TECHNICAL INFORMATION

Telegram Listing

Ranging sensors LMS1xx, LMS5xx, TiM5xx, TiM7xx, LMS1000, MRS1000, MRS6000, NAV310, LD-OEM15xx, LD-LRS36xx, LMS4000

SICK
Sensor Intelligence.

Described product

LMS1xx, LMS5xx, TiM5xx, TiM7xx, LMS1000, MRS1000, MRS6000, NAV310, LD-OEM15xx, LD-LRS36xx, LMS4000

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 6

2 Communication format 7

2.1 Binary telegram (CoLa B) 7
2.2 ASCII telegram (CoLa A) 8
2.3 Variable types 9
2.4 Command basics 9
2.5 Log in: Required user level 10

3 Workflows 11

3.1 Parameterize the scan 11
3.2 Set timestamp/data angle 11

4 Telegrams 12

4.1 Log in 12
4.2 Basic Settings 14
4.2.1 Set frequency and angular resolution/measurement sectors 14
4.2.2 Read for frequency and angular resolution 24
4.2.3 Alignment mode (one layer activation for adjustment) 27
4.2.4 Set scan configuration 28
4.2.5 Activate standby mode 33
4.2.6 Start measurement 34
4.2.7 Stop measurement 35
4.2.8 Autostart measurement 37
4.2.9 Laser Control 38
4.2.10 Activate/deactivate field application 40
4.2.11 Application selection and switching 41
4.2.12 Read Application selection and switching 42
4.2.13 Load factory defaults 43
4.2.14 Load application defaults 44
4.2.15 Change password 46
4.2.16 Check password 48
4.2.17 Reboot device 49
4.2.18 Set contamination measurement settings 51
4.2.19 Read contamination measurement settings 52
4.2.20 Read contamination measurement detailed values 53
4.2.21 Save parameters permanently 55
4.2.22 Set to run 57
4.3 Measurement output telegram 58
4.3.1 Configure the data content for the scan 58
4.3.2 Configure measurement angle of the scandata for output 62

4.3.3 Read for actual output range ... 64
4.3.4 Poll one telegram ... 67
4.3.5 Send data permanently ... 69
4.4 Time stamp ... 88
4.4.1 Set time stamp ... 88
4.4.2 Read time stamp and status of the measurement function ... 90
4.4.3 Read device time ... 92
4.4.4 Set NTP (Network Time Protocol) parameters ... 93
4.5 Filter ... 103
4.5.1 Set particle filter ... 103
4.5.2 Set mean filter ... 104
4.5.3 Set n-pulse to 1-pulse filter (Echo filter) ... 105
4.5.4 Set echo filter ... 106
4.5.5 Set and read fog filter ... 108
4.5.6 Enable/disable digital nearfield filter ... 113
4.5.7 Set digital nearfield filter sector selection ... 114
4.5.8 Set Median Filter ... 115
4.5.9 Set Edge Filter ... 116
4.6 Encoder ... 118
4.6.1 Set increment source ... 118
4.6.2 Set encoder settings ... 119
4.6.3 Set encoder resolution ... 120
4.6.4 Set fixed speed ... 122
4.6.5 Read speed threshold ... 123
4.6.6 Read encoder speed ... 124
4.7 Inputs and Outputs ... 126
4.7.1 Port Configration of all I/Os ... 126
4.7.2 Read state of the outputs ... 128
4.7.3 Send outputstate by event ... 129
4.7.4 Set output state ... 132
4.7.5 Change output 6/3 function ... 134
4.7.6 Change output 1 function ... 135
4.7.7 Change output 1 logic state ... 137
4.7.8 Change output 2 function ... 138
4.7.9 Change output 2 logic state ... 139
4.7.10 Set synchronization mode ... 140
4.7.11 Set synchronization phase ... 141
4.7.12 Change input 4 function ... 142
4.7.13 Set debouncing time for input x ... 143
4.7.14 Read status of external sync signal ... 144
4.8 Status ... 146
4.8.1 Read contamination status of the LMS ... 146
4.8.2 Read firmware version ... 147

4.8.3 Read the device state ... 149
4.8.4 Status commands for LD-XXX and NAV310 ... 151
4.8.5 Read device order number ... 157
4.8.6 Read device type ... 158
4.8.7 Read operating hours ... 160
4.8.8 Read power on counter ... 161
4.8.9 Read temperature ... 162
4.8.10 Set device name ... 164
4.8.11 Read device name ... 165
4.8.12 Read angle compensation sine ... 167
4.8.13 Reset output counter ... 168
4.8.14 Read heating state ... 170
4.9 Interfaces ... 172
4.9.1 Set IP address ... 172
4.9.2 Read IP address ... 174
4.9.3 Set Ethernet gateway ... 175
4.9.4 Read Ethernet gateway ... 176
4.9.5 Set IP mask ... 177
4.9.6 Read IP mask ... 179
4.9.7 Set baud rate for host interface ... 181
4.9.8 Read baud rate of host interface ... 182
4.9.9 Set interface type ... 184
4.9.10 Read interface type ... 185
4.9.11 Set function front panel ... 187
4.9.12 Set front LEDs ... 188
4.9.13 Set function of LED1 ... 190
4.9.14 Set function of LED2 ... 191
4.9.15 Switch on/off LED1 or LED2 ... 192
4.9.16 Set 7-segment display to specific symbol or number ... 194
4.9.17 Switching CoLa dialect on ethernet host port ... 197
4.10 Application ... 199
4.10.1 Request status change of monitoring fields on event ... 199
4.10.2 Request SOPAS field data structure ... 203
4.10.3 Request perpendicular distance once ... 208
4.10.4 Request perpendicular distance continiously on event ... 214
4.10.5 Request latest field infringement info ... 219
4.10.6 Request field infringement info continiously on event ... 223

5 Diagnostics ... 228
5.1 SOPAS error codes ... 228
5.2 Additional information ... 230

6 List of tables ... 231

1 About this document

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

The document shows how to send telegrams via a terminal program using the SICK protocol CoLa A (ASCII and hexadecimal values, with TCP port 2111 or 2112) or CoLa B (binary/hexadecimal values, with TCP port 2112 only) to the laserscanners LMS1xx, LMS5xx, TiM5xx (TiM55x, TiM56x, TiM57x), TiM7xx, LMS1000, MRS1000, MRS6000, NAV310, LD-OEM15xx, LD-LRS36xx and LMS4000. 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:

- LMSxxx: 192.168.0.1
- TiMxxx: 192.168.0.1
- MRSxxxx: 192.168.0.1
- NAV310: 192.168.1.10
- LD-XXXxxxx: 192.168.1.10
- LMS4000: 192.168.0.1

Subnet mask is 255.255.255.0.

IP ports:

- 2111: CoLa A (fixed) (for LMS4000 fixed CoLa B)
- 2112: CoLa A (can be switched to CoLa B)
- 2213: UDP

The document 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 telegram listing is based on the following firmware statuses (or newer):

- LMS1xx: V1.80 (V1.21 for LMS12x/13x)
- LMS5xx: V1.50.6 (V31.39 for LMS531)
- TiMxxx: V2.51
- MRS1000: 1.0.0 (1.0.0.0R)
- LMS1000: 1.4.0 (1.4.0.0R)
- NAV310: V1.03
- LD-OEM15xx: V1.12 (V1.32 for OEM1500)
- LD-LRS36xx: V1.12 (V1.32 for LRS3600)
- LMS4000: V1.4

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.

2 Communication format

2.1 Binary telegram (CoLa B)

The binary telegram is the 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).

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.

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 blank (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.

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

Table 1: Example: Binary telegram

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

- Header = 02 02 02 02
- Length = 23 digit pairs (17h)
- Data:
  - 73 4D 4E 20 = sMN = start of Sopas command (and blank)
  - 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 blank)
  - 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

2.2 ASCII telegram (CoLa A)

The ASCII telegram is an alternative to the binary telegram. Due to the variable string length of ASCII telegrams, the Binary telegram is 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. As leading zeros are being deleted, there is always a blank required between all command parts and parameter parts.

NOTE

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

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.

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

Table 2: Example: ASCII telegram

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)

2.3 Variable types

Variable type	Length (byte)	Value range	Sign
Bool_1	1	0 or 1	No
Uint_8	1	0 ... 255	No
Int_8	1	-128 ... +127	Yes
Uint_16	2	0 ... 65,535	No
Int_16	2	-32,768 ... +32,767	Yes
Uint_32	4	0 ... 4,294,967,295	No
Int_32	4	-2,147,483,648 ... +2,147,483,647	Yes
Enum_8	1	Certain values defined in a list of Choices (0 ... 255)	No
Enum_16	2	Certain values defined in a list of Choices (0 ... 65535)	No
String	Context-dependent	Strings are not terminated in zeroes	
Real		Float nach IEEE754 (see www.h-schmidt.net/FloatConverter/IEEE754de.html)	

Data length is always given in Bytes!

2.4 Command basics

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

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 password	Service

3 Workflows

3.1 Parameterize the scan

1 Log in: sMN SetAccessMode (see 4.1, page 12)
2 Set frequency and resolution: sMN mLMPsetscancfg (see 4.2.1, page 14)
3 Configure scandata content: sWN LMDscandatacfg (see 4.3.1, page 58)
4 Configure scandata output: sWN LMPoutputRange (see 4.3.2 page 62)
5 Store parameters: sMN mEEwriteall (see 4.2.21, page 55)
6 Log out: sMN Run (see 4.2.22, page 57)
7 Request scan:
sRN LMDscandata (see 4.3.4, page 67)
sEN LMDscandata (see 4.3.5, page 69)

(Device output ...)

More detailed command descriptions can be found in the course of this document.

Example: Sequence for LD-OEM1501, NAV310, LD-LR3601, LD-LR3611 to configure 2 sectors and get measurement scans

Sector configuration: Resolution: 10Hz; 0,125°;
Sector 1: 0° ... 44° (0h ... 6B6C0h);
Sector 2: 45° ... 180° (6DDD6h ... 1B7740h)

1 Stop measurement: sMN LMCstopmeas
sAN LMCstopmeas 0
2 Log in: sMN SetAccessMode (see 4.1, page 12)
3 Set Sectors : LCMstate001B7740 04E2 000000 000000 04E2 000000 000000
sAN mLMPsetscancfg 0 3E8 2 4E20 6B6C0 4E2 6DDD6 1B7740 4E2 00 4E2 00
4 Store parameters: sMN mEEwriteall (see 4.2.21, page 55)
5 Log out: sMN Run (see 4.2.22, page 57)
6 Start Measurement: sMN LMCstartmeas
sAN LMCstartmeas 0
7 Request scan:
sRN LMDscandata (see 4.3.4, page 67)
sEN LMDscandata (see 4.3.5, page 69)

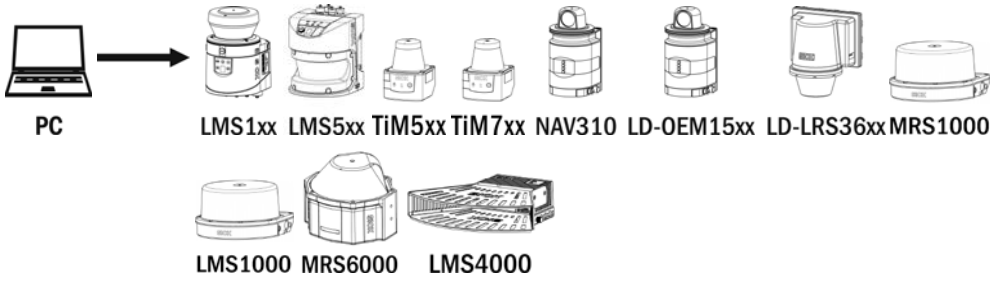
(Device output ...)

3.2 Set timestamp/data angle

1 Log in: sMN SetAccessMode (see 4.1, page 12)
2 Sopas command: sMN LSPsetdatetime (see 4.4.1, page 88)
3 Log out: sMN Run (see 4.2.22, page 57)

4 Telegrams

4.1 Log in



NOTES

- Please note that for TiMxxx and LMS4000 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 (sMN Run).

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

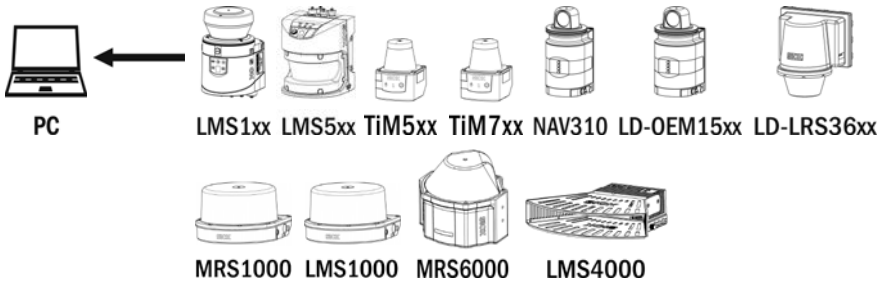
Table 3: Telegram structure: sMN SetAccessMode

Example: sMN SetAccessMode

Log in as “Authorized client” with password “F4724744”.

CoLa A	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
CoLa B	Binary	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

Table 4: Example: sMN SetAccessMode



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

Table 5: Telegram structure: sAN SetAccessMode

Example for LMS100: sAN SetAccessMode

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

Table 6: Example for LMS100: sAN SetAccessMode

4.2 Basic Settings

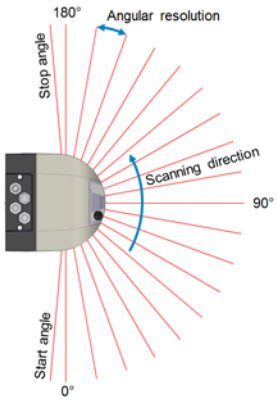
4.2.1 Set frequency and angular resolution/measurement sectors

NOTES

- Please note that the new values will be activated only after log out (from the user level), when re-entering the Run mode (see Table 102 on page 57).

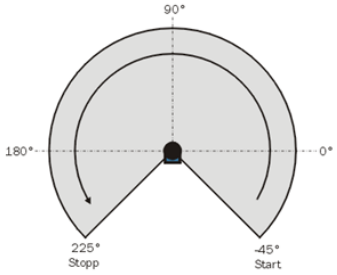
Coordination system of:

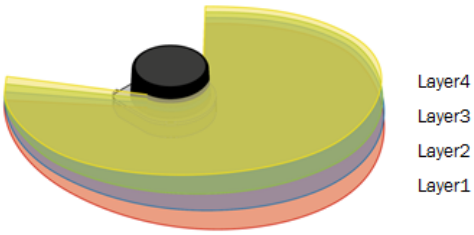
LMS5xx (-5 ° to 185°) → Start angle and stop angle are fixed values and not changeable for LMS5xx, only in the data output! This also applies for LMS1xx series.



LMS1xx and TiMxxx (-45° to 225°)

MRS1000 (-47,5° to 227,5 °) and LMS1000 (-48° to 228°)





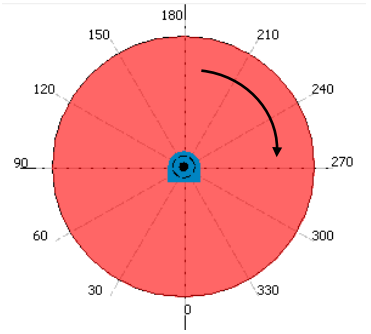
Sequence of the Layers In the Telegram

(Output sequence (DIN70000): 0, -250, 250, -500)

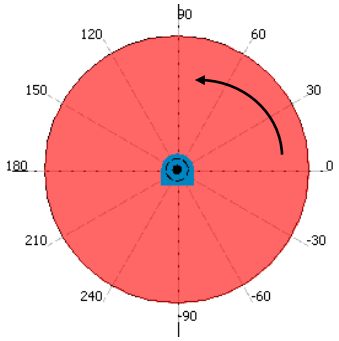
0	→ 0	Layer2
FF06	→ -250	Layer3
FA	→ 250	Layer1
FE0C	→ -500	Layer4

The LD series is available in two versions having a different rotation direction and coordinate system:

LD-OEM1501, NAV310, LD-LR3601, LD-LR3611 (0° to 360°)



LD-OEM1500 and LD-LR3600 (–90° to +270°)



For sending the sector configuration there follow these rules:

- Send the sectors in their ascending sequence.
- For LD and NAV products: Send allways the definition for all sectors (unused sector as "{SPC}0{SPC}0".)
- For LMS products: They have only one measurement sector, send only the first one and leave the rest away.

For more details on sector configuration see examples below.

For complete workflow see example in section 3, page 11.

Telegram structure: sMN mLMPsetscancfg (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3	All	sMN	73 4D 4E
Command	Configuration of scan frequency and angular resolution	String	14	All	mLMPsetscancfg	6D 4C 4D 50 73 65 74 73 63 61 6E 63 66 67
Scan frequency	[1/100 Hz]	Uint_32	4	LMS1xx	25 Hz: +2500d (9C4h) 50 Hz: +5000d (1388h)	25 Hz: 00 00 09 C4 50 Hz: 00 00 13 88
				LMS5xx	25 Hz: +2500d (9C4h) 35 Hz: +3500d (DACh) 50 Hz: +5000d (1388h) 75 Hz: +7500d (1A0Bh) 100 Hz: +10000d (2710h)	25 Hz: 00 00 09 C4 35 Hz: 00 00 0D AC 50 Hz: 00 00 13 88 75 Hz: 00 00 1A 0B 100 Hz: 00 00 27 10
				NAV310 LD-OEM15xx	5 Hz ... 20 Hz: 500d ... 2000d (1F4h ... 7D0h)	5 Hz ... 20 Hz: 00 00 01 F4 ... 00 00 07 D0
				LD-LRS36xx	5 Hz ... 15 Hz: +500d ... +1500d (1F4h ... 5DCh)	5 Hz ... 15 Hz: 00 00 01 F4 ... 00 00 05 DC

Telegram structure: sMN mLMPsetscancfg (Authorized client)							
Telegram part		**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Number of active sectors		Indicates the number of active sectors (e. g. NAV310 with 2 active sectors out of available 4)	Int_16	2	LMS1xx LMS5xx	+1 (0001h)	0001
					NAV310 LD-OEM15xx LD-LRS36xx	+1 ... +4 (0001 ... 0004h)	0001 ... 0100 (binary)
	Angular resolution	[1/10000°] Same value for each sector required.	Uint_32 4		LMS1xx	0.25°: +2500d (9C4h) 0.5°: +5000d (1388h)	0.25°: 00 00 09 C4 0.5°: 00 00 13 88
					LMS5xx	0.1667°: +1667d (683h) 0.25°: +2500d (9C4h) 0.333°: +3333d (D05h) 0.5°: +5000d (1388h) 0.667°: +6667d (1A0Bh) 1°: +10000d (2710h)	0.1667°: 00 00 06 83 0.25°: 00 00 09 C4 0.333°: 00 00 0D 05 0.5°: 00 00 13 88 0.667°: 00 00 1A 0B 1°: 00 00 27 10
					NAV310 LD-OEM15xx LD-LRS36xx	0.125°... 1°: +1250d°... +10000d (4E2h°... 2710h)	0.125°... 1°: 00 00 04 E2 ... 00 00 27 10
	Start angle	[1/10000°] Value for start angle musst always be greater than Stop angle of previous sector. Set to 0 if sector is inactive (not used). **Values for LMSxxx are fixed.**	Int_32	4	LMS1xx	-450000d (FFF92230h)	FF F9 22 30
					LMS5xx	-50000d (FFFF3CB0h)	FF FF 3C B0
					NAV310 LD-OEM15x1 LD-LRS36x1	0°... +3600000d (0h ... 36EE80h)	00 00 00 00 ... 00 36 EE 80
					LD-OEM15x0 LD-LRS36x0	-900000d ... +2700000d (FFF24460h ... 41EB0h)	FF F2 44 60 ... 00 04 1E B0
	Stop angle	[1/10000°] Value for stop angle musst always be greater than start angle of previous sector. Set to 0 if sector is inactive (not used). **Values for LMSxxx are fixed.**	Int_32	4	LMS1xx	+2250000d (225510h)	00 22 55 10
					LMS5xx	+1850000d (1C3A90h)	00 1C 3A 90
					NAV310 LD-OEM15x1 LD-LRS36x1	0 ... +3600000d (0h ... 36EE80h)	00 00 00 00 ... 00 36 EE 80
					LD-OEM15x0 LD-LRS36x0	-900000d ... +2700000d (FFF24460h ... 41EB0h)	FF F2 44 60 ... 00 04 1E B0

Table 7: Telegram structure: sMN mLMPsetscancfg

Example for LMS1xx

Example for LMS1xx with 1 measurement sector of 270 °

ATTENTION: Scan angle can not be changed here, only in the data output! This applies for LMS1xx and LMS5xx series.

