						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command	Output state	String	14		LIDoutputstate	4C 49 44 6F 75 74 70 75 74 73 74 61 74 65
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
Time Block (sensor-time from the last change of min. one of the outputs)	Year	Array	2	E. g.	1970	07 B2
	Month		1		1 ... 12	01 ... 0C
	Day		1		1 ... 31	01 ... 1F
	Hour		1		0 ... 23	00 ... 17
	Minute		1		0 ... 59	00 ... 3B
	Second		1		0 ... 59	00 ... 3B
	Microsecond		4		0 ... 999999	00 00 00 00 ... 00 0F 42 3F

Table 105: Example: sRA LIDoutputstate Out1 active Count 5, Out2 active Count 5, Out3 active Count 5, Out4 active Count 3, all other Outputs not used, no time data

CoLa A	<STX>sRA{SPC}LIDoutput-state{SPC}1{SPC}41F84EC5{SPC}1{SPC}5{SPC}1{SPC}5{SPC}1{SPC}5{SPC}1{SPC}3{SPC}2{SPC}0{SPC}0{SPC}0<ETX>
	02 73 52 41 20 4C 49 44 6F 75 74 70 75 74 73 74 61 74 65 20 31 20 41 F8 4E C5 20 31 20 35 20 31 20 35 20 31 20 35 20 31 20 33 20 32 20 30 20 30 20 30 03
CoLa B	02 02 02 02 00 00 00 5D 73 52 41 20 4C 49 44 6F 75 74 70 75 74 73 74 61 74 65 20 00 01 41 F8 4E C5 01 00 00 00 05 01 00 00 00 05 01 00 00 00 05 01 00 00 00 03 02 00 00 00 00 00 00 06

12.3.1.4.5.4 Set output state [sMN mDOSetOutput]

Set a specific output to high or low via software command.

NOTE
Output source needs to be set to "SOPAS command".

Table 106: 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		1 ... 4	01 ... 04
Output state		Enum_8	1	Active: Not active:	0 1	00 01

Table 107: 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 108: 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 109: Example: sAN mDOSetOutput

CoLa A	<STX>sAN{SPC}mDOSetOutput{SPC}1<ETX>
	02 73 41 4E 20 6D 44 4F 53 65 74 4F 75 74 70 75 74 20 31 03
CoLa B	02 02 02 02 00 00 00 12 73 41 4E 20 6D 44 4F 53 65 74 4F 75 74 70 75 74 20 01 66

12.3.1.4.5.5 Change output logic state [sWN DO1Logic]

NOTE

For Output 2,3 and 4 use the corresponding commands DO2Logic, DO3Logic and DO4Logic

Table 110: Telegram structure: sWN DO1Logic

Telegram structure: sWN DO1Logic (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	Output function	String	8	Output 1: Output 2: Output 3: Output 4:	DO1Logic DO2Logic DO3Logic DO4Logic	44 4F 31 4C 6F 67 69 63 44 4F 32 4C 6F 67 69 63 44 4F 33 4C 6F 67 69 63 44 4F 34 4C 6F 67 69 63
Output 1 logic state	State of the output	Enum_8	1	Active_High: Active_Low:	0 1	00 01

Table 111: Example: sWN DO1Logic → Active_Low

CoLa A	<STX>sWN{SPC}DO1Logic{SPC}1<ETX>	
	<STX>sWN DO1Logic 1<ETX>	
	sWN DO1Logic 1	
	02 73 57 4E 20 44 4F 31 4C 6F 67 69 63 20 31 03	
CoLa B	02 02 02 02 00 00 00 0E 73 57 4E 20 44 4F 31 4C 6F 67 69 63 20 01 1F	
	73 57 4E 20 44 4F 31 4C 6F 67 69 63 20 01	

Table 112: Telegram structure: sWA DO1Logic

Telegram structure: sWA DO1Logic						
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	Output function	String	8		DO1Logic	44 4F 31 4C 6F 67 69 63

Table 113: Example: sWA DO1Logic

CoLa A	<STX>sWA{SPC}DO1Logic<ETX>
	02 73 57 41 20 44 4F 31 4C 6F 67 69 63 03
CoLa B	02 02 02 02 00 00 00 0C 73 57 41 20 44 4F 31 4C 6F 67 69 63 31

12.3.1.4.5.6 Reset output counter [sMN LIDrstoutpcnt]

Reset the counter which keeps track of how often an digital output has been active (and not active). Information from the counter is included in `LIDoutputstate` (see "Receive outputstate by event [sEN LIDoutputstate]", page 91).