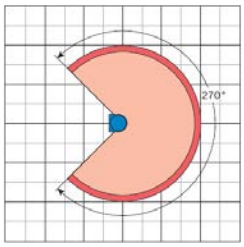
Scan frequency = 50 Hz

Sectors = 1 sector (This value is always 1 for these devices)

Angular resolution = 0,5 °

Start angle of sector = -45 ° (Fix values, angle not changeable)

Stop angle of sector = 225 ° (Fix values, angle not changeable)



CoLa A	ASCII	<STX>sMN{SPC}mLMPsetscancfg{SPC}+5000{SPC}+1{SPC}+5000{SPC}-450000{SPC}+2250000<ETX> Alternatively: <STX>sMN{SPC}mLMPsetscancfg{SPC}1388{SPC}1{SPC}1388{SPC}FFF92230{SPC}225510<ETX>
	Hex	02 73 4D 4E 20 6D 4C 4D 50 73 65 74 73 63 61 6E 63 66 67 20 2B 35 30 30 30 20 2B 31 20 2B 35 30 30 30 20 2D 34 35 30 30 30 30 20 2B 32 32 35 30 30 30 30 03 Alternatively: 02 73 4D 4E 20 6D 4C 4D 50 73 65 74 73 63 61 6E 63 66 67 20 31 33 38 38 20 31 20 31 33 38 38 20 46 46 4639 32 32 33 30 20 32 32 35 35 31 30 03
CoLa B	Binary	02 02 02 02 00 00 00 25 73 4D 4E 20 6D 4C 4D 50 73 65 74 73 63 61 6E 63 66 67 20 00 00 13 88 00 01 00 00 13 88 FF F9 22 30 00 22 55 10 21

Table 8: Example: sMN mLMPsetscancfg for LMS1xx with 1 measurement sector of 270°

Examples for LD-OEM1501, NAV310, LD-LR3601, LD-LR3611

Example for LD-xxx###1 with 1 measurement sector of 360 °

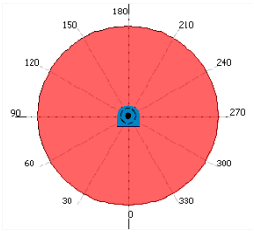
Scan frequency = 8 Hz

Sectors = 1 sector

Angular resolution = 0,25 °

Start angle of sector = 0 °

Stop angle of sector = 360 °



CoLa A	ASCII	<STX>sMN{SPC}mLMPsetscancfg{SPC}0320{SPC}01{SPC}09C4{SPC}0{SPC}0036EE80{SPC}09C4{SPC}0{SPC}0{SPC}09C4{SPC}0{SPC}0{SPC}09C4{SPC}0{SPC}0<ETX>
	Hex	02 73 4D 4E 20 6D 4C 4D 50 73 65 74 73 63 61 6E 63 66 67 20 30 33 32 30 20 30 31 20 30 39 43 34 20 30 20 30 30 33 36 45 45 38 30 20 30 39 43 34 20 30 20 30 20 30 39 43 34 20 30 20 30 20 30 39 43 34 20 30 20 30 20 30 03
CoLa B	Binary	02 02 02 02 00 00 00 55 73 4D 4E 20 6D 4C 4D 50 73 65 74 73 63 61 6E 63 66 67 20 00 00 03 20 00 01 00 00 09 C4 00 00 00 00 00 36 EE 80 00 00 09 C4 00 00 00 00 00 00 00 00 00 00 09 C4 00 00 00 00 00 00 00 00 00 00 00 09 C4 00 00 00 00 00 00 00 00 00 00 09 C4 00 00 00 00 00 00 00 00 E4

Table 9: Example: sMN mLMPsetscancfg for LD-XXX###1 with 1 measurement sector of 360°

Example for LD-XXX###1 with 1 measurement sector of 270 °

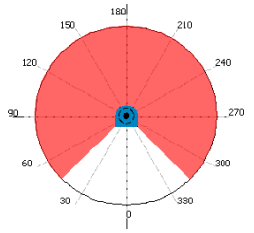
Scan frequency = 10 Hz

Sectors = 1 sector

Angular resolution = 0,50 °

Start angle of sector = +45 °

Stop angle of sector = +315 °



CoLa A	ASCII	<STX>sMN{SPC}mLMPsetscancfg{SPC}+1000{SPC}+1{SPC}+5000{SPC}+450000{SPC}+315000{SPC}+5000{SPC}0{SPC}0{SPC}+5000{SPC}0{SPC}0{SPC}+5000{SPC}0{SPC}0<ETX>
	Hex	02 73 4D 4E 20 6D 4C 4D 50 73 65 74 73 63 61 6E 63 66 67 20 2B 31 30 30 30 20 2B 31 20 2B 35 30 30 30 20 2B 34 35 30 30 30 30 20 2B 33 31 35 30 30 30 30 20 2B 35 30 30 30 20 30 30 30 30 30 30 30 20 30 30 30 30 30 30 20 2B 35 30 30 30 20 30 30 30 30 30 30 20 30 30 30 30 30 30 20 2B 35 30 30 30 20 30 30 30 30 30 30 20 30 30 30 30 30 30 03
CoLa B	Binary	02 02 02 02 00 00 00 49 73 4D 4E 20 6D 4C 4D 50 73 65 74 73 63 61 6E 63 66 67 20 00 00 03 E8 00 01 00 00 13 88 00 06 DD DE 00 30 10 B0 00 00 13 88 00 00 00 00 00 00 00 00 00 00 13 88 00 00 00 00 00 00 00 00 00 00 00 00 13 88 00 00 00 00 00 00 00 00 2C

Table 10: Example: sMN mLMPsetscancfg for LD-XXX###1 with 1 measurement sector of 270°

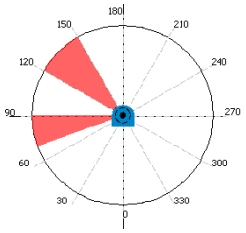
Example for LD-xxx###1 with 2 measurement sectors

Scan frequency = 8 Hz

Sectors = 2 sectors

Sector 1 = +70° ... +90°
Sector 2 = +120° ... +150°

Angular resolution = 0,25°



CoLa A	ASCII	<STX>sMN{SPC}mLMPsetscancfg{SPC}0320{SPC}02{SPC}09C4{SPC}+700000{SPC}+900000{SPC}09C4{SPC}+1200000{SPC}+1500000{SPC}09C4{SPC}0{SPC}0{SPC}09C4{SPC}0{SPC}0<ETX>
	Hex	02 73 4D 4E 20 6D 4C 4D 50 73 65 74 73 63 61 6E 63 66 67 20 30 33 32 30 20 30 32 20 30 39 43 34 20 2B 37 30 30 30 30 30 20 2B 39 30 30 30 30 30 20 30 39 43 34 20 2B 31 32 30 30 30 30 30 20 2B 31 35 30 30 30 30 30 20 30 39 43 34 20 30 20 30 39 43 34 20 30 20 30 03
CoLa B	Binary	02 02 02 02 00 00 00 49 73 4D 4E 20 6D 4C 4D 50 73 65 74 73 63 61 6E 63 66 67 20 00 00 03 20 00 02 00 00 09 C4 00 0A AE 60 00 0D BB A0 00 00 09 C4 00 12 4F 80 00 16 E3 60 00 00 09 C4 00 00 00 00 00 00 00 00 00 00 09 C4 00 00 00 00 00 00 00 00 E8

Table 11: Example: sMN mLMPsetscancfg for LD-XXX###1 with 2 measurement sectors

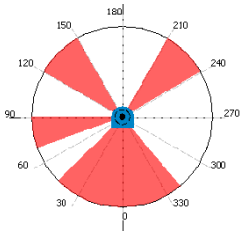
Example for LD-xxx###1 with 4 measurement sectors

Scan frequency = 8 Hz

Sectors = 4 sectors

Sector 1 = +320° ... +45°
Sector 2 = +70° ... +90°
Sector 3 = +120° ... +150°
Sector 4 = +210° ... +240°

Angular resolution = 0,25°



CoLa A	ASCII	<STX>sMN{SPC}mLMPsetscancfg{SPC}0320{SPC}04{SPC}09C4{SPC}+3200000{SPC}+450000{SPC}09C4{SPC}+700000{SPC}+900000{SPC}09C4{SPC}+1200000{SPC}+1500000{SPC}09C4{SPC}+2100000{SPC}+2400000<ETX>
	Hex	02 73 4D 4E 20 6D 4C 4D 50 73 65 74 73 63 61 6E 63 66 67 20 30 33 32 30 20 30 34 20 30 39 43 34 20 2B 33 32 30 30 30 30 30 20 2B 34 35 30 30 30 30 20 30 39 43 34 20 2B 37 30 30 30 30 30 20 2B 39 30 30 30 30 30 20 30 39 43 34 20 2B 31 32 30 30 30 30 30 20 2B 31 35 30 30 30 30 30 20 30 39 43 34 20 2B 32 31 30 30 30 30 30 20 2B 32 34 30 30 30 30 30 03
CoLa B	Binary	02 02 02 02 00 00 00 49 73 4D 4E 20 6D 4C 4D 50 73 65 74 73 63 61 6E 63 66 67 20 00 00 03 20 00 04 00 00 09 C4 00 30 D4 00 00 06 DD D0 00 00 09 C4 00 0A AE 60 00 0D BB A0 00 00 09 C4 00 12 4F 80 00 16 E3 60 00 00 09 C4 00 20 0B 20 00 24 9F 00 B1

Table 12: Example: sMN mLMPsetscancfg for LD-XXX###1 with 4 measurement sectors

Examples for LD-OEM1500 and LD-LR3600

Example for LD-xxx###0 with 1 measurement sector of 360 °

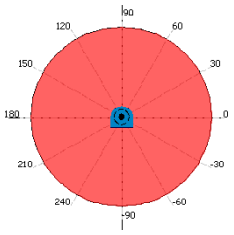
Scan frequency = 8 Hz

Sectors = 1 sector

Angular resolution = 0,25 °

Start angle of sector = –90 °

Stop angle of sector = +270 °



CoLa A	ASCII	<STX>sMN{SPC}mLMPsetscancfg{SPC}0320{SPC}01{SPC}09C4{SPC}-900000{SPC}+2700000{SPC}09C4{SPC}00000000{SPC}000000{SPC}09C4{SPC}000000{SPC}0000000{SPC}09C4{SPC}000000{SPC}000000<ETX>
	Hex	02 73 4D 4E 20 6D 4C 4D 50 73 65 74 73 63 61 6E 63 66 67 20 30 33 32 30 20 30 31 20 30 39 43 34 20 2D 39 30 30 30 30 30 20 2B 32 37 30 30 30 30 30 20 30 39 43 34 20 30 30 30 30 30 30 30 20 30 30 30 30 30 30 20 30 39 43 34 20 30 30 30 30 30 30 20 30 30 30 30 30 30 20 30 39 43 34 20 30 30 30 30 30 30 20 30 30 30 30 30 30 03
CoLa B	Binary	02 02 02 02 00 00 00 49 73 4D 4E 20 6D 4C 4D 50 73 65 74 73 63 61 6E 63 66 67 20 00 00 03 20 00 01 00 00 09 C4 FF F2 44 60 00 29 32 E0 00 00 09 C4 00 00 00 00 00 00 00 00 00 09 C4 00 00 00 00 00 00 00 00 00 00 09 C4 00 00 00 00 00 00 00 00 00 09 C4 00 00 00 00 00 00 00 00 A3

Table 13: Example: sMN mLMPsetscancfg for LD-XXX###0 with 1 measurement sector of 360°

Example for LD-xxx###0 with 1 measurement sector of 270 °

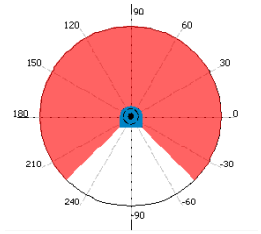
Scan frequency = 10 Hz

Sectors = 1 sector

Angular resolution = 0,50 °

Start angle of sector = –45 °

Stop angle of sector = +225 °



CoLa A	ASCII	<STX>sMN{SPC}mLMPsetscancfg{SPC}+1000{SPC}+1{SPC}+5000{SPC}-450000{SPC}+2250000{SPC}+5000{SPC}0{SPC}0{SPC}+5000{SPC}0{SPC}0{SPC}0{SPC}+5000{SPC}0{SPC}0<ETX>
	Hex	02 73 4D 4E 20 6D 4C 4D 50 73 65 74 73 63 61 6E 63 66 67 20 2B 31 30 30 30 20 2B 31 20 2B 35 30 30 30 20 2D 34 35 30 30 30 30 20 2B 32 32 35 30 30 30 30 20 2B 35 30 30 30 20 2D 34 35 30 30 30 30 20 2B 32 32 35 30 30 30 30 20 2B 35 30 30 30 20 30 20 30 20 2B 35 30 30 30 20 30 20 30 20 2B 35 30 30 30 20 30 20 30 03
CoLa B	Binary	02 02 02 02 00 00 00 49 73 4D 4E 20 6D 4C 4D 50 73 65 74 73 63 61 6E 63 66 67 20 00 00 03 E8 00 01 00 00 13 88 FF F9 22 30 00 22 55 10 00 00 13 88 00 00 00 00 00 00 00 00 00 13 88 00 00 00 00 00 00 00 00 00 00 13 88 00 00 00 00 00 00 00 00 CA

Table 14: Example: sMN mLMPsetscancfg for LD-XXX###0 with 1 measurement sector of 270°

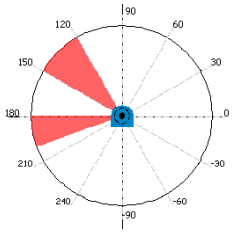
Example for LD-xxx###0 with 2 measurement sectors

Scan frequency = 8 Hz

Sectors = 2 sectors

Sector 1 = +120 ° ... +150 °
Sector 2 = +180 ° ... +200 °

Angular resolution = 0,25 °



CoLa A	ASCII	<STX>sMN{SPC}mLMPsetscancfg{SPC}320{SPC}2{SPC}9C4{SPC}+1200000{SPC}+1500000{SPC}9C4{SPC}+1800000{SPC}+2000000{SPC}9C4{SPC}0{SPC}0{SPC}9C4{SPC}0{SPC}0<ETX>
	Hex	02 73 4D 4E 20 6D 4C 4D 50 73 65 74 73 63 61 6E 63 66 67 20 33 32 30 20 32 20 39 43 34 20 2B 31 32 30 30 30 30 30 20 2B 31 35 30 30 30 30 30 20 39 43 34 20 2B 31 38 30 30 30 30 30 20 2B 32 30 30 30 30 30 30 20 39 43 34 20 30 20 30 20 39 43 34 20 30 20 30 03
CoLa B	Binary	02 02 02 02 00 00 00 49 73 4D 4E 20 6D 4C 4D 50 73 65 74 73 63 61 6E 63 66 67 20 00 00 03 20 00 02 00 00 09 C4 00 12 4F 80 00 16 E3 60 00 00 09 C4 00 1B 77 40 00 1E 84 80 00 00 09 C4 00 00 00 00 00 00 00 00 00 00 09 C4 00 00 00 00 00 00 00 00 0C

Table 15: Example: sMN mLMPsetscancfg for LD-XXX###0 with 2 measurement sectors

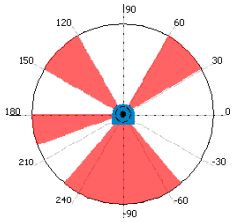
Example for LD-xxx###0 with 4 measurement sectors

Scan frequency = 8 Hz

Sectors = 4 sectors

Sector 1 = +230 ° ... –50°
Sector 2 = +30 ° ... +60 °
Sector 3 = +120 ° ... +150 °
Sector 4 = +210 ° ... +200 °

Angular resolution = 0,25 °



CoLa A	ASCII	<STX>sMN{SPC}mLMPsetscancfg{SPC}320{SPC}4{SPC}9C4{SPC}+2300000{SPC}-500000{SPC}9C4{SPC}+300000{SPC}+600000{SPC}9C4{SPC}+1200000{SPC}+1500000{SPC}9C4{SPC}+1800000{SPC}+2000000<ETX>
	Hex	02 73 4D 4E 20 6D 4C 4D 50 73 65 74 73 63 61 6E 63 66 67 20 33 32 30 20 34 20 39 43 34 20 2B 32 33 30 30 30 30 30 20 2D 35 30 30 30 30 30 20 39 43 34 20 2B 33 30 30 30 30 30 20 2B 36 30 30 30 30 30 20 39 43 34 20 2B 31 32 30 30 30 30 30 20 2B 31 35 30 30 30 30 30 20 39 43 34 20 2B 31 38 30 30 30 30 30 20 2B 32 30 30 30 30 30 30 03
CoLa B	Binary	02 02 02 02 00 00 00 49 73 4D 4E 20 6D 4C 4D 50 73 65 74 73 63 61 6E 63 66 67 20 00 00 03 20 00 04 00 00 09 C4 00 23 18 60 FF F8 5E E0 00 00 09 C4 00 04 93 E0 00 09 27 C0 00 00 09 C4 00 12 4F 80 00 16 E3 60 00 00 09 C4 00 1B 77 40 00 1E 84 80 71

Table 16: Example: sMN mLMPsetscancfg for LD-XXX###0 with 4 measurement sectors

Telegram structure: sAN mLMPsetscancfg						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sAN	73 41 4E
Command	Info of scan frequency and angular resolution	String	14	All	mLMPsetscancfg	6D 4C 4D 50 73 65 74 73 63 61 6E 63 66 67
Status code	Accepted when value is 0	Enum_8	1	All	No error: 0 Frequency error: 1 Resolution error: 2 Resolution and scanarea error: 3 Scanarea error: 4 Other errors: 5	No error: 00 Frequency error: 01 Resolution error: 02 Resolution and scan area error: 03 Scanarea error: 04 Other errors: 05
Scan frequency	[1/100 Hz]	Uint_32	4	LMS1xx	25 Hz: +2500d (9C4h) 50 Hz: +5000d (1388h)	25 Hz: 00 00 09 C4 50 Hz: 00 00 13 88
				LMS5xx	25 Hz: +2500d (9C4h) 35 Hz: +3500d (DACh) 50 Hz: +5000d (1388h) 75 Hz: +7500d (1A0Bh) 100 Hz: +10000d (2710h)	25 Hz: 00 00 09 C4 35 Hz: 00 00 0D AC 50 Hz: 00 00 13 88 75 Hz: 00 00 1A 0B 100 Hz: 00 00 27 10
				NAV310 LD-OEM 15xx	5 Hz ... 20 Hz: +500d ... +2000d (1F4h ... 7D0h)	5 Hz ... 20 Hz: 00 00 01 F4 ... 00 00 07 D0
				LD-LRS 36xx	5 Hz ... 15 Hz: +500d ... +1500d (1F4h ... 5DCh)	5 Hz ... 15 Hz: 00 00 01 F4 ... 00 00 05 DC
Number of active sectors	Indicates the number of active sectors	Int_16	2	LMS1xx LMS5xx	1 (0001h)	0001
				NAV310 LD-OEM 15xx LD-LRS 36xx	1 ... 4 (0001h ... 0004h)	0001 ... 0100 (binary)

Telegram structure: sAN mLMPsetscancfg							
Telegram part		Description	Variable	Length	Sensor	Values CoLa A (ASCII)	Values CoLa B (Binary)
Per sector (active and inactive sectors)	Angular resolution	[1/10000°]	Uint_32	4	LMS1xx	0.25°: +2500d (9C4h) 0.5°: +5000d (1388h)	0.25°: 00 00 09 C4 0.5°: 00 00 13 88
					LMS5xx	0.1667°: +1667d (683h) 0.25°: +2500d (9C4h) 0.333°: +3333d (D05h) 0.5°: +5000d (1388h) 0.667°: +6667d (1A0Bh) 1°: +10000d (2710h)	0.1667°: 00 00 06 83 0.25°: 00 00 09 C4 0.333°: 00 00 0D 05 0.5°: 00 00 13 88 0.667°: 00 00 1A 0B 1°: 00 00 27 10
					NAV310 LD-OEM 15xx LD-LRS 36xx	0.125°... 1°: +1250d°... +10000d (4E2h°... 2710h)	0.125°... 1: 00 00 04 E2 ... 00 00 27 10
	Start angle	[1/10000°]	Int_32	4	LMS1xx	-450000d (FFF92230h)	FF F9 22 30
					LMS5xx	-50000d (FFFF3CB0h)	FF FF 3C B0
					NAV310 LD-OEM 15x1 LD-LRS 36x1	0°... +3600000d (0h ... 36EE80h)	00 00 00 00 ... 00 36 EE 80
					LD-OEM 15x0 LD-LRS 36x0	-900000d ... +2700000d (FFF24460h ... 41EB0h)	FF F2 44 60 ... 00 04 1E B0
	Stop angle	[1/10000°]	Int_32	4	LMS1xx	+2250000d (225510h)	00 22 55 10
					LMS5xx	+1850000d (1C3A90h)	00 1C 3A 90
					NAV310 LD-OEM 15x1 LD-LRS 36x1	0 ... +3600000d (0h ... 36EE80h)	00 00 00 00 ... 00 36 EE 80
					LD-OEM 15x0 LD-LRS 36x0	-900000d ... +2700000d (FFF24460h ... 41EB0h)	FF F2 44 60 ... 00 04 1E B0

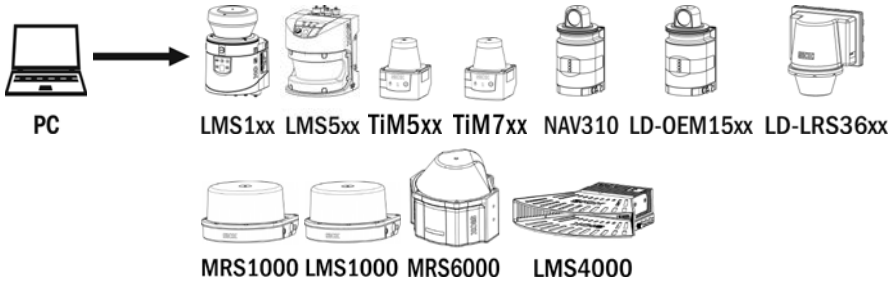
Table 17: Telegram structure: sAN mLMPsetscancfg

Example: sAN mLMPsetscancfg

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

Table 18: Example: sAN mLMPsetscancfg

4.2.2 Read for frequency and angular resolution



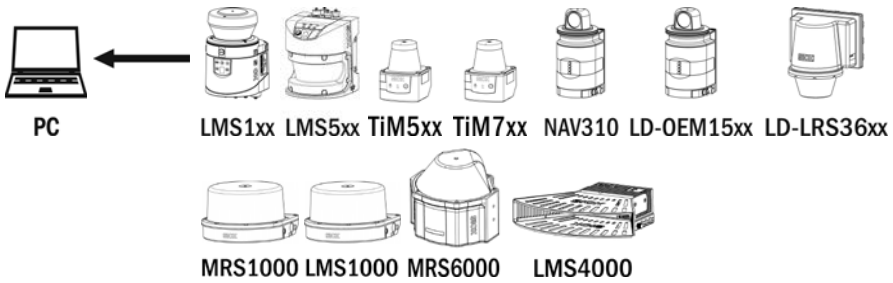
Telegram structure: sRN LMPscancfg						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sRN	73 52 4E
Command	Info of scan frequency and angular resolution	String	10	All	LMPscancfg	4C 4D 50 73 63 61 6E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

Table 19: Telegram structure: sRN LMPscancfg

Example for LMS100: sRN LMPscancfg

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

Table 20: Example for LMS100: sRN LMPscancfg



Telegram structure: sRA LMPscancfg						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sRA	73 52 41
Command	Info of scan frequency and angular resolution	String	10	All	LMPscancfg	4C 4D 50 73 63 61 6E 63 66 67