Table 114: 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 115: 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 116: 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 117: Example: sAN LIDrstoutpcnt

CoLa A	<STX>sAN{SPC}LIDrstoutpcnt{SPC}0<ETX>
	02 73 41 4E 20 4C 49 44 72 73 74 6F 75 74 70 63 6E 74 20 30 03
CoLa B	02 02 02 02 00 00 00 13 73 41 4E 20 4C 49 44 72 73 74 6F 75 74 70 63 6E 74 20 00 2F

12.3.1.4.6 Status

12.3.1.4.6.1 Read firmware version [sRN DeviceIdent]

Table 118: 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

Telegram structure: sRN DeviceIdent						
Telegram part	Description	Variable	Length	Additional details	Values CoLa A (ASCII)	Values CoLa B (Binary)
Command	Read firmware version	String	11		DeviceIdent	44 65 76 69 63 65 49 64 65 6E 74

Table 119: 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 120: 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 121: Example: sRA DeviceIdent with TiM781 V5.01-30.03.23

CoLa A	<STX>sRA{SPC}DeviceIdent{SPC}6{SPC}TiM781{SPC}E{SPC}V5.01-30.03.23<ETX>
	02 73 52 41 20 44 65 76 69 63 65 49 64 65 6E 74 20 36 20 54 69 4D 37 38 31 20 45 20 56 35 2E 30 31 2D 33 30 2E 30 33 2E 32 33 03
CoLa B	02 02 02 02 00 00 00 28 73 52 41 20 44 65 76 69 63 65 49 64 65 6E 74 20 00 06 54 69 4D 37 38 31 00 0E 56 35 2E 30 31 2D 33 30 2E 30 33 2E 32 33 2C

12.3.1.4.6.2 Read the device state [sRN SCdevicestate]

This telegram reads the general device state.

Table 122: 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 123: 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 124: Telegram structure: sRA SCdevicestate

Telegram structure: sRA SCdevicestate						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sRA	73 52 41
Command	Read the device state	String	13		SCdevicestate	53 43 64 65 76 69 63 65 73 74 61 74 65
Status code	Code number	Enum_8	1	Busy / logged-in: Ready: Error:	0 1 2	00 01 02

Table 125: Example: sRA SCdevicestate

CoLa A	<STX>sRA{SPC}SCdevicestate{SPC}1<ETX>
	02 73 52 41 20 53 43 64 65 76 69 63 65 73 74 61 74 65 20 31 03
CoLa B	02 02 02 02 00 00 00 13 73 52 41 20 53 43 64 65 76 69 63 65 73 74 61 74 65 20 01 1E

12.3.1.4.6.3 Read device order number [sRN DIornr]

This telegram reads the device order number.

Table 126: Telegram structure: sRN DIornr

Telegram structure: sRN DIornr						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E

Telegram structure: sRN DIornr						
Telegram part	Description	Variable	Length	Additional details	Values CoLa A (ASCII)	Values CoLa B (Binary)
Command	Read device order number	String	6		DIornr	44 49 6F 72 6E 72

Table 127: Example: sRN DIornr

CoLa A	<STX>sRN{SPC}DIornr<ETX>	
	<STX>sRN DIornr<ETX>	
	sRN DIornr	
	02 73 52 4E 20 44 49 6F 72 6E 72 03	
CoLa B	02 02 02 02 00 00 00 0A 73 52 4E 20 44 49 6F 72 6E 72 43	
	73 52 4E 20 44 49 6F 72 6E 72	

Table 128: Telegram structure: sRA DIornr

Telegram structure: sRA DIornr						
Telegram part	Description	Variable	Length	Additional details	Values CoLa A (ASCII)	Values CoLa B (Binary)
Command type	Answer	String	3		sRA	73 52 41
Command	Read device order number	String	6		DIornr	44 49 6F 72 6E 72
Order number	Order number in 7 digits	String	7		0000000 ... 9999999	00 00 00 00 00 00 00 ... FF FF FF FF FF FF FF