Telegram structure: sRA LMPscancfg						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Scan frequency	[1/100 Hz]	Uint_32	4	LMS1xx	25 Hz: +2500d (9C4h) 50 Hz: +5000d (1388h)	25 Hz: 00 00 09 C4 50 Hz: 00 00 13 88
				LMS5xx	25 Hz: +2500d (9C4h) 35 Hz: +3500d (DACh) 50 Hz: +5000d (1388h) 75 Hz: +7500d (1A0Bh) 100 Hz: +10000d (2710h)	25 Hz: 00 00 09 C4 35 Hz: 00 00 0D AC 50 Hz: 00 00 13 88 75 Hz: 00 00 1A 0B 100 Hz: 00 00 27 10
				TiMxxx	15 Hz: +1500d (5DCh)	15 Hz: 00 00 05 DC
				NAV310 LD-OEM 15xx	5 Hz ... 20 Hz: +500d ... +2000d (1F4h ... 7D0h)	5 Hz ... 20 Hz: 00 00 01 F4 ... 00 00 07 D0
				LD-LRS 36xx	5 Hz ... 15 Hz: +500d ... +1500d (1F4h ... 5DCh)	5 Hz ... 15 Hz: 00 00 01 F4 ... 00 00 05 DC
				MRS1000	50 Hz: +5000d (1388h)	50 Hz: 00 00 13 88
				LMS1000	150 Hz: +15000d (3A98h)	
				MRS6000	10 Hz: +1000d (3E8h)	10 Hz: 00 00 03 E8
Number of sectors	Indicates the number of sectors. The subsequent values will be transmitted 1 ... 4 accordingly.	Int_16	2	LMS1xx LMS5xx TiMxxx MRS1000 LMS1000 MRS6000	Sector 1: 0001h	Sector 1: 0001
				NAV310 LD-OEM 15xx LD-LRS 36xx	Sector 1: 0001h Sector 2: 0002h Sector 3: 0003h Sector 4: 0004h	Sector 1: 0001 Sector 2: 0010 Sector 3: 0011 Sector 4: 0100
Angular resolution	[1/10000°]	Uint_32	4	LMS1xx	0.25°: +2500d (9C4h) 0.5°: +5000d (1388h)	0.25°: 00 00 09 C4 0.5°: 00 00 13 88
				LMS5xx	0.1667°: +1667d (683h) 0.25°: +2500d (9C4h) 0.333°: +3333d (D05h) 0.5°: +5000d (1388h) 0.667°: +6667d (1A0Bh) 1°: +10000d (2710h)	0.1667°: 00 00 06 83 0.25°: 00 00 09 C4 0.333°: 00 00 0D 05 0.5°: 00 00 13 88 0.667°: 00 00 1A 0B 1°: 00 00 27 10
				TiMxxx	0.333°: +3333d (D05h) 1°: +10000d (2710h)	0.333°: 00 00 0D 05 1°: 00 00 27 10
				NAV310 LD-OEM 15xx LD-LRS 36xx	0.125°... 1°: +1250d°... +10000d (4E2h°... 2710h)	0.125°... 1: 00 00 04 E2 ... 00 00 27 10

Telegram structure: sRA LMPscancfg						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
				MRS1000	0.25°: +2500d (9C4h)	0.25°: 00 00 09 C4
				LMS1000	0.75°: +7500d (1D4Ch)	
				MRS6000	0.13°: +1300d (514h)	0.13°: 00 00 05 14
Start angle	[1/10000°]	Int_32	4	LMS1xx TiMxxx	-450000d ... +2250000d (FFF92230h ... 225510h)	FF F9 22 30 ... 00 22 55 10
				LMS5xx	-50000d ... +1850000d (FFFF3CB0h ... 1C3A90h)	FF FF 3C B0 ... 00 1C 3A 90
				NAV310 LD-OEM 15x1 LD-LRS 36x1	0°... +3600000d (0h ... 36EE80h)	00 00 00 00 ... 00 36 EE 80
				LD-OEM 15x0 LD-LRS 36x0	-900000d ... +2700000d (FFF24460h ... 41EB0h)	FF F2 44 60 ... 00 04 1E B0
				MRS1000	-475000d (FFF8C088h)	FF F8 C0 88
				LMS1000	-480000d (FFF8AD00h)	
				MRS6000	+300000d (493E0h)	00 04 93 E0
Stop angle	[1/10000°]	Int_32	4	LMS1xx TiMxxx	-450000d ... +2250000d (FFF92230h ... 225510h)	FF F9 22 30 ... 00 22 55 10
				LMS5xx	-50000d ... +1850000d (FFFF3CB0h ... 1C3A90h)	FF FF 3C B0 ... 00 1C 3A 90
				NAV310 LD-OEM 15x1 LD-LRS 36x1	0 ... +3600000d (0h ... 36EE80h)	00 00 00 00 ... 00 36 EE 80
				LD-OEM 15x0 LD-LRS 36x0	-900000d ... +2700000d (FFF24460h ... 41EB0h)	FF F2 44 60 ... 00 04 1E B0
				MRS1000	+2275000d (22B6B8h)	00 22 B6 B8
				LMS1000	+2280000d (22CA40h)	
				MRS6000	+1500000d (16E360h)	00 16 E3 60

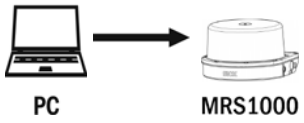
Table 21: Telegram structure: sRA LMPscancfg

Example: sRA LMPscancfg

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

Table 22: Example: sRA LMPscancfg

4.2.3 Alignment mode (one layer activation for adjustment)



Telegram structure: sWN MMAlignmentMode (Service) (sMN SetAccessMode 04 81BE23AA)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Set device to alignment mode	String	15	All	MMAlignmentMode	4D 4D 41 6C 69 67 6E 6D 65 6E 74 4D 6F 64 65
Layer activation					all Layer: 0 red Layer -2.5°: 1 blue Layer 0°: 2 green Layer +2.5°: 3 yellow Layer +5: 4	00 01 02 03 04

Table 23: Telegram structure: sWN MMAlignmentMode

Example: sWN MMAlignmentMode 2

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

Table 24: Example: sWN MMAlignmentMode

Telegram structure: sWA MMAlignmentMode						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command	Set device to standby	String	15	All	MMAlignmentMode	4D 4D 41 6C 69 67 6E 6D 65 6E 74 4D 6F 64 65

Table 25: Telegram structure: sWA MMAlignmentMode

Example: sWA MMAlignmentMode

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

CoLa B	Binary	02 02 02 02 00 00 00 14 73 57 41 20 4D 4D 41 6C 69 67 6E 6D 65 6E 74 4D 6F 64 65 20 19

Table 26: Example: sWA MMAlignmentMode

4.2.4 Set scan configuration

Sets the device to an defined scan configuration, consisting of scan frequency, angular resolution, sector definition and interlace mode.

Telegram structure: sMN mCLsetscancfglist						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3	All	sMN	73 4D 4E
Command	Set scan configuration	String	17	All	mCLsetscancfglist	6D 43 4C 73 65 74 73 63 61 6E 63 66 67 6C 69 73 74
Mode	Interlace mode (see table below)	Enum_8	1	All	+1d, +2d, +3d ... (01h, 02h, 03h ...)	01, 02, 03 ...

Table 27: Telegram structure: sMN mCLsetscancfglist

Interlace mode

The interlace mode allows to achieve a higher angular resolution by combining scans with lower resolution. The individual scans are shifted to each other.

The command *mCLsetscancfglist* selects combinations of scan resolution, scan frequency and resolution. If the scan area will not match to the application then an adjustment is possible by the command “mLMPsetscancfg” (see section 4.2.1 “Set frequency and angular resolution/measurement sectors” on page 14).

Mode	Inter-laced	Scan freq.	Result. scan freq.	Reso-lution	Total Resol.	Field of view	Sector	LRS 3601 3611	OEM 1501	NAV 310	LRS 3600	OEM 1500
1	0x	8 Hz	8 Hz	0.25°	0.25°	360°	0 ... 360°	x	x	x	(x)	(x)
2	0x	15 Hz	15 Hz	0.5°	0.5°	360°	0 ... 360°	x	x	x	(x)	(x)
3	0x	10 Hz	10 Hz	0.25°	0.25°	300°	30 ... 330°	x	x	x	x	x
4	0x	5 Hz	5 Hz	0.125°	0.125°	300°	30 ... 330°	x	x	x	x	x
5	0x	6 Hz	6 Hz	0.1875°	0.1875°	360°	0 ... 360°	x	x	x	(x)	(x)
6	0x	8Hz	8 Hz	0.25°	0.25°	359.5°	0.25° ...359.25°				x	X
8	0x	15 Hz	15 Hz	0.375°	0,375°	300°	30...330°	x	X	x	x	x
9	0x	15 Hz	15 Hz	0.5°	0.5°	359°	0.5 ... . 359.5°				x	x
21	0x	20 Hz	20 Hz	0.5°	0.5°	300°	30 ... 330°		X	x		x
22	0x	20 Hz	20 Hz	0.75°	0.75°	360°	0 ... 360°		x	x		(x)
44	4x	10 Hz	2.5 Hz	0.25°	0.0625°	300°	30 ... 330°	x	x		(x)	(x)
46	4x	16 Hz	4 Hz	0.5°	0.125°	300°	30 ... 330°		x			(x)

Table 28: Interlace mode for sMN mCLsetscancfglist

(x): Only at raw data scan (field application)

Example: Set scan configuration 1: sMN mCLsetscancfglist 1

CoLa A	ASCII	<STX>sMN{SPC}mCLsetscancfglist{SPC}1<ETX>
	Hex	02 73 4D 4E 20 6D 43 4C 73 65 74 73 63 61 6E 63 66 67 6C 69 73 74 20 31 03
CoLa B	Binary	02 02 02 02 00 00 00 17 20 73 4D 4E 20 6D 43 4C 73 65 74 73 63 61 6E 63 66 67 6C 69 73 74 20 01 0F

Table 29: Example: Set scan configuration 1: sMN mCLsetscancfglist 1

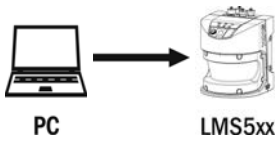
Telegram structure: sAN mCLsetscancfglist						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sAN	73 41 4E
Command	Confirm scan configuration	String	17	All	mCLsetscancfglist	6D 43 4C 73 65 74 73 63 61 6E 63 66 67 6C 69 73 74
Status code	Wrong setting	Enum_8	1	All	Ok: 0 Error frequency: 1 Error resolution: 2 Err. res. and freq.: 3 Err. scan field: 4 Error: 5	Ok: 00 Error frequency: 01 Error resolution: 02 Err. res. and freq.: 03 Err. scan field: 04 Error: 05

Table 30: Telegram structure: sAN mCLsetscancfglist

Example: sAN mCLsetscancfglist Ok

CoLa A	ASCII	<STX>sAN{SPC}mCLsetscancfglist{SPC}0<ETX>
	Hex	02 73 41 4E 20 6D 43 4C 73 65 74 73 63 61 6E 63 66 67 6C 69 73 74 20 30 03
CoLa B	Binary	02 02 02 02 00 00 00 17 73 41 4E 20 6D 43 4C 73 65 74 73 63 61 6E 63 66 67 6C 69 73 74 20 00 10

Table 31: Example: sAN mCLsetscancfglist Ok



Note

After sending this telegram, it will take 30 seconds to process the new configuration in the sensor.

Telegram structure: sMN mCLsetscancfglist (user level 'authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sMN	73 4D 4E
Command	Set scan configuration	String	17	All	mCLsetscancfglist	6D 43 4C 73 65 74 73 63 61 6E 63 66 67 6C 69 73 74

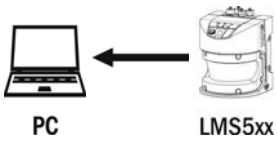
Telegram structure: sMN mCLsetscancfglist (user level 'authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Mode	Scan configuration	Enum_8	1	All	0d= 25Hz, 0.167°	00
					1d = 25Hz, 0.25°	01
					2d = 35Hz, 0.25°	02
					3d = 35Hz, 0.5°	03
					4d = 50Hz, 0.333°	04
					5d = 50Hz, 0.5°	05
					6d = 75Hz, 0.5°	06
					7d = 75Hz, 1.0°	07
					8d = 100Hz, 0.667°	08
					9d = 100Hz, 1.0°	09
					10d = 50Hz, 0.167° interl.	0A
					11d = 75Hz, 0.25° interl.	0B
					12d = 100Hz, 0.167° interl.	0C
					13d = 100Hz, 0.333° interl.	0D
					14d = 100Hz, 0.5° interl.	0E

Table 32: Telegram structure: sMN mCLsetscancfglist

Example: sMN mCLsetscancfglist 5

CoLa A	ASCII	<STX>sMN{SPC}mCLsetscancfglist{SPC}5<ETX>
	Hex	02 73 4D 4E 20 6D 43 4C 73 65 74 73 63 61 6E 63 66 67 6C 69 73 74 20 35 03
CoLa B	Binary	02 02 02 02 00 00 00 17 73 4D 4E 20 6D 43 4C 73 65 74 73 63 61 6E 63 66 67 6C 69 73 74 20 05 0A

Table 33: Example: sMN mCLsetscancfglist 5



Telegram structure: sAN mCLsetscancfglist (user level 'authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sAN	73 41 4E
Command	Confirm scan configuration	String	17	All	mCLsetscancfglist	4F 50 68 65 61 74 73 74 61 74 65 65 78 74
Status code	Result	Enum_8	1	All	0d = Ok	00
					1d = Frequency error	01
					2d = Resolution error	02
					3d = Frequency and resolution combination error	03
					4d = Range error	04
					5d = General error	05

Table 34: Telegram structure: sAN mCLsetscancfglist

Example: sAN mCLsetscancfglist 0

CoLa A	ASCII	<STX>sAN{SPC}mCLsetscancfglist{SPC}0<ETX>
	Hex	02 73 41 4E 20 4F 50 68 65 61 74 73 74 61 74 65 65 78 74 20 30 03
CoLa B	Binary	02 02 02 02 00 00 00 14 73 41 4E 20 4F 50 68 65 61 74 73 74 61 74 65 65 78 74 20 00 03

Table 35: Example: sAN mCLsetscancfglist 0

4.2.5 Activate standby mode

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

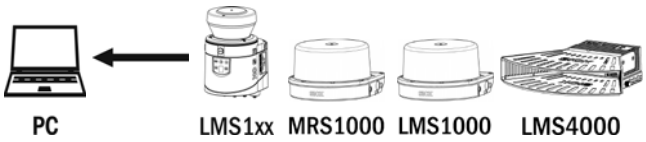
Telegram structure: sMN LMCstandby (All = Authorized client; LMS4000 = Operator)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3	All	sMN	73 4D 4E
Command	Set device to standby	String	10	All	LMCstandby	4C 4D 43 73 74 61 6E 64 62 79

Table 36: Telegram structure: sMN LMCstandby

Example: sMN LMCstandby

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

Table 37: Example: sMN LMCstandby



Telegram structure: sAN LMCstandby						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sAN	73 41 4E
Command	Set device to standby	String	10	All	LMCstandby	4C 4D 43 73 74 61 6E 64 62 79
Status code	Accepted when value is 0	Enum_8	1	All	No error: 0	No error: 00

Table 38: Telegram structure: sAN LMCstandby

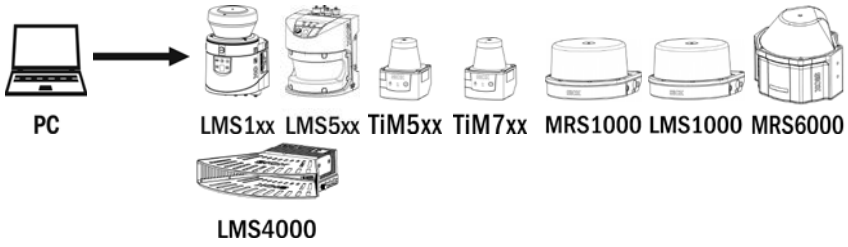
Example: sAN LMCstandby

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

Table 39: Example: sAN LMCstandby

4.2.6 Start measurement

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



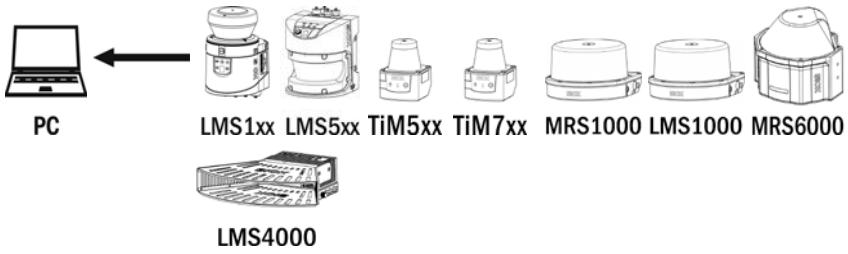
Telegram structure: sMN LMCstartmeas (All = Authorized client; LMS4000 = Operator)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3	All	sMN	73 4D 4E
Command	Start measurement	String	12	All	LMCstartmeas	4C 4D 43 73 74 61 72 74 6D 65 61 73

Table 40: Telegram structure: sMN LMCstartmeas

Example: sMN LMCstartmeas

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

Table 41: Example: sMN LMCstartmeas



Telegram structure: sAN LMCstartmeas						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sAN	73 41 4E
Command	Start measurement	String	12	All	LMCstartmeas	4C 4D 43 73 74 61 72 74 6D 65 61 73
Status code	Accepted when value is 0	Enum_8	1	All	No error: 0 Not allowed: 1	No error: 00 Not allowed: 01

Table 42: Telegram structure: sAN LMCstartmeas

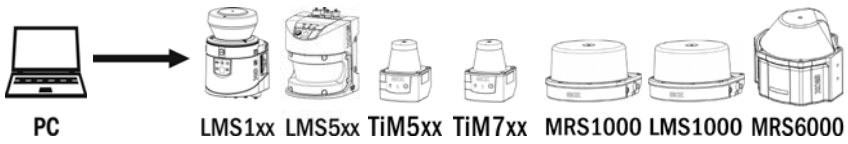
Example: sAN LMCstartmeas

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

Table 43: Example: sAN LMCstartmeas

4.2.7 Stop measurement

Shut off the laser and stop the motor of the device



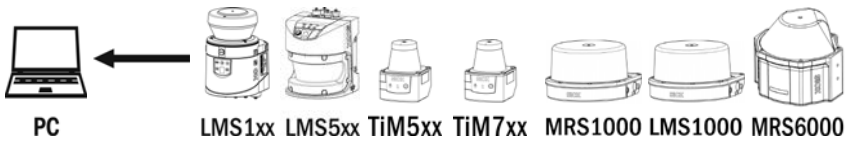
Telegram structure: sMN LMCstopmeas (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3	All	sMN	73 4D 4E
Command	Stop measurement	String	11	All	LMCstopmeas	4C 4D 43 73 74 6F 70 6D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

Table 44: Telegram structure: sMN LMCstopmeas

Example: sMN LMCstopmeas

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

Table 45: Example: sMN LMCstopmeas



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

Table 46: Telegram structure: sAN LMCstopmeas

Example: sAN LMCstopmeas

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

Table 47: Example: sAN LMCstopmeas

4.2.8 Autostart measurement

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

Table 48: Telegram structure: sWN LMPautostartmeas

Example: sWN LMPautostartmeas 1

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

Table 49: Example: sWN LMPautostartmeas 1

This parameter defines whether the scanner will start directly rotate and measure when powering up or remain in idle mode. The changed setting (saved with will the command sMN mEEWriteall) will be then be active with the next power-up cycle.

Telegram structure: sWA LMPautostartmeas						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command	Autostart measure-ment	String	14	All	LMPautostartmeas	4C 4D 43 73 74 61 72 74 6D 65 61 73

Table 50: Telegram structure: sWA LMDautostartmeas

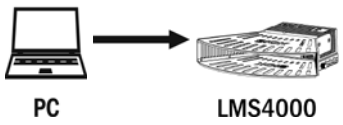
Example: sWA LMPautostartmeas

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

Table 51: Example: sWA LMPautostartmeas

4.2.9 Laser Control

Define if laser is always on or rather switched on and off by specific trigger signal. Also select delay times and timeout.



Telegram structure: sWN IOlasc (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Autostart measurement	String	6	All	IOlasc	49 4F 6C 61 73 63
Trigger Source	Select Trigger Source	Uint_8	1	All	Free running: 0 Software: 1 Input 1: 2 Input 2: 3 Input 1 or 2: 4	00 01 02 03 04
Delay on Start	Delay on Start in ms	Uint_16	2	All	0 ... +65535d (0 ... FFFFh)	00 00 ... FF FF
Delay on Stop	Delay on Stop in ms	Uint_16	2	All	0 ... +65535d (0 ... FFFFh)	00 00 ... FF FF

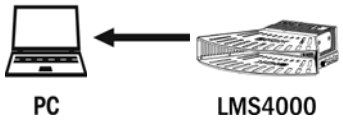
Telegram structure: sWN IOlasc (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Laser Timeout	Laser shuts down after defined time in s, if trigger signal does not disappear	Uint_16	2	All	0 = inactive 1 ... +65535d (0 ... FFFFh)	00 00 00 01 ... FF FF
Delay settings	Reserved	Uint_8	1	All	Always: 0	00

Table 52: Telegram structure: sWN IOlasc

Example: sWN IOlasc 1 +500 0 0 0

CoLa A	ASCII	<STX>sMN{SPC}IOlasc{SPC}1{SPC}+500{SPC}0{SPC}0{SPC}0<ETX>
	Hex	02 73 57 4E 20 49 4F 6C 61 73 63 20 31 20 2B 35 30 30 20 30 20 30 20 30 03
CoLa B	Binary	02 02 02 02 00 00 00 16 73 57 4E 20 49 4F 6C 61 73 63 20 01 01 F4 00 00 00 00 00 00 00 20 A5

Table 53: Example: sWN IOlasc with Software Trigger and 0.5 s delay on start.