Example: sRA DIornr 1071419 (Order Number for TiM561-2050101)

Table 129: Example: sRA DIornr

CoLa A	<STX>sRA{SPC}DIornr{SPC}1071419<ETX>
CoLa B	02 02 02 02 00 00 00 12 73 52 41 20 44 49 6F 72 6E 72 20 31 30 37 31 34 31 39 57

12.3.1.4.6.4 Read device type [sRN DItype]

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

Table 130: 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 131: 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 132: Telegram structure: sRA DItype

Telegram structure: sRA DItype						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sRA	73 52 41
Command	Ask state	String	6		DItype	44 49 74 79 70 65
Length of type key	Number of digits of the following type code length	Uint_8	1		0d ... 255d (0h ... FFh)	00 ... FF
Device type	Type code of the device	String	(var.)		(Device type)	(Device type)

Table 133: sRA DItype Example for TiM561-2050101:

CoLa A	<STX>sRA{SPC}DItype{SPC}E{SPC}TIM561-2050101<ETX>
	02 73 52 41 20 44 49 74 79 70 65 20 45 20 54 49 4D 35 36 31 2D 32 30 35 30 31 30 31 03
CoLa B	02 02 02 02 00 00 00 1A 73 52 41 20 44 49 74 79 70 65 20 0E 54 49 4D 35 36 31 2D 32 30 35 30 31 30 31 03

12.3.1.4.6.5 Read operating hours [sRN ODoprh]

Table 134: 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 135: 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 136: 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 137: Example: sRA ODoprh

CoLa A	<STX>sRA{SPC}ODoprh{SPC}2DC8B<ETX>
	02 73 52 41 20 4F 44 6F 70 72 68 20 32 44 43 38 42 03
CoLa B	02 02 02 02 00 00 00 0F 73 52 41 20 4F 44 6F 70 72 68 20 00 02 DC 8B 36

Calculation of the value: 2DC8B (hex) → 187531 (dez) × 1/10 h = 18753.1 h

12.3.1.4.6.6 Read power on counter [sRN ODpwrc]

Shows the number of power on cycles.

Table 138: 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 139: 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 140: 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 141: Example: sRA ODpwrc

CoLa A	<STX>sRA{SPC}ODpwrc{SPC}752D<ETX>
	02 73 52 41 20 4F 44 70 77 72 63 20 752D 03
CoLa B	02 02 02 02 00 00 00 0F 73 52 41 20 4F 44 70 77 72 63 20 00 00 75 2D 36

12.3.1.4.6.7 Read temperature [sRN OPcurtmpdev]

With this command the internal temperature of the device can be identified. Please note that it does not give an indication of the current ambient temperature.

Table 142: 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 143: 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 144: 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)

Table 145: Example: sRA OPcurtmpdev

CoLa A	<STX>sRA{SPC}OPcurtmpdev{SPC}420C0000<ETX>
	02 73 52 41 20 4F 50 63 75 72 74 6D 70 64 65 76 20 34 32 30 43 30 30 30 30 03
CoLa B	02 02 02 02 00 00 00 14 73 52 41 20 4F 50 63 75 72 74 6D 70 64 65 76 20 42 0C 00 00 4B

12.3.1.4.6.8 Set device name [sWN LocationName]

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

Table 146: 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 147: Example: sWN LocationName +13 OutdoorDevice

CoLa A	<STX>sWN{SPC}LocationName{SPC}+13{SPC}OutdoorDevice<ETX>	
	<STX>sWN LocationName +13 OutdoorDevice<ETX>	
	sWN LocationName +13 OutdoorDevice	
	02 73 57 4E 20 4C 6F 63 61 74 69 6F 6E 4E 61 6D 65 20 2B 31 33 20 4F 75 74 64 6F 6F 72 44 65 76 69 63 65 03	
CoLa B	02 02 02 02 00 00 00 20 73 57 4E 20 4C 6F 63 61 74 69 6F 6E 4E 61 6D 65 20 00 0D 4F 75 74 64 6F 6F 72 44 65 76 69 63 65 1D	
	73 57 4E 20 4C 6F 63 61 74 69 6F 6E 4E 61 6D 65 20 00 0D 4F 75 74 64 6F 6F 72 44 65 76 69 63 65	

Table 148: 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 149: Example: sWA LocationName