Telegram structure: sWA IOlasc						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command	Autostart measure-ment	String	14	All	IOlasc	49 4F 6C 61 73 63

Table 54: Telegram structure: sWA IOlasc

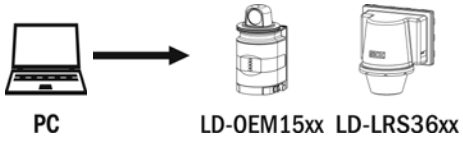
Example: sWA IOlasc

CoLa A	ASCII	<STX>sWA{SPC}IOlasc<ETX>
	Hex	02 73 57 41 20 49 4F 6C 61 73 63 03
CoLa B	Binary	02 02 02 02 00 00 00 0B 73 57 41 20 49 4F 6C 61 73 63 20 7E

Table 55: Example: sWA IOlasc

4.2.10 Activate/deactivate field application

With the aid of the integrated field application, the LD-OEM1500/LD-LRS3600 evaluates up to four evaluation fields within its scan area.



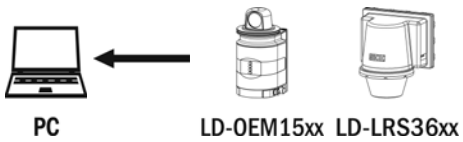
Telegram structure: sWN CLApplication (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Activate/deactivate field application	String	13	All	CLApplication	43 4C 41 70 70 6C 69 63 61 74 69 6F 6E
Mode	Application	Enum_ 16	2	All	Scan only: 00 Field application: 11	Scan only: 00 00 Field application: 00 11

Table 56: Telegram structure: sWN CLApplication

Example: Activate the field application: sWN CLApplication 11

CoLa A	ASCII	<STX>sWN{SPC}CLApplication{SPC}11<ETX>
	Hex	02 73 57 4E 20 43 4C 41 70 70 6C 69 63 61 74 69 6F 6E 20 31 31 03
CoLa B	Binary	02 02 02 02 00 00 00 17 73 57 4E 20 43 4C 41 70 70 6C 69 63 61 74 69 6F 6E 20 00 11 1F

Table 57: Example: Activate the field application: sWN CLApplication 11



Telegram structure: sWA CLApplication						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command	Activate/deactivate field application	String	13	All	CLApplication	43 4C 41 70 70 6C 69 63 61 74 69 6F 6E

Table 58: Telegram structure: sWA CLApplication

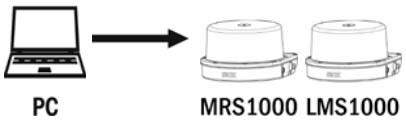
Example: sWA CLApplication correct and accepted

CoLa A	ASCII	<STX>sWA{SPC}CLApplication<ETX>
	Hex	02 73 57 41 20 43 4C 41 70 70 6C 69 63 61 74 69 6F 6E 03
CoLa B	Binary	02 02 02 02 00 00 00 11 73 57 41 20 43 4C 41 70 70 6C 69 63 61 74 69 6F 6E 1A

Table 59: Example: sWA CLApplication correct and accepted

4.2.11 Application selection and switching

Selection between the field application (default) and the ranging application in the device.



Telegram structure: sWN SetActiveApplications (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Selects all currently active applications of the scanner	String	13	All	SetActiveApplications	43 4C 41 70 70 6C 69 63 61 74 69 6F 6E
Array lenght				All	0..1	00...01
Identifier	Application	String			FEVL (Field Application) RANG (Ranging)	46 45 56 4C 52 41 4E 47
Active		Bool			False = 0 True = 1	False = 00 True = 01

Table 60: Telegram structure: sWN SetActiveApplications

Example: Activate the field application: sWN CLApplication 11

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

Table 61: Example: Activate the field application:: sWN SetActiveApplications 1 FEVL 1

Telegram structure: sWA SetActiveApplications						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command	Selects all currently active applications of the scanner	String		All	SetActiveApplications	53 65 74 41 63 74 69 76 65 41 70 70 6C 69 63 61 74 69 6F 6E 73

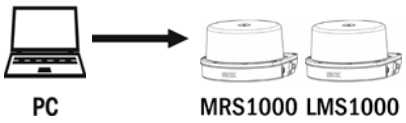
Table 62: Telegram structure: sWA SetActiveApplications

Example: sWA CLApplication correct and accepted

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

Table 63: Example: sWA SetActiveApplications correct and accepted

4.2.12 Read Application selection and switching



Telegram structure: sRN SetActiveApplications						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sRN	73 52 4E
Command	Info of scan frequency and angular resolution	String	10	All	SetActiveApplications	53 65 74 41 63 74 69 76 65 41 70 70 6C 69 63 61 74 69 6F 6E 73

Table 64: Telegram structure: sRN SetActiveApplications

Example for MRS1000: sRN SetActiveApplications

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

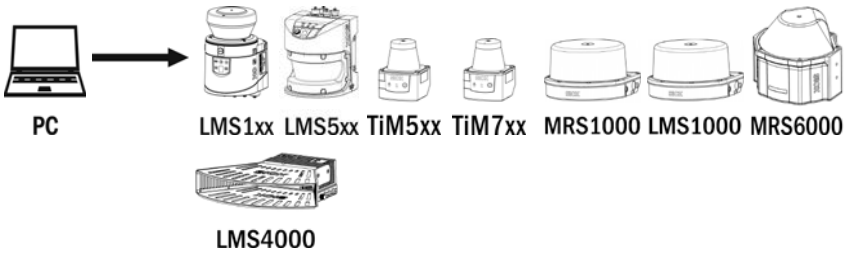
Table 65: Example for MRS1000: sRN SetActiveApplications

Telegram structure: sRA SetActiveApplications						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sRA	73 52 41
Command	Info of scan frequency and angular resolution	String	10	All	SetActiveApplications	73 52 4E 20 53 65 74 41 63 74 69 76 65 41 70 70 6C 69 63 61 74 69 6F 6E 73

4.2.13 Load factory defaults

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.



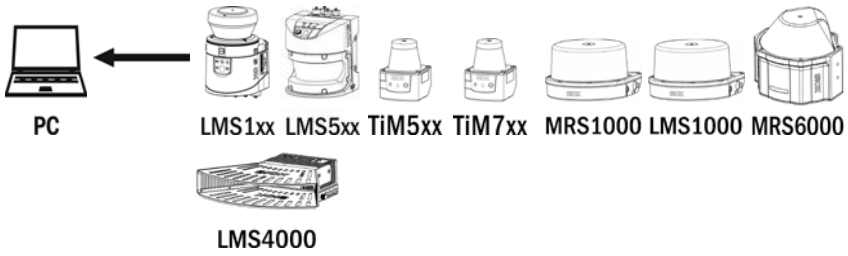
Telegram structure: sMN mSCloadfacdef (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3	All	sMN	Not possible
Command	Load factory defaults	String	13	All	mSCloadfacdef	Not possible

Table 66: Telegram structure: sMN mSCloadfacdef

Example: sMN mSCloadfacdef

CoLa A	ASCII	<STX>sMN{SPC}mSCloadfacdef<ETX>
	Hex	02 73 4D 4E 20 6D 53 43 6C 6F 61 64 66 61 63 64 65 66 03
CoLa B	Binary	Not possible

Table 67: Example: sMN mSCloadfacdef



Telegram structure: sAN mSCloadfacdef						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sAN	Not possible
Command	Load factory defaults	String	13	All	mSCloadfacdef	Not possible

Table 68: Telegram structure: sAN mSCloadfacdef

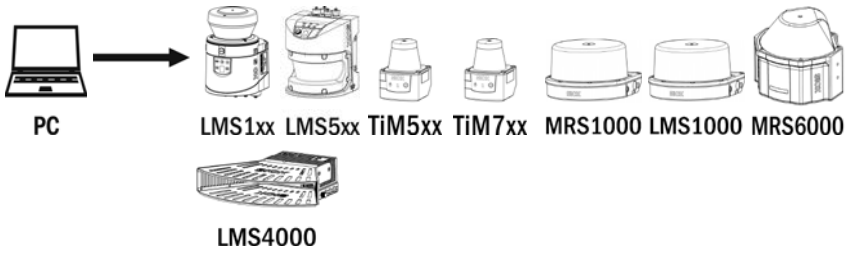
Example: sAN mSCloadfacdef

CoLa A	ASCII	<STX>sAN{SPC}mSCloadfacdef<ETX>
	Hex	02 73 41 4E 20 6D 53 43 6C 6F 61 64 66 61 63 64 65 66 03
CoLa B	Binary	Not possible

Table 69: Example: sAN mSCloadfacdef

4.2.14 Load application defaults

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.



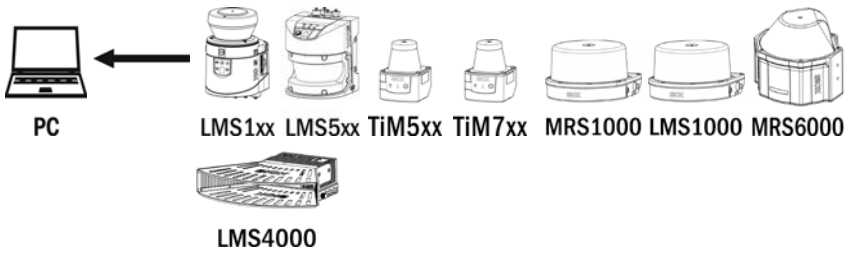
Telegram structure: sMN mSCloadappdef (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3	All	sMN	Not possible
Command	Load application defaults	String	13	All	mSCloadappdef	Not possible

Table 70: Telegram structure: sMN mSCloadappdef

Example: sMN mSCloadappdef

CoLa A	ASCII	<STX>sMN{SPC}mSCloadappdef<ETX>
	Hex	02 73 4D 4E 20 6D 53 43 6C 6F 61 64 61 70 70 64 65 66 03
CoLa B	Binary	Not possible

Table 71: Example: sMN mSCloadappdef



Telegram structure: sAN mSCloadappdef						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sAN	Not possible
Command	Load application defaults	String	13	All	mSCloadappdef	Not possible

Table 72: Telegram structure: sAN mSCloadappdef

Example: sAN mSCloadappdef

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

CoLa B	Binary	Not possible

Table 73: Example: sAN mSCloadappdef

4.2.15 Change password

NOTE

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

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

Table 74: Telegram structure: sMN SetPassword

Example: sMN SetPassword

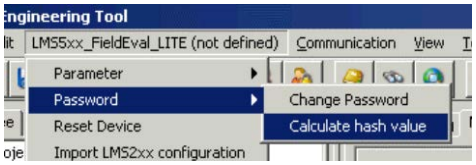
Set password for Authorized user to “testtest”.

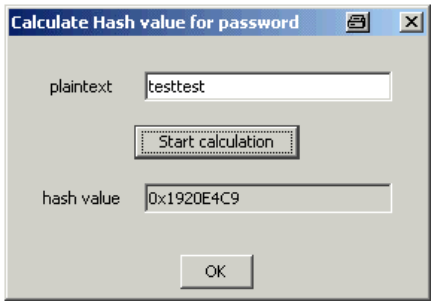
CoLa A	ASCII	<STX>sMN{SPC}SetPassword{SPC}03{SPC}19 20 E4 C9<ETX>
	Hex	02 73 4D 4E 20 53 65 74 50 61 73 73 77 6F 72 64 20 30 33 20 19 20 E4 C9 03
CoLa B	Binary	02 02 02 02 00 00 00 17 73 4D 4E 20 53 65 74 50 61 73 73 77 6F 72 64 20 30 33 20 19 20 E4 C9 3A

Table 75: Example: sMN SetPassword

Calculating the hash value of the password

- Login SOPAS with user level “Service”.
- Select [**Device**] > **Password** > **Calculate Hash value**.





Telegram structure: sAN SetPassword						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sAN	Not possible
Command	Set password requested	String	13	All	SetPassword	53 65 74 50 61 73 73 77 6F 72 64
Success	Confirmation	Int_8	1	All	0: Failed 1: Success	0: Failed 1: Success

Table 76: Telegram structure: sAN SetPassword

Example: sAN SetPassword

CoLa A	ASCII	<STX>sAN{SPC}SetPassword{SPC}1<ETX>
	Hex	02 73 4D 4E 20 53 65 74 50 61 73 73 77 6F 72 64 20 31 03
CoLa B	Binary	02 02 02 02 00 00 00 11 73 41 4E 20 53 65 74 50 61 73 73 77 6F 72 64 20 31 30

Table 77: Example: sAN SetPassword

4.2.16 Check password

Telegram structure: sMN CheckPassword (the same User level or higher)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3	All	sMN	73 4D 4E
Command	Check password request	String	13	All	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	All	Maintenance: 02 Authorized client: 03 Service: 04	Maintenance: 02 Authorized client: 03 Service: 04
Password	Hash value of the password to be checked	Uint_32	4	All	<Hash value>	<Hash value>

Table 78: Telegram structure: sMN CheckPassword

Example: sMN CheckPassword

Check password “testtest” for Authorized user.

CoLa A	ASCII	<STX>sMN{SPC}CheckPassword{SPC}03{SPC}19 20 E4 C9<ETX>
	Hex	02 73 4D 4E 20 43 68 65 63 6B 50 61 73 73 77 6F 72 64 20 30 33 20 19 20 E4 C9 03
CoLa B	Binary	02 02 02 02 00 00 00 19 73 4D 4E 20 43 68 65 63 6B 50 61 73 73 77 6F 72 64 20 30 33 20 19 20 E4 C9 0E

Table 79: Example: sMN CheckPassword

Telegram structure: sAN CheckPassword						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sAN	Not possible
Command	Check password requested	String	13	All	CheckPassword	43 68 65 63 6B 50 61 73 73 77 6F 72 64
Success	Confirmation	Int_8	1	All	0: Failed 1: Success	0: Failed 1: Success

Table 80: Telegram structure: sAN CheckPassword

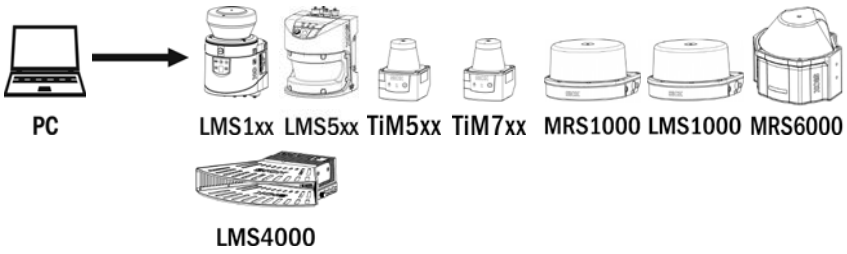
Example: sAN CheckPassword

CoLa A	ASCII	<STX>sAN{SPC}CheckPassword{SPC}1<ETX>
	Hex	02 73 41 4E 20 43 68 65 63 6B 50 61 73 73 77 6F 72 64 20 30 31 03
CoLa B	Binary	02 73 41 4E 20 43 68 65 63 6B 50 61 73 73 77 6F 72 64 20 31 03

Table 81: Example: sAN CheckPassword

4.2.17 Reboot device

This command includes saving all parameters.



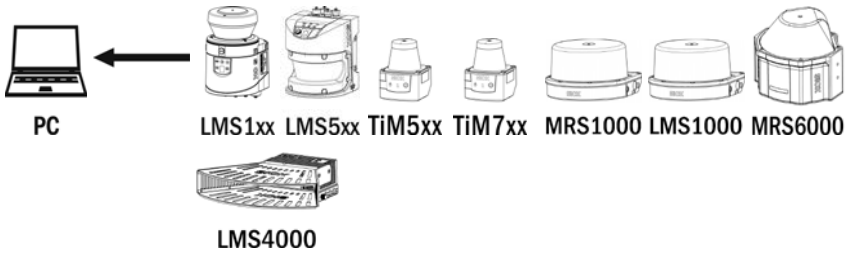
Telegram structure: sMN mSCreboot (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3	All	sMN	73 4D 4E
Command	Reboot device	String	9	All	mSCreboot	6D 53 43 72 65 62 6F 6F 74

Table 82: Telegram structure: sMN mSCreboot

Example: sMN mSCreboot

CoLa A	ASCII	<STX>sMN{SPC}mSCreboot<ETX>
	Hex	02 73 4D 4E 20 6D 53 43 72 65 62 6F 6F 74 03
CoLa B	Binary	02 02 02 02 00 00 00 0D 73 4D 4E 20 6D 53 43 72 65 62 6F 6F 74 2C

Table 83: Example: sMN mSCreboot



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

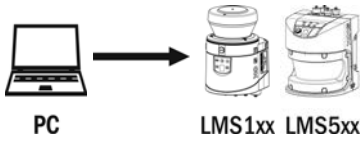
Table 84: Telegram structure: sAN mSCreboot

Example: sAN mSCreboot

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

Table 85: Example: sAN mSCreboot

4.2.18 Set contamination measurement settings



Telegram structure: sWN LCMcfg (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command		String	6	All	LCMcfg	4C 43 4D 63 66 67
Strategy	Strategy code	Enum_8	1	All	Inactive: 0 High available: 1 Available: 2 Sensitive: 3 Semi-sensitive: 4	Inactive: 00 High available: 01 Available: 02 Sensitive: 03 Semi-sensitive: 04
Response time	Time lapse	Uint_32	4	All	+1d ... +60d (01h ... 3Ch)	00 00 00 01 ... 00 00 00 3C
Threshold warning	Threshold value	Uint_32	4	All	0d ... +100d (00h ... 64h)	00 00 00 00 ... 00 00 00 64
Threshold error	Threshold value	Uint_32	4	All	0d ... +100d (00h ... 64h)	00 00 00 00 ... 00 00 00 64

Table 86: Telegram structure: sWN LCMcfg

Example: sWN LCMcfg

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

Table 87: Example: sWN LCMcfg

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

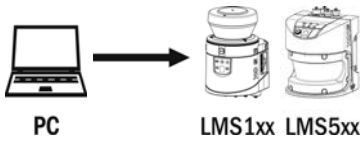
Table 88: Telegram structure: sWA LCMcfg

Example: sWA LCMcfg

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

Table 89: Example: sWA LCMcfg

4.2.19 Read contamination measurement settings



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

Table 90: Telegram structure: sRN LCMcfg

Example: sRN LCMcfg

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

Table 91: Example: sRN LCMcfg

Telegram structure: sRA LCMcfg						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sRA	73 52 41
Command	Read for settings	String	6	All	LCMcfg	4C 43 4D 63 66 67
Strategy	Strategy code	Enum_8	1	All	Inactive: 0 High available: 1 Available: 2 Sensitive: 3 Semi-sensitive: 4	Inactive: 00 High available: 01 Available: 02 Sensitive: 03 Semi-sensitive: 04
Response time	Time lapse	Uint_16	2	All	+1d ... +60d (00h ... 3Ch)	00 00 ... 00 3C
Threshold warning	Threshold value	Uint_16	2	All	0d ... +100d (00h ... 64h)	00 00 ... 00 64
Threshold error	Threshold value	Uint_16	2	All	0d ... +100d (00h ... 64h)	00 00 ... 00 64

Table 92: Telegram structure: sRA LCMcfg

Example: sRA LCMcfg

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

Table 93: Example: sRA LCMcfg

4.2.20 Read contamination measurement detailed values

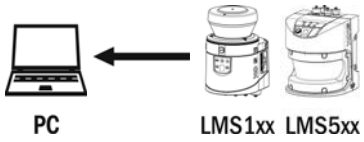
Telegram structure: sRN CMContLvlM						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sRN	73 52 4E
Command		String	10	All	CMContLvlM	43 4D 43 6F 6E 74 4C 76 6C 4D

Table 94: Telegram structure: sRN CMContLvlM

Example: sRN CMContLvlM

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

Table 95: Example: sRN CMContLvlM



Telegram structure: sRA CMContLvlM						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sRA	73 52 41
Command		String	10	All	CMContLvlM	43 4D 43 6F 6E 74 4C 76 6C 4D
Contamination data for different channels	[% of transparency] in order of the different channels	Uint_8	1	LMS1xx	Order of 7 channels: -25.8°/12.8°/51.4°/90°/128.6°/167.2°/205.8° 0d ... +100d (00h ... 64h)	Order of 7 channels: -25.8°/12.8°/51.4°/90°/128.6°/167.2°/205.8° 00 ... 64
				LMS5xx NAV310 LD-OEM 15xx LD-LRS 36xx	Order of 6 channels: 5°/35°/70°/110°/145°/175° 0d ... +100d (00h ... 64h)	Order of 6 channels: 5°/35°/70°/110°/145°/175° 00 ... 64

Table 96: Telegram structure: sRA CMContLvlM

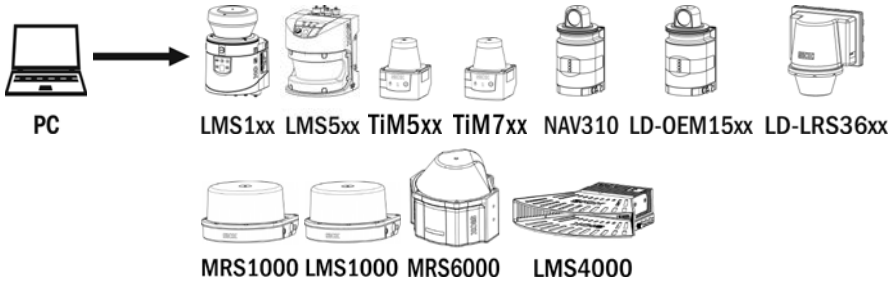
Example for LMS5xx: sRA CMContLvlM

5°-to 110°-channel: 100%, 145°-and 175°-channel only 84% availability:

CoLa A	ASCII	<STX>sRA{SPC}CMContLvlM{SPC}64{SPC}64{SPC}64{SPC}64{SPC}54{SPC}54{SPC}<ETX>
	Hex	02 73 52 41 20 43 4D 43 6F 6E 74 4C 76 6C 4D 20 64 64 64 64 54 54 03
CoLa B	Binary	02 02 02 02 00 00 00 15 73 52 41 20 43 4D 43 6F 6E 74 4C 76 6C 4D 20 64 64 64 64 54 54 43

Table 97: Example for LMS5xx: sRA CMContLvlM

4.2.21 Save parameters permanently



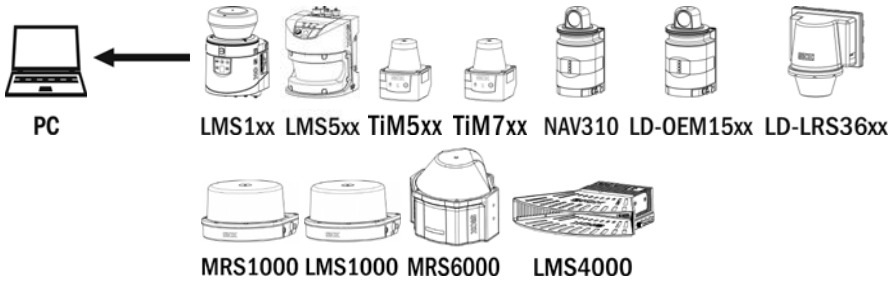
Telegram structure: sMN mEEwriteall (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3	All	sMN	73 4D 4E
Command	Store parameters permanently	String	11	All	mEEwriteall	6D 45 45 77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

Table 98: Telegram structure: sMN mEEwriteall

Example: sMN mEEwriteall

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

Table 99: Example: sMN mEEwriteall



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

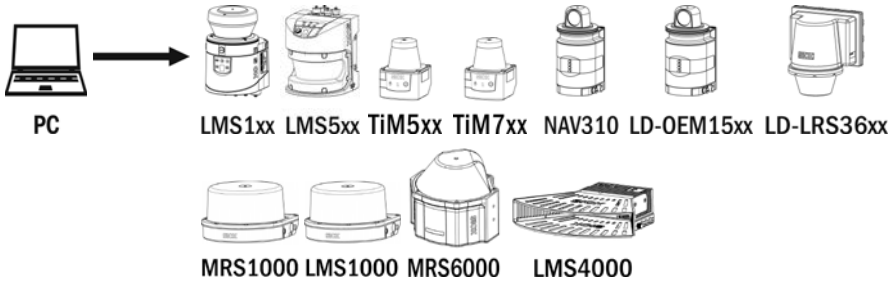
Table 100: Telegram structure: sAN mEEwriteall

Example: sAN mEEwriteall

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

Table 101: Example: sAN mEEwriteall

4.2.22 Set to run



Log out from device and activate all parameter changes.

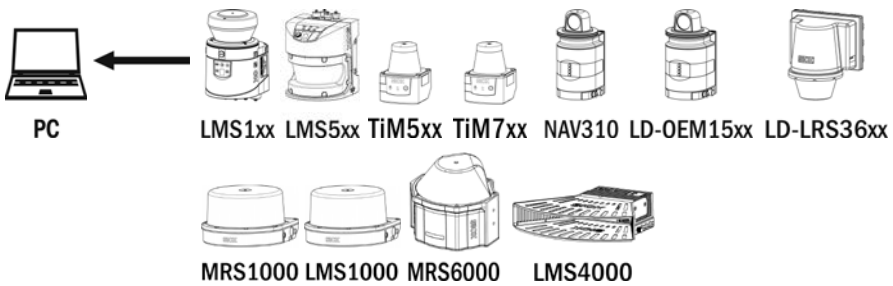
Telegram structure: sMN Run						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3	All	sMN	73 4D 4E
Command	Start the device	String	3	All	Run	52 75 6E

Table 102: Telegram structure: sMN Run

Example: sMN Run

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

Table 103: Example: sMN Run



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

Table 104: Telegram structure: sAN Run

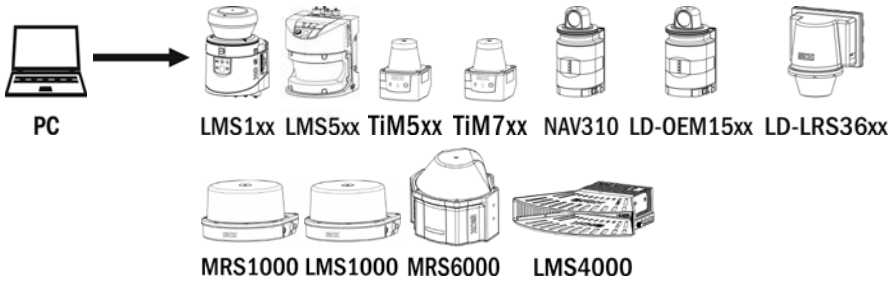
Example: sAN Run

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

Table 105: Example: sAN Run

4.3 Measurement output telegram

4.3.1 Configure the data content for the scan



Telegram structure: sWN LMDscandatacfg (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Configure scandata	String	14	All	LMDscandatacfg	4C 4D 44 73 63 61 6E 64 61 74 61 63 66 67
Data channel	Defines the telegram content (DIST)	Uint_8	2	LMS1xx	Output channel 1: 1 0 Output channel 2: 2 0 Output channel 1+2: 3 0	01 00 02 00 03 00
				LMS5xx	Set via Echo Filter, therefore: 0	00
				TiMxxx NAV310 LD-OEM 15xx LD-LRS 36xx	Output channel 1: 1 0	01 00
				MRS1000 LMS1000	Output channel 1: 1 0 Output channel 2: 2 0 Output channel 1+2: 3 0 Output channel 1+2+3: 7 0	01 00 02 00 03 00 07 00

Telegram structure: sWN LMDscandatacfg (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
				MRS 6000	Output channel 1: 01 00 Output channel 2: 02 00 Output channel 3: 04 00 Output channel 4: 08 00 Veritcle-angle: 10 00 Channel 1+2+3+4+Verticle angle: 1F 00	01 00 02 00 04 00 08 00 10 00 1F 00
				LMS 4000	No distance values: 0 0 Distance values: 1 0	00 00 01 00
Remission & Angle	Remission & Angle offset data output	Uint_8	1	All	No values: 0 RSSI: 1	00 01
				MRS1000 LMS	No values: 0 RSSI: 1 AINF: 8 RSSI & AINF: 9	00 01 08 09
				LMS4000	No values: 0 Remission only: 1 Angle only: 2 Remission & Angle: 3	00 01 02 03
Resolution	Resolution of remission data [1]	Enum_8	1	All	8 Bit: 0 16 Bit: 1	00 01
				LMS4000	Always: 1	01
Unit	Unit of remission data	Enum_8	1	All	Digits: 0	00
				LMS4000	Digits (RSSI): 0 Percent (REFL): 1	00 01
Encoder	Encoder data	Uint_8	2	LMS1xx LMS5xx LMS4000	No encoder: 0 0 Channel 1: 1 0	00 00 01 00
				NAV310 LD-OEM 15xx LD-LRS 36xx TiMxxx MRS1000 LMS1000 MRS6000	No encoder: 0 0	No encoder: 00 00
Position	Position values	Bool_1	1	All	No: 0 Yes: 1	00 01

[1] LMS5xx since V1.10, 8 bit only. ; MRS1000/LMS1000 8bit only

Telegram structure: sWN LMDscandatacfg (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Device name	Sends the device name	Bool_1	1	All	No: 0 Yes: 1	00 01
Comment	Saved comment	Bool_1	1	All	No: 0 Yes: 1	00 01
Time	Sends time information	Bool_1	1	All	No: 0 Yes: 1	00 01
Output rate	Sends the output rate	Uint_16	2	LMS1xx LMS5xx TiMxxx LMS4000	All scans: +1d (1h) Each 2nd scan: +2d (2h) Each 50000th scan: +50000d (C350h)	00 01 00 02 C3 50
				MRS1000 LMS1000	All scans: +1d (1h)	00 01
				MRS6000	All scans: +1d (1h) Each 2nd scan: +2d (2h) ... Max: Each 100th Scan: +100d (64h)	00 01 00 02 ... 00 64
				NAV310 LD-OEM 15xx LD-LRS 36xx	All scans: +1d (1h) Each 2nd scan: +2d (2h) Each 200th scan: +200d (C8h)	00 01 00 02 00 C8

Table 106: Telegram structure: sWN LMDscandatacfg

Example 1: output channel 1, no encoder and all scans

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

Table 107: Example 1: sWN LMDscandatacfg

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

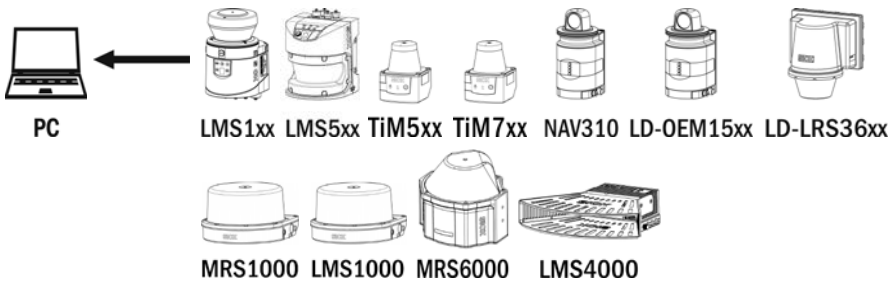
CoLa A	ASCII	<STX>sWN{SPC}LMDscandatacfg{SPC}01{SPC}00{SPC}1{SPC}1{SPC}0{SPC}00{SPC}00{SPC}0{SPC}0 {SPC}0{SPC}0{SPC}+10<ETX>
	Hex	02 73 57 4E 20 4C 4D 44 73 63 61 6E 64 61 74 61 63 66 67 20 30 31 20 30 30 20 30 20 31 20 30 20 30 30 20 30 30 20 30 20 30 20 30 20 30 20 2B 31 30 03
CoLa B	Binary	02 02 02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

Table 108: Example 2: sWN LMDscandatacfg

Example 3: output channel 2, encoder active, each 10th scan

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

Table 109: Example3: sWN LMDscandatacfg



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

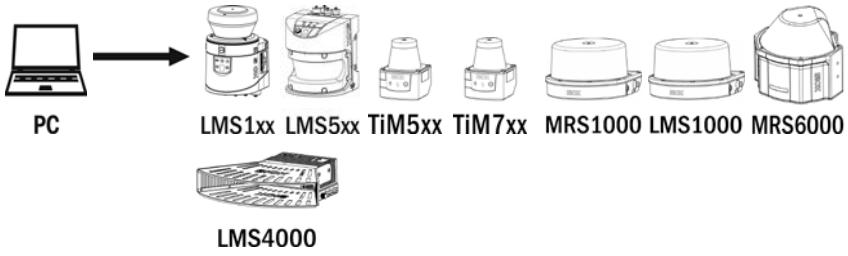
Table 110: Telegram structure: sWA LMDscandatacfg

Example: sWA LMDscandatacfg

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

Table 111: Example: sWA LMDscandatacfg

4.3.2 Configure measurement angle of the scandata for output



Telegram structure: sWN LMPoutputRange (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Change output angle range	String	14	All	LMPoutputRange	4C 4D 50 6F 75 74 70 75 74 52 61 6E 67 65
Status code	Length	Int_16	2	All	1	00 01
Angular resolu-tion [2]	[1/10000°]	Uint_32	4	LMS1xx	0.25°: +2500d (9C4h) 0.5°: +5000d (1388h)	0.25°: 00 00 09 C4 0.5°: 00 00 13 88
				LMS5xx	0.1667°: +1667d (683h) 0.25°: +2500d (9C4h) 0.333°: +3333d (D05h) 0.5°: +5000d (1388h) 0.667°: +6667d (1A0Bh) 1°: +10000d (2710h)	0.1667°: 00 00 06 83 0.25°: 00 00 09 C4 0.333°: 00 00 0D 05 0.5°: 00 00 13 88 0.667°: 00 00 1A 0B 1°: 00 00 27 10
				TiMxxx	0.333°: +3333d (D05h) 1°: +10000d (2710h)	0.333°: 00 00 0D 05 1°: 00 00 27 10
				MRS1000	0.25°: +2500d (9C4h)	0.25°: 00 00 09 C4
				LMS1000	0.75°: +7500d (1D4Ch)	
				MRS 6000	0.13°: +1300d (514h)	0.13°: 00 00 05 14
				LMS 4000	0.0833°: +833 (341h)	0.0833°: 00 00 03 41
Start angle	[1/10000°]	Int_32	4	LMS1xx TiMxxx	-450000d ... +2250000d (FFF92230h ... 225510h)	FF F9 22 30 ... 00 22 55 10
				LMS5xx	-50000d ... +1850000d (FFFF3CB0h ... 1C3A90h)	FF FF 3C B0 ... 00 1C 3A 90
				MRS1000	-475000d ... +2275000d (FFF8C088h ... 22B6B8h)	FF F8 C0 88 ... 00 22 B6 B8
				LMS1000	-480000d (FFF8AD00h)	
				MRS 6000	30000d...1500000d	00 04 93 E0 ...

[2] Note: Angular resolution can not be changed here, it is taken automatically from the basic scan settings! The angular resolution is not exactly 0.1667 degree, and this value should not be used for calculations. The result is an angular resolution of $0,1\overline{6}$ or 1/6 of a degree (six measurements per degree). When used for calculations a customer should recover the real value, e.g. by double AngRes = 2.0 / round(2.0 / GivenAngRes).

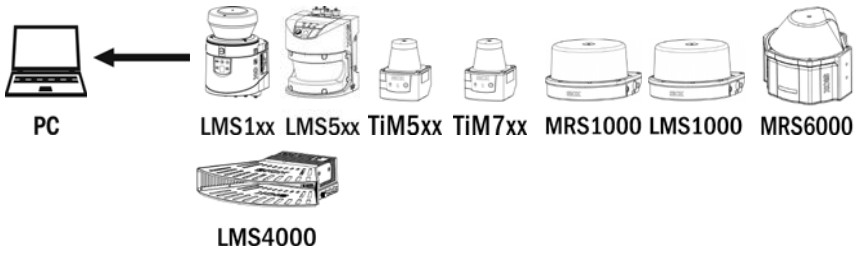
Telegram structure: sWN LMPoutputRange (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
					(493E0h ... 16E360h)	00 16 E3 60
				LMS 4000	+550000d... +1250000d (86470h ... 1312D0h)	00 08 64 70 ... 00 13 12 D0
Stop angle	[1/10000°]	Int_32	4	LMS1xx TiMxxx	-450000d ... +2250000d (FFF92230h ... 225510h)	FF F9 22 30 ... 00 22 55 10
				LMS5xx	-50000d ... +1850000d (FFFF3CB0h ... 1C3A90h)	FF FF 3C B0 ... 00 1C 3A 90
				MRS1000	-475000d ... +2275000d (FFF8C088h ... 22B6B8h)	FF F8 C0 88 ... 00 22 B6 B8
				LMS1000	+2280000d (22CA40h)	
				MRS 6000	30000d...1500000d (493E0h ... 16E360h)	00 04 93 E0h ... 00 16 E3 60h
				LMS 4000	+550000d... +1250000d (86470h ... 1312D0h)	00 08 64 70 ... 00 13 12 D0

Table 112: Telegram structure: sWN LMPoutputRange

Example: sWN LMPoutputRange 0,50°resolution, 0°-90°

CoLa A	ASCII	<STX>sWN{SPC}LMPoutputRange{SPC}1{SPC}1388{SPC}0{SPC}DBBA0<ETX>
	Hex	02 73 57 4E 20 4C 4D 50 6F 75 74 70 75 74 52 61 6E 67 65 20 31 20 31 33 38 38 20 30 20 44 42 42 41 30 03
CoLa B	Binary	02 02 02 02 00 00 00 21 73 57 4E 20 4C 4D 50 6F 75 74 70 75 74 52 61 6E 67 65 20 00 01 00 00 13 88 00 00 00 00 00 0D BB A0 F7

Table 113: Example: sWN LMPoutputRange 0,50°resolution, 0°-90°



Telegram structure: sWA LMPoutputRange						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command	Store parameters	String	14	All	LMPoutputRange	4C 4D 50 6F 75 74 70 75 74 52 61 6E 67 65

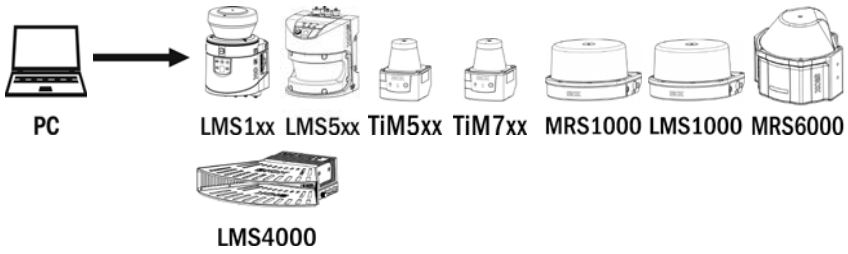
Table 114: Telegram structure: sWA LMPoutputRange

Example: sWA LMPoutputRange

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

Table 115: Example: sWA LMPoutputRange

4.3.3 Read for actual output range



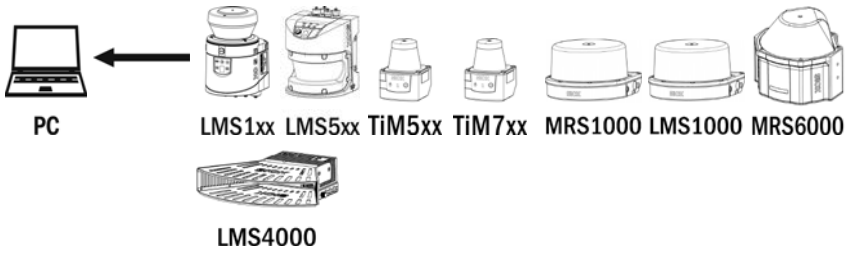
Telegram structure: sRN LMPoutputRange						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sRN	73 52 4E
Command	Output range	String	14	All	LMPoutputRange	4C 4D 50 6F 75 74 70 75 74 52 61 6E 67 65

Table 116: Telegram structure: sRN LMPoutputRange

Example: sRN LMPoutputRange

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

Table 117: Example: sRN LMPoutputRange



Telegram structure: sRA LMPoutputRange						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sRA	73 52 41
Command	Output range	String	14	All	LMPoutputRange	4C 4D 50 6F 75 74 70 75 74 52 61 6E 67 65
Number of sectors	Indicates the number of sectors. The subsequent values will be transmitted 1 ... 4 accordingly.	Int_16	2	All	Sector 1: 0001h	Sector 1: 0001
Angular resolution	[1/10000°]	Uint_32	4	LMS1xx	0.25°: +2500d (9C4h) 0.5°: +5000d (1388h)	0.25°: 00 00 09 C4 0.5°: 00 00 13 88
				LMS5xx	0.1667°: +1667d (683h) 0.25°: +2500d (9C4h) 0.333°: +3333d (D05h) 0.5°: +5000d (1388h) 0.667°: +6667d (1A0Bh) 1°: +10000d (2710h)	0.1667°: 00 00 06 83 0.25°: 00 00 09 C4 0.333°: 00 00 0D 05 0.5°: 00 00 13 88 0.667°: 00 00 1A 0B 1°: 00 00 27 10
				TiMxxx	0.333°: +3333d (D05h) 1°: +10000d (2710h)	0.333°: 00 00 0D 05 1°: 00 00 27 10
				MRS1000	0.25°: +2500d (9C4h)	0.25°: 00 00 09 C4
				LMS1000	0.75°: +7500d (1D4Ch)	
				MRS 6000	0.13°: +1300d (514h)	0.13°: 00 00 05 14
				LMS 4000	0.0833°: +833 (341h)	0.0833°: 00 00 03 41
Start angle	[1/10000°]	Int_32	4	LMS1xx TiMxxx	-450000d ... +2250000d (FFF92230h ... 225510h)	FF F9 22 30 ... 00 22 55 10
				LMS5xx	-50000d ... +1850000d (FFFF3CB0h ... 1C3A90h)	FF FF 3C B0 ... 00 1C 3A 90
				MRS1000	-475000d ... +2275000d (FFF8C088h ... 22B6B8h)	FF F8 C0 88 ... 00 22 B6 B8
				LMS1000	-480000d ... +2280000d (FFF8AD00h ... 22CA40h)	
				MRS 6000	30000d...1500000d (493E0h ... 16E360h)	00 04 93 E0h ... 00 16 E3 60h
				LMS 4000	+550000d... +1250000d (86470h ... 1312D0h)	00 08 64 70 ... 00 13 12 D0

Telegram structure: sRA LMPoutputRange						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Stop angle	[1/10000°]	Int_32	4	LMS1xx TiMxxx	-450000d ... +2250000d (FFF92230h ... 225510h)	FF F9 22 30 ... 00 22 55 10
				LMS5xx	-50000d ... +1850000d (FFFF3CB0h ... 1C3A90h)	FF FF 3C B0 ... 00 1C 3A 90
				MRS1000	-475000d ... +2275000d (FFF8C088h ... 22B6B8h)	FF F8 C0 88 ... 00 22 B6 B8
				LMS1000	-480000d ... +2280000d (FFF8AD00h ... 22CA40h)	
				MRS 6000	30000d...1500000d (493E0h ... 16E360h)	00 04 93 E0h ... 00 16 E3 60h
				LMS 4000	+550000d... +1250000d (86470h ... 1312D0h)	00 08 64 70 ... 00 13 12 D0

Table 118: Telegram structure: sRA LMPoutputRange

Example: sRA LMPoutputRange

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

Table 119: Example: sRA LMPoutputRange

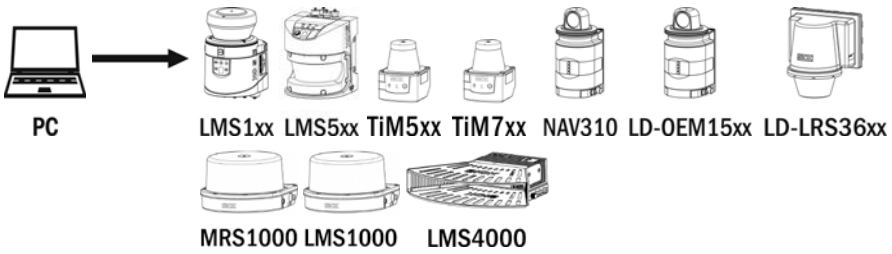
4.3.4 Poll one telegram

Output of values from last scan.

Asking the device for the measurement values of the last valid scan. The device will respond, even if it is not running at the moment.

NOTE

After changing the scanning frequency, there will be no data telegram or answer from the devices LMS1xx, LMS5xx and TiMxxx for up to 30 seconds. The same applies when the device is powering up or rebooting.