CoLa A	<STX>sWA{SPC}LocationName<ETX>
	02 73 57 41 20 4C 6F 63 61 74 69 6F 6E 4E 61 6D 65 03
CoLa B	02 02 02 02 00 00 00 11 73 57 41 20 4C 6F 63 61 74 69 6F 6E 4E 61 6D 65 20 7F

12.3.1.4.6.9 Read device name [sRN LocationName]

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

Table 150: 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 151: 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 152: 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 146, page 102						

12.3.1.4.7 Interfaces

12.3.1.4.7.1 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 153: Telegram structure: sWN EIIpAddr

Telegram structure: sWN EIIpAddr (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3		sWN	73 57 4E
Command	Set IP address	String	8		EIIpAddr	45 49 49 50 41 64 64 72

Telegram structure: sWN EIIpAddr (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
IP address	Set values	Uint_8	1	First part of IP adress	0 ...+255d (00 ... FF)	00 ... FF
				Second part of IP adress	0 ...+255d (00 ... FF)	00 ... FF
				Third part of IP adress	0 ...+255d (00 ... FF)	00 ... FF
				Fourth part of IP adress	0 ...+255d (00 ... FF)	00 ... FF

Table 154: 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 155: 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 156: Example: sWA EIIpAddr

CoLa A	<STX>sWA{SPC}EIIpAddr<ETX>
	02 73 57 41 20 45 49 49 70 41 64 64 72 03
CoLa B	02 02 02 02 00 00 00 0D 73 57 41 20 45 49 49 70 41 64 64 72 20 63

12.3.1.4.7.2 Read IP address [sRN EIIpAddr]

Read the IP address of the device.

Table 157: 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 158: 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 159: 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 160: Example: sRA EIIpAddr 192.168.0.2

CoLa A	<STX>sRA{SPC}EIIpAddr{SPC}C0{SPC}A8{SPC}00{SPC}02<ETX>
	02 73 57 41 20 45 49 49 70 41 64 64 72 20 C0 20 A8 20 00 20 02 03
CoLa B	02 02 02 02 00 00 00 11 73 52 41 20 45 49 49 70 41 64 64 72 20 C0 A8 00 02 0C

12.3.1.4.7.3 Set Ethernet gateway [sWN EIgate]

Change Ethernet gateway IP address (TCP/IP)

NOTE

- Save permanently to set values. Changes will be active after rebooting the device.
- Settings must correspond with network in which scanner is used. Else device cannot be found any more.

Table 161: 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

Telegram structure: sWN Elgate (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
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 162: Example: sWN Elgate 192.168.0.1

CoLa A	<STX>sWN{SPC}Elgate{SPC}C0{SPC}A8{SPC}00{SPC}01<ETX>
	<STX>sWN Elgate C0 A8 00 01<ETX>
	sWN Elgate C0 A8 00 01
	02 73 57 4E 20 45 49 67 61 74 65 20 43 30 20 41 38 20 30 30 20 30 31 03
CoLa B	02 02 02 02 00 00 00 0F 73 57 4E 20 45 49 67 61 74 65 20 C0 A8 00 01 18
	73 57 4E 20 45 49 67 61 74 65 20 C0 A8 00 01

Table 163: 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 164: Example: sWA Elgate

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

12.3.1.4.7.4 Read Ethernet gateway [sRN Elgate]

Read for the Ethernet gateway (TCP/IP)

Table 165: 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 166: Example: sRN EIgate

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

Table 167: Telegram structure: sRA EIgate

Telegram structure: sRA EIgate						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sRA	73 52 41
Command	Read gateway address	String	6		EIgate	45 49 67 61 74 65
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 168: 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

12.3.1.4.7.5 Set IP mask [sWN EImask]

Define the subnet mask of the device.

NOTE

- Save permanently to set values. Changes will be active after rebooting the device.
- Settings must correspond with network in which scanner is used. Else device cannot be found any more.

Table 169: 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

Telegram structure: sWN EImask (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
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 170: 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 171: 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 172: Example: sWA EImask

CoLa A	<STX>sWA{SPC}EImask<ETX>
	02 73 57 41 20 45 49 6D 61 73 6B 03
CoLa B	02 02 02 02 00 00 00 0B 73 57 41 20 45 49 6D 61 73 6B 20 7D

12.3.1.4.7.6 Read IP mask [sRN EImask]

Read the subnet mask of the device.