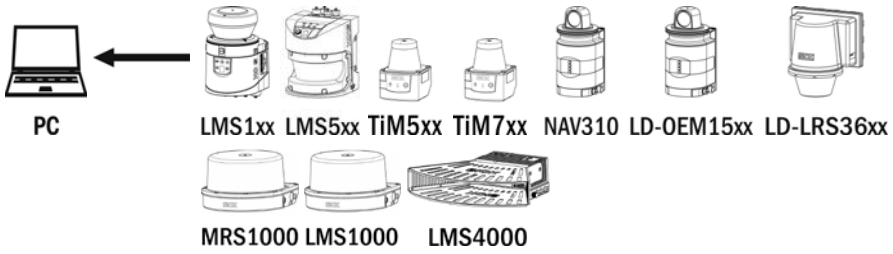
Telegram structure: sRN LMDscandata						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sRN	73 52 4E
Command	Only one telegram	String	11	All	LMDscandata	4C 4D 44 73 63 61 6E 64 61 74 61

Table 120: Telegram structure: sRN LMDscandata

Example: sRN LMDscandata

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

Table 121: Example: sRN LMDscandata



Telegram structure: sRA LMDscandata						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Find complete telegram structure of the answer in section 4.3.5 „Send data permanent“ on page 69.						

Table 122: Telegram structure: sRA LMDscandata

Example: sRA LMDscandata

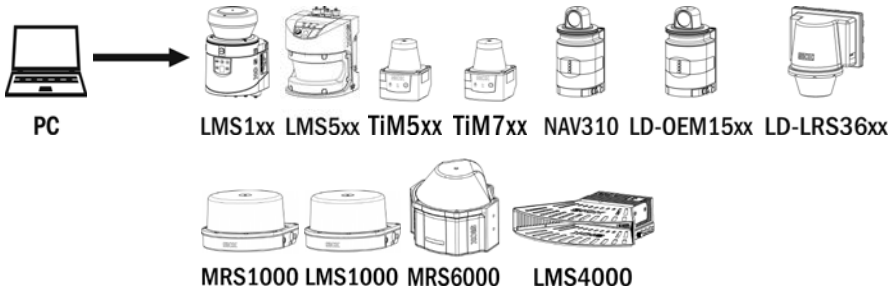
CoLa A	ASCII	No ASCII answer possible.
	Hex	02 73 52 41 20 4C 4D 44 73 63 61 6E 64 61 74 61 20 01 20 01 20 89 C9 97 20 00 20 00 20 1A AE 1A B1 20 58 1C BC 15 20 58 1D 15 3D 20 00 20 00 20 07 20 00 20 00 20 13 88 20 15 20 F6 20 F9 20 F5 20 EF 20 F6 20 F2 20 EF 20 ED 20 F5 20 E9 20 F2 20 FA 20 FC 20 FF 20 F1 20 F2 20 01 07 20 FC 20 FC 20 01 02 20 FF 20 00 20 00 20 00 20 00 20 00 20 00 03
CoLa B	Binary	Find complete telegram structure of the answer in section 4.3.5 „Send data permanent“ on page 69.

Table 123: Example: sRA LMDscandata

4.3.5 Send data permanently

NOTE

After changing the scanning frequency, there will be no data telegram or answer from the devices LMS1xx, LMS5xx and TiMxxx for up to 30 seconds. The same applies when the device is powering up or rebooting.



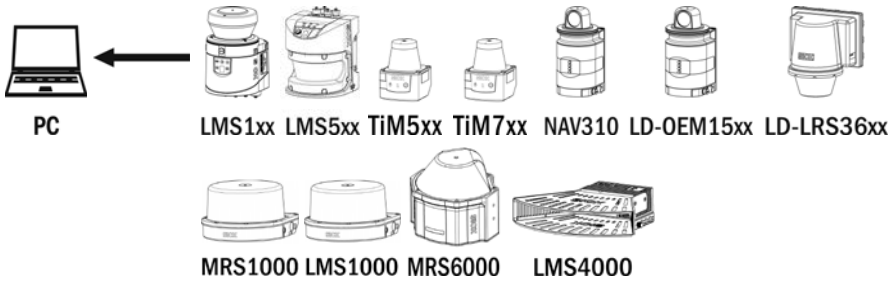
Telegram structure: sEN LMDscandata						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Event	String	3	All	sEN	73 45 4E
Command	Data telegram	String	11	All	LMDscandata	4C 4D 44 73 63 61 6E 64 61 74 61
Measure-ment	Start/stop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

Table 124: Telegram structure: sEN LMDscandata

Example: sEN LMDscandata

CoLa A	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
CoLa B	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

Table 125: Example: sEN LMDscandata



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

Table 126: Telegram structure: sEA LMDscandata

Example: Confirmation of sEA LMDscandata

CoLa A	ASCII	<STX>sEA{SPC}LMDscandata{SPC}1<ETX>
	Hex	02 73 45 41 20 4C 4D 44 73 63 61 6E 64 61 74 61 20 31 03
CoLa B	Binary	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 33

Table 127: Example: Confirmation of sEA LMDscandata

Telegram stream

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

NOTE

Leading zeros of a value will not be displayed in ASCII.

Telegram structure: sRA LMDscandata/sSN LMDscandata							
Telegram part		**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type		Read	String	3	All	sRA sSN	73 52 41 73 53 4E
Command		Data telegram	String	11	All	LMDscandata	4C 4D 44 73 63 61 6E 64 61 74 61
Version number		For detecting format changes by the version. Version is always 1 up to now.	Uint_16	2	All	0000h ... FFFFh	00 00 ... FF FF
Device	Device number	Defined with SOPAS	Uint_16	2	All	0000h ... FFFFh	00 00 ... FF FF

Telegram structure: sRA LMDscandata/sSN LMDscandata							
Telegram part		Description	Variable	Length	Sensor	Values CoLa A (ASCII)	Values CoLa B (Binary)
	Serial number	Defined in factory	Uint_32	4	All	00000000h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
	Device status	(See values column)	Uint_8	2 × 1	All	Ok: 00 00 Error: 00 01	00 00 00 01
					LMS1xx LMS5xx	Pollution warning: 00 02 Pollution error: 00 05	00 02 00 05
Status info	Telegram counter	Number of measurement telegrams finished in the scanner and given to the interface. [3]	Uint_16	2	All	0000h ... 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	All	0000h ... 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	All	00000000h ... 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	All	00000000h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
	Status of digital inputs	Low byte represents input 1.	Uint_8	2 × 1	LMS1xx LMS5xx	All inputs low: 00 00 All inputs high: 00 03	00 00 00 03
					MRS 1000 LMS 1000 MRS 6000 LMS 4000	00 00	00 00

[3] Does not count how many telegrams were really given out; is relevant if not all scans are delivered from the scan core.

Telegram structure: sRA LMDscandata/sSN LMDscandata							
Telegram part		Description	Variable	Length	Sensor	Values CoLa A (ASCII)	Values CoLa B (Binary)
	Status of digital outputs	Low byte represents output 1.	Uint_8	2 × 1	All	All outputs low: 00 00 TiMxxx: • All internal outputs high: 00 0F LMS1xx: • All internal outputs high: 00 07 •All outputs high (inkl. Ext. Out): 07 FF LMS5xx: • All internal outputs high: 00 3F • All outputs high (inkl. Ext. Out): 3F FF LDXXX •All outputs high: 00 0F MRS1000 (1.0.0) LMS1000 MRS6000 (1.0.1) LMS4000 •Always 00 00	All outputs low: 00 00 TiMxxx: • All internal outputs high: 00 0F LMS1xx: • All internal outputs high: 00 07 •All outputs high (inkl. Ext. Out): 07 FF LMS5xx: • All internal outputs high: 00 3F • All outputs high (inkl. Ext. Out): 3F FF LDXXX •All outputs high: 00 0F MRS1000 (1.0.0) LMS1000 MRS6000 (1.0.1) LMS4000 •Always 00 00
	former Reserved now Layer angle.	–	Uint_16	2	All except MRS 1000	0 0→ 0 Layer2 FF06→ -250 Layer3 FA→ 250 Layer1 FE0C → -500 Layer4 (value 1/100)	0 00 00 00 00 46 46 30 36 00 00 46 41 46 45 30 43
			Int_16		MRS6000	Angle = value / 200 Example: F5B2h →-2638/200= -13.19° →Layer 24 EDh→237/200= 1.185°→Layer 1 Range: -13.19°~1.185° (each layer is 0.625°)	Angle = value / 200 Example: F5 B2→Layer 24 00 ED→Layer 1
Frequencie	Scan frequency	[1/100 Hz]	Uint_32	4	LMS1xx	25 Hz: +2500d (9C4h) 50 Hz: +5000d (1388h)	09 C4 13 88

Telegram structure: sRA LMDscandata/sSN LMDscandata							
Telegram part		**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
					LMS5xx	25 Hz: +2500d (9C4h) 35 Hz: +3500d (DACh) 50 Hz: +5000d (1388h) 75 Hz: +7500d (1A0Bh) 100 Hz: +10000d (2710h)	09 C4 0D AC 13 88 1A 0B 27 10
					TiMxxx	15 Hz: +1500d (5DCh)	05 DC
					NAV310 LD-OEM 15xx	5 Hz ... 20 Hz: +500d ... +2000d (1F4h ... 7D0h)	01 F4 ... 07 D0
					LD-LRS 36xx	5 Hz ... 15 Hz: +500d ... +1500 (1F4h ... 5DCh))	01 F4 ... 05 DC
					MRS 1000	50 Hz: +5000d (1388h)	50 Hz: 13 88
					LMS 1000	150 Hz: +15000d (3A98h)	
					MRS 6000	10 Hz: +1000d (3E8h)	10 Hz: 03 E8
					LMS 4000	600 Hz: +60000d (EA60h)	600 Hz: EA 60
	Measure-ment fre-quency	Inverse of the time between two measurement shots (in 100 Hz), Example: 50 Hz, 0.5° resolution ➜ 720 shots/20 ms ➜ 36 kHz	Uint_32	4	All	00000000h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
					LMS 4000	000021C0h	00 00 21 C0
Amount of encoder		If 0, then next two values are missing.	Enum_16	2	TiMxxx MRS 1000 LMS 1000 MRS 6000	Always: 0	Always: 00 00
					LMS1xx LMS5xx	0 ... 3	00 00 ... 00 03
					LMS 4000	0 ... 1	00 00 ... 00 01
Values	Encoder position	Info in ticks	Uint_32	4	LMS1xx LMS5xx LMS 4000	00000000h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
	Encoder speed	Ticks/mm	Uint_16	2	LMS1xx LMS5xx	0000h ... FFFFh	00 00 ... FF FF
					LMS 4000	Always: 0	Always: 00 00
Amount of		Number of 16 bit	Uint_16	2	TiMxxx	Output channel: 1	Output channel: 01

Telegram structure: sRA LMDscandata/sSN LMDscandata							
Telegram part		**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
16 bit channels		channels that provide measured data			LMS1xx	Output channels: 1, 2 or 4	Output channels: 01, 02 or 04
					LMS5xx	Output channels: 1 or 5	Output channels: 01 or 05
					MRS 1000 LMS 1000	Output channels: 1 or 3	Output channels: 01 or 03
					MRS 6000	Output channels:1..9	Output channels:01..09
					LMS 4000	Output channels: 0...3	Output channels: 00...03
					NAV310 LD-OEM 15xx LD-LRS 36xx	Depending on amount of sectors and selection of output of distance or distance and remission RSSI Example (2 sectors): If 2 channels: sectors 1 + 2 contain Dist1 If 4 channels: sectors 1 + 2 contain Dist + RSSI1	Depending on amount of sectors and selection of output of distance or distance and remission RSSI Example (2 sectors): If 2 channels: sectors 1 + 2 contain Dist1 If 4 channels: sectors 1 + 2 contain Dist + RSSI1
Output channel (16 bit)	Content	Defines the content of the output channel Unit of radial distance values (DIST) is mm	String	5	LMS1xx	DIST1: Distance values of first pulse DIST2: Distance valus of second pulse RSSI1: Energy values of first pulse RSSI2: Energy values of second pulse	44 49 53 54 31 44 49 53 54 32 52 53 53 49 31 52 53 53 49 32

Telegram structure: sRA LMDscandata/sSN LMDscandata							
Telegram part		Description	Variable	Length	Sensor	Values CoLa A (ASCII)	Values CoLa B (Binary)
					LMS5xx (with Software ≥V1.10 only)	DIST1: Distance values of first pulse	44 49 53 54 31
						DIST2: Distance valus of second pulse	44 49 53 54 32
						DIST3: Distance values of third pulse	44 49 53 54 33
						DIST4: Distance valus of fourth pulse	44 49 53 54 34
						DIST5: Distance values of fifth pulse	44 49 53 54 35
						RSSI1: Energy values of first pulse	52 53 53 49 31
						RSSI2: Energy values of second pulse	52 53 53 49 32
						RSSI3: Energy values of third pulse	52 53 53 49 33
						RSSI4: Energy values of forth pulse	52 53 53 49 34
						RSSI5: Energy values of fifth pulse	52 53 53 49 35
					TiMxxx	DIST1: Distance values	44 49 53 54 31
					MRS 1000 LMS 1000	DIST1: Distance values	44 49 53 54 31
						DIST2: Distance values	44 49 53 54 32
						DIST3: Distance values	44 49 53 54 33
						RSSI1: Energy values	52 53 53 49 31
						RSSI2: Energy values	52 53 53 49 32
						RSSI3: Energy values	52 53 53 49 33
					MRS6000	DIST1: Distance values	44 49 53 54 31
						DIST2: Distance values	44 49 53 54 32
						DIST3: Distance values	44 49 53 54 33
						DIST4: Distance values	44 49 53 54 34
						RSSI1: Energy values	52 53 53 49 31
						RSSI2: Energy values	52 53 53 49 32
						RSSI3: Energy values	52 53 53 49 33
						RSSI4: Energy values	52 53 53 49 34
						VANGL: Vertical Angle	56 41 4E 47 4C
					NAV310 LD-OEM 15xx LD-LRS 36xx	DIST1: Distance values	44 49 53 54 31
						RSSI1: Energy values	52 53 53 49 31

Telegram structure: sRA LMDscandata/sSN LMDscandata							
Telegram part		Description	Variable	Length	Sensor	Values CoLa A (ASCII)	Values CoLa B (Binary)
					LMS 4000	DIST1: Distance values RSSI1: Energy values in digit REFL1: Calibrated energy values in percent ANGL1: Angle Offset values	44 49 53 54 31 52 53 53 49 31 52 45 46 4C 31 41 4E 47 4C 31
	Scale factor	Scale factor or factor of the measurement values (for the LMS5xx this depends on the angular resolution)	Real as float according to IEEE754	4	LMS1xx LMS5xx TiMxxx MRS 1000 LMS 1000	Factor × 1: 3F800000h Factor × 2: 40000000h	3F 80 00 00 40 00 00 00
					MRS 6000	Factor x 12.5: 41480000h Factor × 1: 3F800000h Factor x -0.00025: B983126Fh	41 48 00 00 3F 80 00 00 B9 83 12 6F
					NAV310 LD-OEM 15xx LD-LRS 36xx	Factor × 4: 40800000h	04 08 00 00
					LMS 4000	Factor x 0.1: 3DCCCDDh (DIST1) Factor × 1: 3F800000h (RSSI1) Factor × 1: 3F800000h (REFL1) Factor × 1: 3F800000h (ANGL1)	3D CC CC CD 3F 80 00 00 3F 80 00 00 3F 80 00 00
	Scale factor offset	Sets starting point of measurement	Real as float according to IEEE754	4	LMS1xx LMS5xx TiMxxx MRS 1000 LMS 1000	00000000h	00 00 00 00
					MRS6000	Offset 0 : 00000000 Offset 1.5: 3FC00000 (1.5 offset for VANGL)	00 00 00 00 3F C0 00 00 (1.5 offset for VANGL)
					NAV310 LD-OEM 15xx LD-LRS 36xx	00000000h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF

Telegram structure: sRA LMDscandata/sSN LMDscandata							
Telegram part		Description	Variable	Length	Sensor	Values CoLa A (ASCII)	Values CoLa B (Binary)
					LMS 4000	DIST1: 00000000h RSSI1: 00000000h REFL1: 00000000h ANGL1: C7000000h (-32768)	00 00 00 00 00 00 00 00 00 00 00 00 C7 00 00 00
	Start angle	[1/10000°]	Uint_32	*4*	LMS1xx TiMxxx	-450000d ... +2250000d (FFF92230h ... 225510h)	FF F9 22 30 ... 00 22 55 10
					LMS5xx	-50000d ... +1850000d (FFFF3CB0h ... 1C3A90h)	FF FF 3C B0 ... 00 1C 3A 90
					MRS 1000	-475000d ... +2275000d (FFF8C088h ... 22B6B8h)	FF F8 C0 88 ... 00 22 B6 B8
					LMS 1000	-480000d ... +2280000d (FFF8AD00h ... 22CA40h)	
					MRS 6000	+30000d...+1500000d (493E0h ... 16E360h)	00 04 93 E0h ... 00 16 E3 60h
					NAV310 LD-OEM 15x1 LD-LRS 36x1	0d ... +3600000d (0h ... 36EE80h)	00 00 00 00 ... 00 36 EE 80
					LD-OEM 15x0 LD-LRS 36x0	-900000d ... +2700000d (FFF24460h ... 41EB0h)	FF F2 44 60 ... 00 04 1E B0
					LMS 4000	+550000d... +1250000d (86470h ... 1312D0h)	00 08 64 70 ... 00 13 12 D0
	Size of single angular step	Output format in degree: 1/10000°	Uint_16	2	LMS1xx	+2500d ... +5000d (9C4h ... 1388h)	09 C4 ... 13 88
					LMS5xx	+1667d ... +10000d (683h ... 2710h)	06 83 ... 27 10
					TiMxxx	+333d ... +10000d (D05h ... 2710h)	0D 05 ... 27 10
					MRS 1000	+2500d (9C4h)	09 C4
					LMS 1000	+7500d (1D4Ch)	
					MRS 6000	+1300d (= 0.13°) (514h)	05 14
					NAV310 LD-OEM 15xx LD-LRS 36xx	0.125° ... 1° def. 0.25° +1250d ... +10000d (4E2h ... 2710h) (Default: 09C4h = 0.25°)	04 E2 ... 27 10 (Default: 09 C4)
					LMS 4000	0.0833°: +833 (341h)	0.0833°: 03 41
	Amount of data	Defines the number of items on measured output	Uint_16	2	All	0000h ... FFFFh	00 00 ... FF FF

Telegram structure: sRA LMDscandata/sSN LMDscandata							
Telegram part		**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
	Data_1 Data_n	Data stream starting Data_1 to Data_n [4]	Uint_16	2	LMS100	0000h ... 4E20h	00 00 00 00 ... 00 00 4E 20
					LMS150	0000h ... C350h	00 00 00 00 ... 00 00 C3 50
					LMS5xx	0000h ... FDE8h	00 00 00 00 ... 00 00 FD E8
					TiMxxx	0000h ... 61A8h	00 00 00 00 ... 00 00 61 A8
					MRS 1000 LMS 1000	0000h ... FA00h	00 00 00 00 .. 00 00 FA 00
					MRS 6000	0000h...8CA0h(DIST) 0000h...FFFFh(RSSI) 00EDh...F5B2h(VANGL)	00 00...8C A0 (DIST) 00 00...FF FF(RSSI) 00 ED...F5 B2 (VANGL)
					LMS 4000	0000h...C4Eh (DIST1) 0000h...FFFFh (RSSI1) 0000h...FFFFh (REFL1) 0000h...FFFFh (ANGL1)	00 00 ... 0C 4E 00 00 ... FF FF 00 00 ... FF FF 00 00 ... FF FF
					NAV310 LD-OEM 15xx LD-LRS 36xx	0000h ... 0992h	00 00 00 00 ... 00 00 09 92
For NAV310/LD-OEM15xx/LRS: The array "Output channel 16 bit " has various dimensions "Amount of 16 Bit Channels", depending on the amount of sectors and if RSSI (output of remission values) is selected as on or off: • If RSSI was not selected (by LMDscandatacfg); there are 2 channels with the contents ○ Channel 1: First sector (Test target), content: DIST1 ○ Channel 2: Second sector (Main profile data), content: DIST1 • If RSSI was selected (by LMDscandatacfg); there are 4 channels with the contents ○ Channel 1: First sector (Test target), content: DIST1 ○ Channel 2: First sector (Test target), content: RSSI1 ○ Channel 3: Second sector (Main profile data), content: DIST1 ○ Channel 4: Second sector (Main profile data), content: RSSI1 The number behind DIST and RSSI is the order number of the pulse. As the NAV310/LD-OEM15xx/LD-LRS36xx scanner are working with a single pulse measurement, it is always "1".							
Amount of 8 bit channels		Amount of 8 bit channels, giving out the measured data	Enum_16	2	LMS1xx	Output channels: 1 or 2	Output channels: 01 or 02
					LMS5xx	Output channels: 1 or 5	Output channels: 01 or 05
					MRS 1000 LMS 1000	Output channels: 1 or 3	Output channels: 01 or 03

[4] LMS1xx without limit.