Table 173: 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 174: 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 175: Telegram structure: sRA EImask

Telegram structure: sRA EImask						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sRA	73 52 41
Command	Read IP mask	String	6		EImask	45 49 6D 61 73 6B
IP mask	Default: 255.255.255.0	Uint_8	1	First part of IP mask	0 ...+255d (00...FF)	00 ... FF
				Second part of IP mask	0 ...+255d (00...FF)	00 ... FF
				Third part of IP mask	0 ...+255d (00...FF)	00 ... FF
				Fourth part of IP mask	0 ...+255d (00...FF)	00 ... FF

Table 176: Example: sRA EImask 255.255.254.0

CoLa A	<STX>sRA{SPC}EImask{SPC}FF{SPC}FF{SPC}FE{SPC}00<ETX>
	02 73 52 41 20 45 49 6D 61 73 6B 20 45 49 6D 61 73 6B 03
CoLa B	02 02 02 02 00 00 00 0F 73 52 41 20 45 49 6D 61 73 6B 20 FF FF FE 00 86

12.3.1.4.7.7 Set Host port number [sWN EIHstPort]

Define the TCP/IP host port number.

Table 177: Telegram structure: sWN EIHstPort

Telegram structure: sWN EIHstPort (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3		sWN	73 57 4E
Command	Read hardware settings of host interface	String	7	Ethernet host TCP/ IP	EIHstPort	45 49 48 73 74 50 6F 72 74
Port number	Host port number (Default = 2112)	Uint_16	2		+0d ... +65535d (0h ... FF FFh)	00 00 ... FF FF

Example: sWN EIHstPort +2110 (Host Port 2110)

Table 178: Example: sWN EIHstPort +2110

CoLa A	<STX>sWN{SPC}EIHstPort{SPC}+2110<ETX>
	<STX>sWN EIHstPort +2110<ETX>
	sWN EIHstPort +2110
	02 73 57 4E 20 45 49 48 73 74 50 6F 72 74 20 2B 32 31 31 30 03
CoLa B	02 02 02 02 00 00 00 10 73 57 4E 20 45 49 48 73 74 50 6F 72 74 20 08 3E 26
	73 57 4E 20 45 49 48 73 74 50 6F 72 74 20 08 3E

Table 179: Telegram structure: sWA EIHstPort

Telegram structure: sWA EIHstPort						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sWA	73 57 41
Command	Read hardware settings of host interface	String	7	Ethernet host TCP/ IP	EIHstPort	45 49 48 73 74 50 6F 72 74

Example: sWA EIHstPort (Host Port)

Table 180: Example: sWA EIHstPort

CoLa A	<STX>sWA{SPC}EIHstPort<ETX>
	02 73 57 41 20 45 49 48 73 74 50 6F 72 74 03
CoLa B	02 02 02 02 00 00 00 0E 73 57 41 20 45 49 48 73 74 50 6F 72 74 20 1F

Example: sWA EIUDPPort (UDP Port)

Table 181: Example: sWA EIUDPPort

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

12.3.1.4.7.8 Set Host port Command Language (CoLa dialect) [sWN EIHstCola]

Define the communcation language (SICK specific) of the TCP/IP host port.

NOTE

It is not allowed to use this telegram in a faster cycle than 10 ms!

After switching the CoLa dialect by this telegram, you have to store the changes permanentely and reboot the sensor to activate the chosen CoLA dialect.

Table 182: Telegram structure: sWN EIHstCola

Telegram structure: sWN EIHstCola (User level 'Authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3		sWN	73 57 4E
Command	Set hardware settings of host interface	String	7		EIHstCola	45 49 48 73 74 43 6F 6C 61
Command language	Host port Command language	Enum_8	1	CoLa ASCII: CoLa binary:	0 1	00 01

Example: sWN EIHstCola 1 (Host Port CoLa binary)