Telegram structure: sRA LMDscandata/sSN LMDscandata							
Telegram part		Description	Variable	Length	Sensor	Values CoLa A (ASCII)	Values CoLa B (Binary)
					TiMxxx NAV310 LD-OEM 15xx LD-LRS 36xx MRS 6000 LMS 4000	Output channels: 0	Output channels: 00
Output channel (8 bit)	Content	Defines the content of the output channel	String	5	LMS1xx	DIST1 DIST2 RSSI1 RSSI2	44 49 53 54 31 44 49 53 54 32 52 53 53 49 31 52 53 53 49 32
				5	LMS5xx (with Software ≥V1.10 only)	DIST1 DIST2 DIST3 DIST4 DIST5 RSSI1 RSSI2 RSSI3 RSSI4 RSSI5	44 49 53 54 31 44 49 53 54 32 44 49 53 54 33 44 49 53 54 34 44 49 53 54 35 52 53 53 49 31 52 53 53 49 32 52 53 53 49 33 52 53 53 49 34 52 53 53 49 35
				5	TiMxxx	DIST1 RSSI1	44 49 53 54 31 52 53 53 49 31
					MRS 1000 LMS 1000	DIST1 DIST2 DIST3 RSSI1 RSSI2 RSSI3 AINF1 (Ambient light)	44 49 53 54 31 44 49 53 54 32 44 49 53 54 33 52 53 53 49 31 52 53 53 49 32 52 53 53 49 33 41 49 4E 46 31
	Scale factor	Scale factor or of the measurement values (in LMS5xx depends on the angular resolution)	Real as float according to IEEE754	4	All	Factor × 1: 3F800000h Factor × 2 (values have to be scaled by factor two): 40000000h	3F 80 00 00 40 00 00 00
	Scale factor offset	Sets starting point of measurement	Real as float according to IEEE754	4	LMS1xx LMS5xx MRS 1000 LMS 1000	00000000h	00 00 00 00

Telegram structure: sRA LMDscandata/sSN LMDscandata							
Telegram part		**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
	Start angle	Output format: 1/10000°	Int_32	4	LMS1xx	-450000d ... +2250000d	FF F9 22 30 ... 00 22 55 10
					LMS5xx	-50000d ... 1850000d	FF FF 3C B0 ... 00 1C 3A 90
					LMS 1000	-480000d ... +2280000d (FFF8AD00h ... 22CA40h)	
					MRS 1000	-475000d ... +2275000d (FFF8C088h ... 22B6B8h)	FF F8 C0 88 ... 00 22 B6 B8
	Size of single angular step	Output format: 1/10000°	Uint_16	2	LMS1xx	+1000d ... +10000d	03 E8 ... 27 10
					LMS5xx	+1667d ... +10000d	06 83 ... 27 10
					LMS 1000	+7500d (1D4Ch)	
					MRS 1000	+2500d (9C4h)	09 C4
	Amount of data	Amount	Uint_16	2	All	0000h ... FFFFh	00 00 ... FF FF
	Data_1 Data_n	Data stream starting Data_1 to Data_n	Uint_8	1	All	00h ... FFh	00 ... FF
					MRS 1000 LMS 1000	DIST & RSSI: 00h ... FFh AINF (Intensity): 0: 00 1: 04 2: 08 3: 0C 4: 10 5: 14 6: 18 7: 1C	00 ... FF 00 04 08 0C 10 14 18 1C
Position		Output of position data	Enum_16	2	All	No position data: 0 Position data: 1	No position data: 00 00 Position data: 00 01
Position information	X position	X-coordinate as float acco. to IEEE754	Real	4	All	0h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
	Y position	Y-coordinate as float acco. to IEEE754	Real	4	All	0h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
	Z position	Z-coordinate as float acco. to IEEE754	Real	4	All	0h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
	X rotation	X rotation in the coordinate system	Real	4	All	0h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
	Y rotation	Y rotation in the coordinate system	Real	4	All	0h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
	Z rotation	Z rotation in the coordinate system	Real	4	All	0h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF

Telegram structure: sRA LMDscandata/sSN LMDscandata							
Telegram part		Description	Variable	Length	Sensor	Values CoLa A (ASCII)	Values CoLa B (Binary)
	Rotations type	Kind of rotation	Enum_8	1	All	No rotation: 0 Pitch: 1 Roll: 2 Free: 3	No rotation: 00 Pitch: 01 Roll: 02 Free: 03
	Transmits the name of device	Device name	Uint_8	1	All	No name: 0 Name: 1	No name: 00 Name: 01
Name		Device name	Uint_16	2	All	No name: 0 Name: 1	No name: 00 00 Name: 00 01
Name information	Length	Length of name	Uint_8	1	All	0h ... Fh	00 ... 0F
	Name	Device name in characters	String	16	All	20h ... 7Ah	20 ... 7A
Comment		Comment	Uint_16	2	All	No comment: 0 Comment: 1	No comment: 00 00 Comment: 00 01
Comment information	Length	Length of comment	Uint_8	1	All	0h ... Fh	00 ... 0F
	Comment	Transmits a comment in characters	String	16	All	20h ... 7Ah	20 ... 7A
Time		Transmits a time stamp	Uint_16	2	All	No time: 0	No time: 00 00
					LMS1xx, LMS5xx, NAV310, LD-OEM 15xx LD-LRS 36xx	Time: 1	Time: 00 01
Time info	Year		Uint_16	2	All	0000h ... 270Fh	00 00 ... 27 0F
	Month	1 to 12	Uint_8	1	All	00h ... 0Ch	00 ... 0C
	Day	Day of month 1 to 31	Uint_8	1	All	00h ... 1Fh	00 ... 1F
	Hour	0 to 23	Uint_8	1	All	00h ... 17h	00 ... 17
	Minute	0 to 59	Uint_8	1	All	00h ... 3Bh	00 ... 3B
	Second	0 to 59	Uint_8	1	All	00h ... 3Bh	00 ... 3B
	Microsecond	0 to 999999	Uint_32	4	All	00000000h ... 000F423Fh	00 00 00 00 ... 00 0F 42 3F
Event info		Display event info	Uint_16	2	All	No info: 0 Transmit info: 1	No info: 00 00 Transmit info: 00 01
Event information	Type	Fast digital input	String	4	All	FDIN	FDIN
	Encoder position	Position of encoder when event happened	Uint_32	4	All	00000000h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF

Telegram structure: sRA LMDscandata/sSN LMDscandata							
Telegram part		Description	Variable	Length	Sensor	Values CoLa A (ASCII)	Values CoLa B (Binary)
	Time of event	Time (µs) of encoder when event happened	Uint_32	4	All	00000000h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
	Angle of event	Angle of encoder when event happened	Int_32	4	All	0 ... 3600000	00 00 00 00 ... 00 36 EE 80

Table 128: Telegram structure: Datastream of sRA LMDscandata/sSN LMDscandata

NOTE

- The grey written parts are not given out by the sensor.
- The event information is not available with the LMS1xx and with the LMS5xx only with firmware V1.20 or higher.
- The order of events within the data structure is “newest” first.

LMDscandata - reserved values

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

DIST	RSSI	Description
0d	0h	no meas value detected; means that in the angle, there was no valid measurement value. Probably the object to measure was out of the range of the or the object was reflecting too less light back (black objects)
1d	FFFFh (16Bit output) FFh (8Bit output)	dazzled, blinded
2d	0h	implausible measurement values
3d	0h	value was set to invalid by a filter (Echo Filter, Particle Filter in old firmware)
4d – 15d	0h	reserved, at the moment not given out, if there occurs a value in that range anyway → perform a Softwareupdate
≥16d	>0h	valid measurement values

Valid for LMS1xx/5xx, TiMxxx

max. measurement value TiM5xx: Dez: 10.000mm --> Hex: 2710
max. measurement value TiM57x: Dez: 25.000mm --> Hex: 61A8
max. measurement value LMS1xx: Dez: 20.000mm --> Hex: 4E20
max. measurement value LMS15x: Dez: 50.000mm --> Hex: C350
max. measurement value LMS5xx: Dez: 65.000mm --> Hex: FDE8
max. measurement value LMS5xx: Dez: 80.000mm --> Hex: 9C40 with scale factor 2 --> 13880

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

Calculation and amount of data for LMS5xx

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

Sizes of values and telegram parts:

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

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

0.5 ° = 2 measurements per degree

0.25 ° = 4 measurements per degree

Always one additional measurement for the last measurement

Number of measurement values =

Number of degrees × measurements per degree + 1

Example for measurement of 56 ° in 0.5 ° resolution (without RSSI data):

56 × 2 + 1 = 113 Measurement values

Amount of Data for this measurement values:

113 × 5 Byte = 565 Byte

Calculation of amount of data per telegram:

Data of one Telegram = Header + Measurements + end of telegram

81 Byte + 113 Measurements + 12 Byte

81 Byte + (113 × 5Byte) + 12 Byte =

658 Byte per Telegram (= 5264 Bit (658 × 8 Bit))

Possible amout for delivery with special Speed:

Number of telegrams per second = Speed ÷ telegram size

Speed Example:

115200 Bit/s = 11520 Byte/s = 11,52 Byte/s

11520 (Byte/s) ÷ 658 Byte = 17,5 Telegrams/s

Telegram size with **0,25 °** resolution:

Degrees: 270 °

Resolution: 0.25°

→Measurement Values = 270 × 4 + 1 = 1081

Data per Telegram =

81 Byte + (1081 × 5 Byte) + 12 Byte = **5498 Byte** (= 43984 Bit)

Telegram size with **0,5 °** resolution:

Degrees: 270 °

Resolution: 0.5 °

→Measurement Values = 270 × 2 + 1 = 541

Data per Telegram =

81 Byte + (541 × 5 Byte) + 12 Byte = **2798 Byte** (= 22384 Bit)

As a result in that configuration a 10 MBit connection will not be enough. With a 100 MBit Hub, 3-4 scanner can be used, with a 1 GBit Hub accordingly more.

Example of a telegram stream

Example: telegram LMS1xx, LMS5xx similar with corresponding values (10 °-20 ° data range)

ASCII

<STX>sRA{SPC}LMDscandata{SPC}1{SPC}1{SPC}89A27F{SPC}0{SPC}0{SPC}343{SPC}347{SPC}27477BA9{SPC}2747813B{SPC}0{SPC}0{SPC}7{SPC}0{SPC}0{SPC}1388{SPC}168{SPC}0{SPC}1{SPC}DIST1{SPC}3F800000{SPC}00000000{SPC}186A0{SPC}1388{SPC}15{SPC}8A1{SPC}8A5{SPC}8AB{SPC}8AC{SPC}8A6{SPC}8AC{SPC}8B6{SPC}8C8{SPC}8C2{SPC}8C9{SPC}8CB{SPC}8C4{SPC}8E4{SPC}8E1{SPC}8EB{SPC}8E0{SPC}8F5{SPC}908{SPC}8FC{SPC}907{SPC}906{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0<ETX>

BINARY

02 02 02 02 00 00 00 83 73 52 41 20 4C 4D 44 73 63 61 6E 64 61 74 61 20 00 01 00 01 00 89 A2 7F 00 00 03 43 03 47 27 47 7B A9 27 47 81 3B 00 00 07 00 00 00 00 00 13 88 00 00 01 68 00 00 00 01 44 49 53 54 31 3F 80 00 00 00 00 00 00 00 01 86 A0 13 88 00 15 08 93 08 95 08 AF 08 B3 08 B0 08 A4 08 B0 08 BF 08 B9 08 BA 08 D0 08 D3 08 CF 08 DE 08 EB 08 E3 08 FE 08 EC 09 03 08 FD 08 FD 00 00 00 00 00 00 00 00 00 00 00 00 2B

Telegram structure: sRA LMDscandata (Example)				
Telegram part	**Variable**	**Length**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Frame/header			02 <STX>	02 02 02 02
Length				00 00 00 83
Command type	String	3	sRA{SPC}	73 52 41 20
Command	String	11	LMDscandata{SPC}	4C 4D 44 73 63 61 6E 64 61 74 61 20
Version number	Uint_16	2	1{SPC}	00 01
Device number	Uint_16	2	1{SPC}	00 01

Telegram structure: sRA LMDscandata (Example)					
Telegram part		Variable	Length	Values CoLa A (ASCII)	Values CoLa B (Binary)
	Serial number	Uint_32	4	89A27F{SPC} Dec: 9020031	00 89 A2 7F
	Device status	Uint_8	2 × 1	0{SPC}0{SPC}	00 00
Status information	Telegram counter	Uint_16	2	343{SPC} Dec: 835	03 43
	Scan counter	Uint_16	2	347{SPC} Dec: 839	03 47
	Time since start up [µs]	Uint_32	4	27477BA9{SPC} Dec: 658996137	27 47 7B A9
	Time of transmission [µs]	Uint_32	4	2747813B{SPC} Dec: 568997563	27 47 81 3B
	Status of digital inputs	Uint_8	2 × 1	0{SPC}0{SPC}	00 00
	Status of digital outputs	Uint_8	2 × 1	7{SPC}0{SPC} Corresponds to status 0111	07 00
	Reserved	Uint_16	2	0{SPC}	00 00
Frequencies	Scan frequency	Uint_32	4	1388{SPC} Dec: 50 Hz: 5000	00 00 13 88
	Measurement frequency	Uint_32	4	168{SPC}	00 00 01 68
Amount of encoder		Enum_16	2	0{SPC} No encoder data	00 00
Position information	Encoder position	Uint_16	2	Not generated, not existing because amount is 0	Not generated, not existing because amount is 0
	Encoder speed	Uint_16	2	Not generated, not existing because amount is 0	Not generated, not existing because amount is 0
Amount of 16 bit channels		Enum_16	2	1{SPC}	00 01
Output channel (16 bit)	Content	String	5	DIST1{SPC}	44 49 53 54 31
	Scale factor according to IEEE754	Real	4	3F800000{SPC} Floating Point: Value = 1	3F 80 00 00
	Scale factor offset acco. to IEEE754	Real	4	0{SPC} Floating Point: Value = 0	00 00 00 00
	Start angle	Int_32	4	186A0{SPC} Dec: 100000	00 01 86 A0
	Size of single angular step	Uint_16	2	1388{SPC} Dec: 5000	13 88
	Amount of data	Uint_16	2	15{SPC} Dec: 21 measurement points	00 15
	Data_1 ... Data_21	Uint_16	2	8A1{SPC}8A5{SPC}8AB{SPC}8AC{SPC}8A6{SPC}8AC{SPC}8B6{SPC}8C8{SPC}8C2{SPC}8C9{SPC}8CB{SPC}8C4{SPC}8E4{SPC}8E1{SPC}8EB{SPC}8E0{SPC}8F5{SPC}908{SPC}8FC{SPC}907{SPC}906{SPC} Measurement data Min. 22 mm: 16h Max. 20000 mm: 4E20h	08 A1 08 A5 08 AB 08 AC 08 A6 08 AC 08 B6 08 C8 08 C2 08 C9 08 CB 08 C4 08 E4 08 E1 08 EB 08 E0 08 F5 09 08 08 FC 09 07 09 06

Telegram structure: sRA LMDscandata (Example)					
Telegram part		Variable	Length	Values CoLa A (ASCII)	Values CoLa B (Binary)
Amount of 8 bit channels		Enum_16	2	0{SPC} No 8 bit data	00 00 No 8 bit data
Output channel (8 bit)	Content	String	5	–	–
	Scale factor	Real	4	–	–
	Scale factor offset	Real	4	–	–
	Start angle	Int_32	4	–	–
	Size of single angular step	Uint_16	2	–	–
	Amount of data	Uint_16	2	–	–
	Data_1 Data_n	Uint_8	1	–	–
Position		Enum_16	2	0{SPC} No position data	00 00 No position data
Position information	X position	Real	4	–	–
	Y position	Real	4	–	–
	Z position	Real	4	–	–
	X rotation	Real	4	–	–
	Y rotation	Real	4	–	–
	Z rotation	Real	4	–	–
	Rotations type	Enum_8	1	–	–
	Transmits the name of device	Uint_8	1	–	–
Name		Enum_16	2	0{SPC} No device name	00 00 No device name
Name info	Length of name	Enum_8	1	–	–
	Name in characters	String	2	–	–
Comment		Enum_16	2	0{SPC} No comment	00 00 No comment
Com-ment	Length of comment	Enum_8	1	–	–
	Comment in characters	String	2	–	–

Telegram structure: sRA LMDscandata (Example)					
Telegram part		Variable	Length	Values CoLa A (ASCII)	Values CoLa B (Binary)
Time		Enum_16	2	0{SPC} No time transmitted	00 00 No time transmitted
Time info	Year	Uint_16	2	–	–
	Month	Uint_8	1	–	–
	Day	Uint_8	1	–	–
	Hour	Uint_8	1	–	–
	Minute	Uint_8	1	–	–
	Second	Uint_8	1	–	–
	Microsecond	Uint_32	4	–	–
Event info		Enum_16	2	0{SPC} No event info available	00 00 No event info available
Event information	Type	String	4	–	–
	Encoder position	Uint_32	4	–	–
	Time of event	Uint_32	4	–	–
	Angle of event	Int_32	4	–	–
Frame				03 <ETX>	2B Checksum

Table 129: Example of one telegram stream

4.4 Time stamp

4.4.1 Set time stamp

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

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

The answer is always in ASCII format.

Attention: There is no real time clock inside the device. When the scanner is switched off and after a reboot, the time has to be set again.

However, it is possible to analyze the Off-time in order to evade this issue.

Telegram structure: sMN LSPsetdatetime (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3	All	sMN	73 4D 4E
Command	Set time stamp	String	14	All	LSPsetdatetime	4C 53 50 73 65 74 64 61 74 65 74 69 6D 65
Year		Uint_16	2	All	1970d ... +2099d (07B2h ... 0833h)	07 b2 ... 08 33
Month		Uint_8	1	All	01d ... +12d (01h ... 0Ch)	01 ... 0C
Day		Uint_8	1	All	01d ... +31d (01h ... 1Fh)	00 ... 1F
Hour		Uint_8	1	All	00d ... +23d (00h ... 17h)	00 ... 17
Minute		Uint_8	1	All	00d ... +59d (00h ... 3Bh)	00 ... 3B
Second		Uint_8	1	All	00d ... +59d (00h ... 3Bh)	00 ... 3B
Micro-second		Uint_32	4	All	00000000d ... +00999999d (00000000h ... 000F423Fh)	00 00 00 00 ... 00 0F 42 3F

Table 130: Telegram structure: sMN LSPsetdatetime

Example 1: sMN LSPsetdatetime

CoLa A	ASCII	<STX>sMN{SPC}LSPsetdatetime{SPC}7D9{SPC}2{SPC}11{SPC}10{SPC}22{SPC}0{SPC}0<ETX>
	Hex	02 73 4D 4E 20 4C 53 50 73 65 74 64 61 74 65 74 69 6D 65 20 37 44 39 20 32 20 31 31 20 31 30 20 32 32 20 30 20 30 03
CoLa B	Binary	02 02 02 02 00 00 00 1E 73 4D 4E 20 4C 53 50 73 65 74 64 61 74 65 74 69 6D 65 20 07 D9 02 11 10 22 00 00 00 00 00 A3

Table 131: Example 1: sMN LSPsetdatetime

Example 2: sMN LSPsetdatetime

CoLa A	ASCII	<STX>sMN{SPC}LSPsetdatetime{SPC}+2010{SPC}+01{SPC}+26{SPC}+10{SPC}+35{SPC}0{SPC}0<ETX>
	Hex	02 73 4D 4E 20 4C 53 50 73 65 74 64 61 74 65 74 69 6D 65 20 2B 32 30 31 30 20 2B 30 31 20 2B 32 36 20 2B 31 30 20 2B 33 35 20 2B 30 30 20 2B 30 30 30 30 03
CoLa B	Binary	02 02 02 02 00 00 00 1E 73 4D 4E 20 4C 53 50 73 65 74 64 61 74 65 74 69 6D 65 20 07 DA 01 1A 0A 23 00 00 00 00 00 A3

Table 132: Example 2: sMN LSPsetdatetime

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

Table 133: Telegram structure: sAN LSPsetdatetime

Example 1, 2: sAN LSPsetdatetime

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

Table 134: Example 1, 2: sAN LSPsetdatetime

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

4.4.2 Read time stamp and status of the measurement function

Command: sRN STIms

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

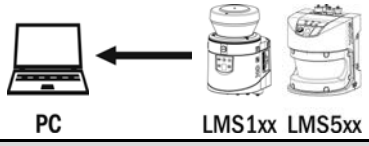
Table 135: Telegram structure: sRN STIms

Example: sRN STIms

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

Table 136: Example: sRN STIms

Answer: sRA STIms



Telegram structure: sRA STIms						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sRA	73 52 41
Command	Status and time	String	5	All	STIms	53 54 6C 6D 73
Status code	Status of the measurement function.	Enum_ 16	2	All	Undefined: 0 Initialization: 1 Configuration: 2 Lower case: 3 Rotating: 4 In preparation: 5 Ready: 6 Measurement active: 7	Undefined: 00 00 Initialization: 00 01 Configuration: 00 02 Lower case: 00 03 Rotating: 00 04 In preparation: 00 05 Ready: 00 06 Measurement active: 00 07
Temp. out of range	Device running in temp. range or not	Uint_8	1	All	False (in range) = 0 True (out of range = 1	False (in range) = 00 True (out of range = 01
Length of time para-		Uint_16	2	All	0d ... +65535d (00h ... FFFFh)	00 00 ... FF FF

Telegram structure: sRA STIms						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
meter						
Time	HH HH	Uint_16	2	All	0d ... 99d	00 00 ... 00 63
	:	Uint_8	1	All	:	3A
	MM MM	Uint_16	2	All	0d ... 99d	00 00 ... 00 63
	:	Uint_8	1	All	:	3A
	SS SS	Uint_16	2	All	0d ... 99d	00 00 ... 00 63
Length of date parameter		Uint_16	2	All	0d ... +65535d (00h ... FFFFh)	00 00 ... FF FF
Date	DD DD	Uint_16	2	All	0d ... 99d	00 00 ... 00 63
	.	Uint_8	1	All	.	2E
	MM MM	Uint_16	2	All	0d ... 99d	00 00 ... 00 63
	.	Uint_8	1	All	.	2E
	YY YY YY YY	Uint_32	4	All	0d ... 9999d	00 00 00 00 ... 00 00 27 0F
LED1		Uint_16	2	All	Inactive: 0 Active: 1	Inactive: 00 00 Active: 00 01
LED2		Uint_16	2	All	Inactive: 0 Active: 1	Inactive: 00 00 Active: 00 01
LED3		Uint_16	2	All	Inactive: 0 Active: 1	Inactive: 00 00 Active: 00 01
Reserved		Uint_16	3 × 2	All	0 0 0	00 00 00 00 00 00

Table 137: Telegram structure: sRA STIms

Example: sRA STIms

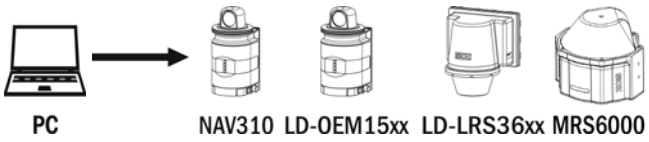
CoLa A	ASCII	<STX>sRA{SPC}STIms{SPC}7{SPC}0{SPC}8{SPC}16:36:54{SPC}10{SPC}17.03.2030{SPC}0{SPC}0{SPC}0<ETX>
	Hex	Not available
CoLa B	Binary	02 02 02 02 00 00 00 2F 73 52 41 20 53 54 6C 6D 73 20 00 07 00 00 08 00 10 3A 00 24 3A 00 36 00 0A 00 11 2E 00 03 2E 00 00 07 EE 00 00 00 00 00 00 00 00 00 00 00 00 17

Table 138: Example: sRA STIms

4.4.3 Read device time

Command to read the actual time of the internal clock (ms).

The timer is 32 counter with a resolution of 1 ms.