Table 183: Example: sWN EIHstCola 1

CoLa A	<STX>sWN{SPC}EIHstCola{SPC}1<ETX>	
	<STX>sWN EIHstCola 1<ETX>	
	sWN EIHstCola 1	
	02 73 57 4E 20 45 49 48 73 74 43 6F 6C 61 20 31 03	
CoLa B	02 02 02 02 00 00 00 0F 73 57 4E 20 45 49 48 73 74 43 6F 6C 61 20 01 09	
	73 57 4E 20 45 49 48 73 74 43 6F 6C 61 20 01	

Table 184: Telegram structure: sWA EIHstCola

Telegram structure: sWA EIHstCola						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sWA	73 57 41
Command	Set hardware settings of host interface	String	7		EIHstCola	45 49 48 73 74 43 6F 6C 61

Example: sWA EIHstCola (Host Port)

Table 185: Example: sWA EIHstCola

CoLa A	<STX>sWA{SPC}EIHstCola<ETX>
	02 73 57 41 20 45 49 48 73 74 43 6F 6C 61 03
CoLa B	02 02 02 02 00 00 00 0E 73 57 41 20 45 49 48 73 74 43 6F 6C 61 20 07

12.3.1.4.8 Application

12.3.1.4.8.1 Set Fieldset selection method [sWN FieldSetSelectionMethod]

Define if the device will switch from one to another defined fieldset via a pre-defined status change of the digital inputs or via a telegram command.

Table 186: Telegram structure: sWN FieldSetSelectionMethod

Telegram structure: sWN FieldSetSelectionMethod (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	Fieldset selection method	String	23		FieldSetSelectionMethod	46 69 65 6C 64 53 65 74 53 65 6C 65 63 74 69 6F 6E 4D 65 74 68 6F 64
Method selection	Set fieldset selection method	Enum_8	1	Digital inputs: Telegram:	0 1	00 01

Table 187: Example for TiM781: Example: sWN FieldSetSelectionMethod

CoLa A	<STX>sWN{SPC}FieldSetSelectionMethod{SPC}1<ETX>
	<STX>sWN FieldSetSelectionMethod 1<ETX>
	sWN FieldSetSelectionMethod 1
	02 73 57 4E 20 46 69 65 6C 64 53 65 74 53 65 6C 65 63 74 69 6F 6E 4D 65 74 68 6F 64 20 31 03
CoLa B	02 02 02 02 00 00 00 13 73 57 4E 20 46 69 65 6C 64 53 65 74 53 65 6C 65 63 74 69 6F 6E 4D 65 74 68 6F 64 20 01 14
	73 57 4E 20 46 69 65 6C 64 53 65 74 53 65 6C 65 63 74 69 6F 6E 4D 65 74 68 6F 64 20 01

Table 188: Telegram structure: sWA FieldSetSelectionMethod

Telegram structure: sWA FieldSetSelectionMethod						
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	Fieldset selection Method	String	23		FieldSetSelectionMethod	46 69 65 6C 64 53 65 74 53 65 6C 65 63 74 69 6F 6E 4D 65 74 68 6F 64

Table 189: Example: sWA FieldSetSelectionMethod

CoLa A	<STX>sWA{SPC}FieldSetSelectionMethod<ETX>
	02 73 57 41 20 46 69 65 6C 64 53 65 74 53 65 6C 65 63 74 69 6F 6E 4D 65 74 68 6F 64 03
CoLa B	02 02 02 02 00 00 00 15 73 52 41 20 46 69 65 6C 64 53 65 74 53 65 6C 65 63 74 69 6F 6E 4D 65 74 68 6F 64 20 1A

12.3.1.4.8.2 Read Fieldset selection method [sRN FieldSetSelectionMethod]

Read if the device will switch from one to another defined fieldset via a pre-defined status change of the digital inputs or via a telegram command.

Table 190: Telegram structure: sRN FieldSetSelectionMethod

Telegram structure: sRN FieldSetSelectionMethod						
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	Fieldset selection Method	String	23		FieldSetSelectionMethod	46 69 65 6C 64 53 65 74 53 65 6C 65 63 74 69 6F 6E 4D 65 74 68 6F 64

Table 191: Example: sRN FieldSetSelectionMethod

CoLa A	<STX>sRN{SPC}FieldSetSelectionMethod<ETX>
	<STX>sRN FieldSetSelectionMethod<ETX>
	sRN FieldSetSelectionMethod
	02 73 52 4E 20 46 69 65 6C 64 53 65 74 53 65 6C 65 63 74 69 6F 6E 4D 65 74 68 6F 64 03

CoLa B	02 02 02 02 00 00 00 1B 73 52 4E 20 46 69 65 6C 64 53 65 74 53 65 6C 65 63 74 69 6F 6E 4D 65 74 68 6F 64 64
	73 52 4E 20 46 69 65 6C 64 53 65 74 53 65 6C 65 63 74 69 6F 6E 4D 65 74 68 6F 64

Table 192: Telegram structure: sRA FieldSetSelectionMethod

Telegram structure: sRA FieldSetSelectionMethod						
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	Fieldset selection Method	String	23		FieldSetSelec-tionMethod	46 69 65 6C 64 53 65 74 53 65 6C 65 63 74 69 6F 6E 4D 65 74 68 6F 64
Method selection	Read fieldset selection method	Enum_8	1	Digital inputs: Telegram:	0 1	00 01

Table 193: Example: sRA FieldSetSelectionMethod

CoLa A	<STX>sRA{SPC}FieldSetSelectionMethod{SPC}1<ETX>
	02 73 52 41 20 46 69 65 6C 64 53 65 74 53 65 6C 65 63 74 69 6F 6E 4D 65 74 68 6F 64 20 31 03
CoLa B	02 02 02 02 00 00 00 1D 73 52 41 20 46 69 65 6C 64 53 65 74 53 65 6C 65 63 74 69 6F 6E 4D 65 74 68 6F 64 20 01 1E

12.3.1.4.8.3 Set active fieldset [sWN ActiveFieldSet]

Define via telegram which predefined fieldset should be set active.

Table 194: Telegram structure: sWN ActiveFieldSet

Telegram structure: sWN ActiveFieldSet						
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	Active fieldset	String	14		ActiveFieldSet	41 63 74 69 76 65 46 69 65 6C 64 53 65 74
Active Fieldset		Uint_16	2		0h ... 10h	00 01 ... 00 10

Table 195: Example: sWN ActiveFieldSet

CoLa A	<STX>sWN{SPC}ActiveFieldSet{SPC}E<ETX>
	<STX>sWN ActiveFieldSet E<ETX>
	sWN ActiveFieldSet E
	02 73 57 4E 20 41 63 74 69 76 65 46 69 65 6C 64 53 65 74 20 45 03
CoLa B	02 02 02 02 00 00 00 13 73 57 4E 20 41 63 74 69 76 65 46 69 65 6C 64 53 65 74 20 00 0E 63
	73 57 4E 20 41 63 74 69 76 65 46 69 65 6C 64 53 65 74 20 00 0E

Table 196: Telegram structure: sWA ActiveFieldSet

Telegram structure: sWA ActiveFieldSet						
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	Active fieldset	String	14		ActiveFieldSet	41 63 74 69 76 65 46 69 65 6C 64 53 65 74

Table 197: Example: sWA ActiveFieldSet

CoLa A	<STX>sWA{SPC}ActiveFieldSet<ETX>
	02 73 57 41 20 41 63 74 69 76 65 46 69 65 6C 64 53 65 74 03
CoLa B	02 02 02 02 00 00 00 13 73 57 41 20 41 63 74 69 76 65 46 69 65 6C 64 53 65 74 20 49

12.3.1.4.8.4 Read active fieldset [sRN ActiveFieldSet]

Read which predefined fieldset is currently active.

Table 198: Telegram structure: sRN ActiveFieldSet

Telegram structure: sRN ActiveFieldSet						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3		sRN	73 52 4E
Command	Info of active fieldset	String	14		ActiveFieldSet	41 63 74 69 76 65 46 69 65 6C 64 53 65 74

Table 199: Example: sRN ActiveFieldSet

CoLa A	<STX>sRN{SPC}ActiveFieldSet<ETX>	
	<STX>sRN ActiveFieldSet<ETX>	
	sRN ActiveFieldSet	
	02 73 52 4E 20 41 63 74 69 76 65 46 69 65 6C 64 53 65 74 03	
CoLa B	02 02 02 02 00 00 00 13 73 52 4E 20 41 63 74 69 76 65 46 69 65 6C 64 53 65 74 20 63	
	73 52 4E 20 41 63 74 69 76 65 46 69 65 6C 64 53 65 74 20	

Table 200: Telegram structure: sRA ActiveFieldSet

Telegram structure: sRA ActiveFieldSet						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3		sRA	73 52 41
Command	Info of active fieldset	String	14		ActiveFieldSet	41 63 74 69 76 65 46 69 65 6C 64 53 65 74

Telegram structure: sRA ActiveFieldSet						
Telegram part	**Description**	**Variable**	**Length**	**Additional details**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Active Fieldset		Uint_16	2		1h ...10h	00 01 ... 00 10

Table 201: Example: sRA ActiveFieldSet

CoLa A	<STX>sRA{SPC}ActiveFieldSet{SPC}F<ETX>
	02 73 52 41 20 41 63 74 69 76 65 46 69 65 6C 64 53 65 74 20 46 03
CoLa B	02 02 02 02 00 00 00 15 73 52 41 20 41 63 74 69 76 65 46 69 65 6C 64 53 65 74 20 00 0F 43

12.3.1.5 Diagnostics

12.3.1.5.1 SOPAS error codes

sFA ErrorCode

Table 202: SOPAS error codes

Telegram structure: sFA ErrorCode			
Error code	**Description**	**Dec.**	**Hex.**
Sopas_Ok	No error	0	0
Sopas_Error_METHODIN_ACCESSDENIED	Wrong userlevel, access to method not allowed	1	1
Sopas_Error_METHODIN_UNKNOWNINDEX	Trying to access a method with an unknown Sopas index	2	2
Sopas_Error_VARIABLE_UNKNOWNINDEX	Trying to access a variable with an unknown Sopas index	3	3
Sopas_Error_LOCALCONDITIONFAILED	Local condition violated, e.g. giving a value that exceeds the minimum or maximum allowed value for this variable	4	4
Sopas_Error_INVALID_DATA	Invalid data given for variable, this errorcode is deprecated (is not used anymore).	5	5
Sopas_Error_UNKNOWN_ERROR	An error with unknown reason occurred, this errorcode is deprecated.	6	6
Sopas_Error_BUFFER_OVERFLOW	The communication buffer was too small for the amount of data that should be serialised.	7	7
Sopas_Error_BUFFER_UNDERFLOW	More data was expected, the allocated buffer could not be filled.	8	8
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

Telegram structure: sFA ErrorCode			
Error code	**Description**	**Dec.**	**Hex.**
Sopas_Error_FLEX_OUT_OF_BOUNDS	An array was accessed over its maximum length.	14	E
Sopas_Error_EVENTREG_UNKNOWNINDEX	The event you wanted to register for does not exist, the index is unknown.	15	F
Sopas_Error_COLA_A_VALUE_OVERFLOW	The value does not fit into the value field, it is too large.	16	10
Sopas_Error_COLA_A_INVALID_CHARACTER	Character is unknown, probably not alphanumeric.	17	11
Sopas_Error_OSAI_NO_MESSAGE	Only when using SRTOS in the firmware and distributed variables this error can occur. It is an indication that no operating system message could be created. This happens when trying to GET a variable.	18	12
Sopas_Error_OSAI_NO_ANSWER_MESSAGE	This is the same as `Sopas_Error_OSAI_NO_MESSAGE` with the difference that it is thrown when trying to PUT a variable.	19	13
Sopas_Error_INTERNAL	Internal error in the firmware, problably a pointer to a parameter was null.	20	14
Sopas_Error_HubAddressCorrupted	The Sopas Hubaddress is either too short or too long.	21	15
Sopas_Error_HubAddressDecoding	The Sopas Hubaddress is invalid, it can not be decoded (Syntax).	22	16
Sopas_Error_HubAddressAddressExceeded	Too many hubs in the address	23	17
Sopas_Error_HubAddressBlankExpected	When parsing a HubAddress an expected blank was not found. The HubAddress is not valid.	24	18
Sopas_Error_AsyncMethodsAreSuppressed	An asynchronous method call was made although the device was built with “AsyncMethodsSuppressed”. This is an internal error that should never happen in a released device.	25	19
Sopas_Error_ComplexArraysNotSupported	Device was built with „ComplexArraysSuppressed“ because the compiler does not allow recursions. But now a complex array was found. This is an internal error that should never happen in a released device.	26	20

Table 203: 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

8024231/1ON0/2024-09-25/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.