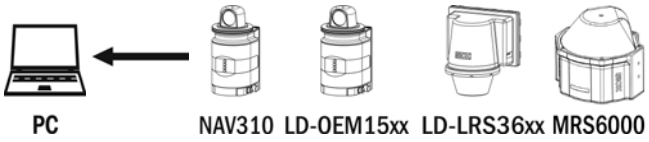
Telegram structure: sRN DeviceTime						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sRN	73 52 4E
Command		String	10	All	DeviceTime	44 65 76 69 63 65 54 69 6D 65

Table 139: Telegram structure: sRN DeviceTime

Example: sRN DeviceTime

CoLa A	ASCII	<STX>sRN{SPC}DeviceTime<ETX>
	Hex	02 73 52 4E 20 44 65 76 69 63 65 54 69 6D 65 03
CoLa B	Binary	02 02 02 02 00 00 00 0E 73 52 4E 20 44 65 76 69 63 65 54 69 6D 65 42

Table 140: Example: sRN DeviceTime



Telegram structure: sRA DeviceTime						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sRA	73 52 41
Command		String	10	All	DeviceTime	44 65 76 69 63 65 54 69 6D 65
Device time	Time	Uint_32	4	All	0d ... +9999d (0h ... 270Fh)	00 00 00 00 ... 00 00 27 0F

Table 141: Telegram structure: sRA DeviceTime

Example: sRA DeviceTime 0

CoLa A	ASCII	<STX>sRA{SPC}DeviceTime{SPC}0<ETX>
	Hex	0273 52 41 20 44 65 76 69 63 65 54 69 6D 65 20 00 03
CoLa B	Binary	02 02 02 02 00 00 00 10 73 52 41 20 44 65 76 69 63 65 54 69 6D 65 00 00 00 00 6D

Table 142: Example: sRA DeviceTime 0

4.4.4 Set NTP (Network Time Protocol) parameters

Set time synchronization

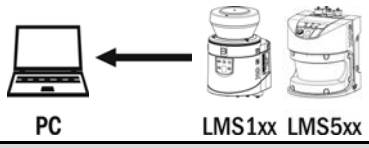
Telegram structure: sWN TSCRole (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Set NTP role	String	7	All	TSCRole	54 53 43 52 6F 6C 65
Variable data	NTP role	Uint_8	1	All	None: 0 Client: 1 Server: 2	None: 00 Client: 01 Server: 02

Table 143: Telegram structure: sWN TSCRole

Example: sWN TSCRole

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

Table 144: Example: sWN TSCRole



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

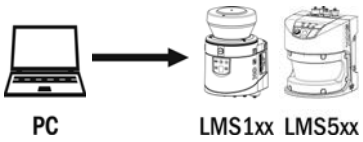
Table 145: Telegram structure: sWA TSCRole

Example: sWA TSCRole

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

Table 146: Example: sWA TSCRole

Set time synchronization interface



Telegram structure: sWN TSCTCInterface (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Set time synchronization interface	String	14	All	TSCTCInterface	54 53 43 54 43 49 6E 74 65 72 66 61 63 65
Variable data	Time synchronization interface data	Uint_8	1	All	Ethernet: 0 CAN: 1	Ethernet: 00 CAN: 01

Table 147: Telegram structure: sWN TSCTCInterface

Example: sWN TSCTCInterface

CoLa A	ASCII	<STX>sWN{SPC}TSCTCInterface{SPC}0<ETX>
	Hex	02 73 57 4E 20 54 53 43 54 43 49 6E 74 65 72 66 61 63 65 20 30 03
CoLa B	Binary	02 02 02 02 00 00 00 14 73 57 4E 20 54 53 43 54 43 49 6E 74 65 72 66 61 63 65 20 00 7C

Table 148: Example: sWN TSCTCInterface

Telegram structure: sWA TSCTCInterface						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command	Set time synchronization	String	14	All	TSCTCInterface	54 53 43 54 43 49 6E 74 65 72 66 61 63 65

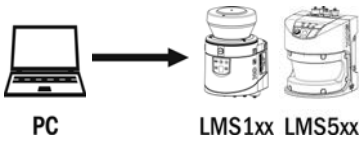
Table 149: Telegram structure: sWA TSCTCInterface

Example: sWA TSCTCInterface

CoLa A	ASCII	<STX>sWA{SPC}TSCTCInterface<ETX>
	Hex	02 73 57 41 20 54 53 43 54 43 49 6E 74 65 72 66 61 63 65 03
CoLa B	Binary	02 02 02 02 00 00 00 13 73 57 41 20 54 53 43 54 43 49 6E 74 65 72 66 61 63 65 20 73

Table 150: Example: sWA TSCTCInterface

Set time server IP address



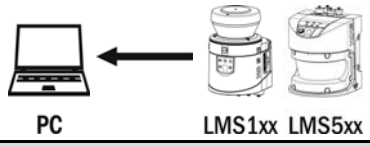
Telegram structure: sWN TSCTCSrvAddr (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Set time server IP address	String	12	All	TSCTCSrvAddr	54 53 43 54 43 53 72 76 41 64 64 72
IP address data	Set values in hex	Uint_32	4	All	00 00 00 00h ... FF FF FF FFh (decimal values unwieldy)	00 00 00 00 ... FF FF FF FF

Table 151: Telegram structure: sWN TSCTCSrvAddr

Example: sWN TSCTCSrvAddr 192.168.0.11

CoLa A	ASCII	<STX>sWN{SPC}TSCTCSrvAddr{SPC}C0{SPC}A8{SPC}00{SPC}0B<ETX>
	Hex	02 73 57 4E 20 54 53 43 54 43 53 72 76 41 64 64 72 20 C0 A8 00 0B 03
CoLa B	Binary	02 02 02 02 00 00 00 15 73 57 4E 20 54 53 43 54 43 53 72 76 41 64 64 72 20 C0 A8 00 0B 3E

Table 152: Example: sWN TSCTCSrvAddr 192.168.0.11



Telegram structure: sWA TSCTCSrvAddr						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command	Set time server IP address	String	12	All	TSCTCSrvAddr	54 53 43 54 43 53 72 76 41 64 64 72

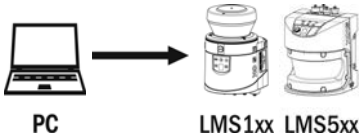
Table 153: Telegram structure: sWA TSCTCSrvAddr

Example: sWA TSCTCSrvAddr

CoLa A	ASCII	<STX>sWA{SPC}TSCTCSrvAddr<ETX>
	Hex	02 73 57 41 20 54 53 43 54 43 53 72 76 41 64 64 72 03
CoLa B	Binary	02 02 02 02 00 00 00 11 73 57 41 20 54 53 43 54 43 53 72 76 41 64 64 72 20 52

Table 154: Example: sWA TSCTCSrvAddr

Set time zone



Telegram structure: sWN TSCTCtimezone (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Set time zone	String	13	All	TSCTCtimezone	54 53 43 54 43 74 69 6D 65 7A 6F 6E 65
Time zone data	Set values in number of hours relative to GMT, hex specially coded	Int_8	1	All	[GMT + ...] -12d ... +12d (00h ... 18h)	[GMT + ...] 00 ... 18

Table 155: Telegram structure: sWN TSCTCtimezone

Example: sWN TSCTCtimezone GMT + 1 hour

CoLa A	ASCII	<STX>sWN{SPC}TSCTCtimezone{SPC}+1<ETX>
	Hex	02 73 57 4E 20 54 53 43 54 43 74 69 6D 65 7A 6F 6E 65 20 0D 03
CoLa B	Binary	02 02 02 02 00 00 00 13 73 57 4E 20 54 53 43 54 43 74 69 6D 65 7A 6F 6E 65 20 0D 3F

Table 156: Example: sWN TSCTCtimezone GMT + 1 hour

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

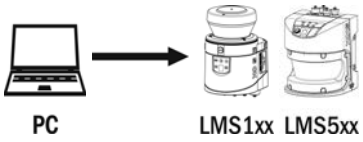
Table 157: Telegram structure: sWA TSCTCtimezone

Example: sWA TSCTCtimezone

CoLa A	ASCII	<STX>sWA{SPC}TSCTCtimezone<ETX>
	Hex	02 73 57 41 20 54 53 43 54 43 74 69 6D 65 7A 6F 6E 65 03
CoLa B	Binary	02 02 02 02 00 00 00 12 73 57 41 20 54 53 43 54 43 74 69 6D 65 7A 6F 6E 65 20 3D

Table 158: Example: sWA TSCTCtimezone

Set update time



Telegram structure: sWN TSCTCupdatetime (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Set update time of synchronization	String	15	All	TSCTCupdatetime	54 53 43 54 43 75 70 64 61 74 65 74 69 6D 65
Update time of synchroni-zation	Set values in seconds	Uint_32	4	All	+1d ... +3600d (01h ... 0E10h)	00 00 00 00 ... 00 00 0E 10

Table 159: Telegram structure: sWN TSCTCupdatetime

Example: sWN TSCTCupdatetime 600 s

CoLa A	ASCII	<STX>sWN{SPC}TSCTCupdatetime{SPC}+600<ETX>
	Hex	02 73 57 4E 20 54 53 43 54 43 75 70 64 61 74 65 74 69 6D 65 20 02 58 03
CoLa B	Binary	02 02 02 02 00 00 00 18 73 57 4E 20 54 53 43 54 43 75 70 64 61 74 65 74 69 6D 65 20 00 00 02 58 67

Table 160: Example: sWN TSCTCupdatetime 600 s

Telegram structure: sWA TSCTCupdatetime						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command	Set update time of synchronization	String	15	All	TSCTCupdatetime	54 53 43 54 43 75 70 64 61 74 65 74 69 6D 65

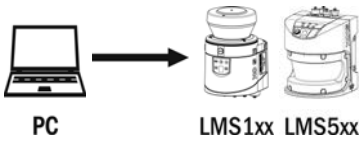
Table 161: Telegram structure: sWA TSCTCupdatetime

Example: sWA TSCTCupdatetime

CoLa A	ASCII	<STX>sWA{SPC}TSCTCupdatetime<ETX>
	Hex	02 73 57 41 20 54 53 43 54 43 75 70 64 61 74 65 74 69 6D 65 03
CoLa B	Binary	02 02 02 02 00 00 00 14 73 57 41 20 54 53 43 54 43 75 70 64 61 74 65 74 69 6D 65 20 32

Table 162: Example: sWA TSCTCupdatetime

Read for maximum offset time



Telegram structure: sRN TSCTCmaxoffset (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sRN	73 52 4E
Command	Read maximum offset time	String	14	All	TSCTCmaxoffset	54 53 43 54 43 6D 61 78 6F 66 66 73 65 74

Table 163: Telegram structure: sRN TSCTCmaxoffset

Example: sRN TSCTCmaxoffset

CoLa A	ASCII	<STX>sRN{SPC}TSCTCmaxoffset<ETX>
	Hex	02 73 52 4E 20 54 53 43 54 43 6D 61 78 6F 66 66 73 65 74 03
CoLa B	Binary	02 02 02 02 00 00 00 12 73 52 4E 20 54 53 43 54 43 6D 61 78 6F 66 66 73 65 74 65

Table 164: Example: sRN TSCTCmaxoffset

Telegram structure: sRA TSCTCmaxoffset						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sRA	73 52 41
Command	Read maximum offset time	String	14	All	TSCTCmaxoffset	54 53 43 54 43 6D 61 78 6F 66 66 73 65 74
Max. offset time	[Seconds as float according to IEEE754]	Real	4	All	0h ... FFFFFFFFh Min Value ~ -3.403*10^38 s Max Value ~+3.403*10^38 s	00 00 00 00 ... FF FF FF FF

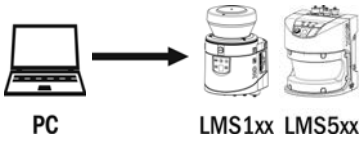
Table 165: Telegram structure: sRA TSCTCmaxoffset

Example: sRA TSCTCmaxoffset (18000 s)

CoLa A	ASCII	<STX>sRA{SPC}TSCTCmaxoffset{SPC}468CA000<ETX>
	Hex	02 73 52 41 20 54 53 43 54 43 6D 61 78 6F 66 66 73 65 74 20 46 8C A0 00 03
CoLa B	Binary	02 02 02 02 00 00 00 17 73 52 41 20 54 53 43 54 43 6D 61 78 6F 66 66 73 65 74 20 46 8C A0 00 20

Table 166: Example: sRA TSCTCmaxoffset 18000 s

Read for delay time



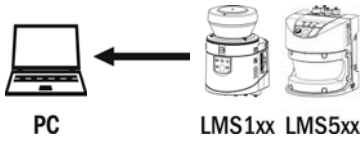
Telegram structure: sRN TSCTCdelay (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sRN	73 52 4E
Command	Read delay time	String	10	All	TSCTCdelay	54 53 43 54 43 64 65 6C 61 79

Table 167: Telegram structure: sRN TSCTCdelay

Example: sRN TSCTCdelay

CoLa A	ASCII	<STX>sRN{SPC}TSCTCdelay<ETX>
	Hex	02 73 52 4E 20 54 53 43 54 43 64 65 6C 61 79 03
CoLa B	Binary	02 02 02 02 00 00 00 0E 73 52 4E 20 54 53 43 54 43 64 65 6C 61 79 69

Table 168: Example: sRN TSCTCdelay



Telegram structure: sRA TSCTCdelay						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sRA	73 52 41
Command	Read for delay time	String	10	All	TSCTCdelay	54 53 43 54 43 64 65 6C 61 79
Max. offset time	[Seconds as float according to IEEE754]	Real	4	All	0h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF

Table 169: Telegram structure: sRA TSCTCdelay

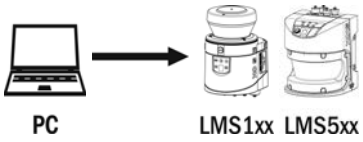
Example: sRA TSCTCdelay (0.003 s)

CoLa A	ASCII	<STX>sRA{SPC}TSCTCdelay{SPC}3B435B02<ETX>
	Hex	02 73 52 41 20 54 53 43 54 43 64 65 6C 61 79 20 3B 43 5B 02 03
CoLa B	Binary	02 02 02 02 00 00 00 13 73 52 41 20 54 53 43 54 43 64 65 6C 61 79 20 3B 43 5B 02 67

Table 170: Example: sRA TSCTCdelay 0.003 s

Reset maximum offset time

This command resets the maximum offset time, i.e. sets it to zero (0).



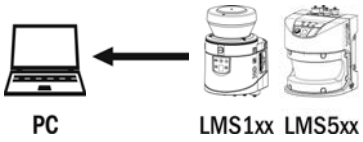
Telegram structure: sMN mResetMaxOff (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3	All	sMN	73 4D 4E
Command	Reset maximum offset time	String	12	All	mResetMaxOff	6D 52 65 73 65 74 4D 61 78 4F 66 66

Table 171: Telegram structure: sMN mResetMaxOff

Example: sMN mResetMaxOff

CoLa A	ASCII	<STX>sMN{SPC}mResetMaxOff<ETX>
	Hex	02 73 4D 4E 20 6D 52 65 73 65 74 4D 61 78 4F 66 66 03
CoLa B	Binary	02 02 02 02 00 00 00 10 73 4D 4E 20 6D 52 65 73 65 74 4D 61 78 4F 66 66 73

Table 172: Example: sMN mResetMaxOff



Telegram structure: sAN mResetMaxOff						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sAN	73 41 4E
Command	Read maximum offset time	String	12	All	mResetMaxOff	6D 52 65 73 65 74 4D 61 78 4F 66 66

Table 173: Telegram structure: sAN mResetMaxOff

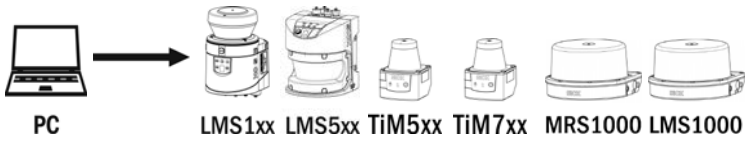
Example: sAN mResetMaxOff

CoLa A	ASCII	<STX>sAN{SPC}mResetMaxOff<ETX>
	Hex	02 73 41 4E 20 6D 52 65 73 65 74 4D 61 78 4F 66 66 03
CoLa B	Binary	02 02 02 02 00 00 00 11 73 41 4E 20 6D 52 65 73 65 74 4D 61 78 4F 66 66 20 5F

Table 174: Example: sAN mResetMaxOff

4.5 Filter

4.5.1 Set particle filter



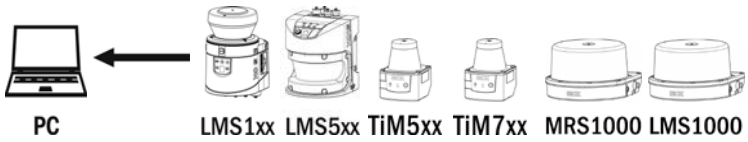
Telegram structure: sWN LFPparticle (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Set particle filter	String	11	All	LFPparticle	4C 46 50 70 61 72 74 69 63 6C 65
Status code	Code number	Bool_1	1	All	Inactive: 0 Active: 1	Inactive: 00 Active: 01
Thresh-old [5]	Particle threshold in mm	Uint_16	2	All	+500d (must be taken) (1F4h)	01 F4 (must be taken)

Table 175: Telegram structure: sWN LFPparticle

Example: sWN LFPparticle

CoLa A	ASCII	<STX>sWN{SPC}LFPparticle{SPC}1{SPC}+500<ETX>
	Hex	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	Binary	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

Table 176: Example: sWN LFPparticle



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

Table 177: Telegram structure: sWA LFPparticle

[5] Never change the threshold here, it is taken by the device to handle the particles.

Example: sWA LFPparticle

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

Table 178: Example: sWA LFPparticle

4.5.2 Set mean filter

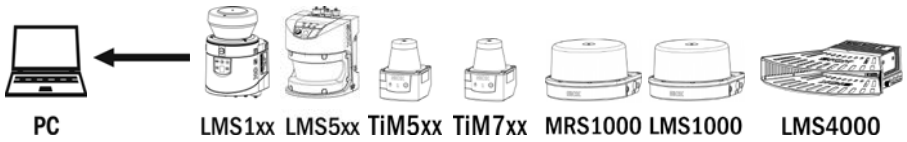
Telegram structure: sWN LFPmeanfilter (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Set mean filter	String	13	All	LFPmeanfilter	4C 46 50 6D 65 61 6E 66 69 6C 74 65 72
Status code	Code number	Bool_1	1	All	Inactive: 0 Active: 1	Inactive: 00 Active: 01
Number of scans	Number	Uint_16	2	All	+2d ... +100d (00 02h ... 00 64h)	00 02 ... 00 64
Final part	Reserved	Enum_8	1	All	0	00

Table 179: Telegram structure: sWN LFPmeanfilter

Example: sWN LFPmeanfilter

CoLa A	ASCII	<STX>sWN{SPC}LFPmeanfilter{SPC}1{SPC}+10{SPC}0<ETX>
	Hex	02 73 57 4E 20 4C 46 50 6D 65 61 6E 66 69 6C 74 65 72 20 31 20 2B 31 30 20 30 03
CoLa B	Binary	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 64 00 52

Table 180: Example: sWN LFPmeanfilter



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

Table 181: Telegram structure: sWA LFPmeanfilter

Example: sWA LFPmeanfilter

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

Table 182: Example: sWA LFPmeanfilter

4.5.3 Set n-pulse to 1-pulse filter (Echo filter)

Only LMS1xx, for LMS5xx take the echo filter.

Telegram structure: sWN LFPnto1filter (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	LMS1xx	sWN	73 57 4E
Command	Set n-to-1 filter	String	13	LMS1xx	LFPnto1filter	4C 46 50 6E 74 6F 31 66 69 6C 74 65 72
Status code	Code number	Bool_1	1	LMS1xx	Inactive: 0 Active: 1	Inactive: 00 Active: 01

Table 183: Telegram structure: sWN LFPnto1filter

Example: sWN LFPnto1filter

CoLa A	ASCII	<STX>sWN{SPC}LFPnto1filter{SPC}1<ETX>
	Hex	02 73 57 4E 20 4C 46 50 6E 74 6F 31 66 69 6C 74 65 72 20 31 03
CoLa B	Binary	02 02 02 02 00 00 00 13 73 57 4E 20 4C 46 50 6E 74 6F 31 66 69 6C 74 65 72 20 01 75

Table 184: Example: sWN LFPnto1filter

Telegram structure: sWA LFPnto1filter						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	LMS1xx	sWA	73 57 41
Command	Set n-to-1 filter	String	13	LMS1xx	LFPnto1filter	4C 46 50 6E 74 6F 31 66 69 6C 74 65 72

Table 185: Telegram structure: sWA LFPnto1filter

Example: sWA LFPnto1filter

CoLa A	ASCII	<STX>sWA{SPC}LFPnto1filter<ETX>
	Hex	02 73 57 41 20 4C 46 50 6E 74 6F 31 66 69 6C 74 65 72 03
CoLa B	Binary	02 02 02 02 00 00 00 12 73 57 41 20 4C 46 50 6E 74 6F 31 66 69 6C 74 65 72 7B

Table 186: Example: sWA LFPnto1filter

4.5.4 Set echo filter

Only LMS5xx. For LMS1xx use the n-pulse to 1-pulse filter.

Telegram structure: sWN FREchoFilter (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Set echo filter	String	12	All	FREchoFilter	46 52 45 63 68 6F 46 69 6C 74 65 72

Telegram structure: sWN FREchoFilter (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Status code	Code number	Enum_8	1	All	First echo: 0 All echos: 1 Last echo: 2	First echo: 00 All echos: 01 Last echo: 02

Table 187: Telegram structure: sWN FREchoFilter

Example: sWN FREchoFilter

CoLa A	ASCII	<STX>sWN{SPC}FREchoFilter{SPC}1<ETX>
	Hex	02 73 57 4E 20 46 52 45 63 68 6F 46 69 6C 74 65 72 20 31 03
CoLa B	Binary	02 02 02 02 00 00 00 12 73 57 4E 20 46 52 45 63 68 6F 46 69 6C 74 65 72 20 01 7E Only available with firmware versions > V1.10.

Table 188: Example: sWN FREchoFilter

Telegram structure: sWA FREchoFilter						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command	Set echo filter	String	12	All	FREchoFilter	46 52 45 63 68 6F 46 69 6C 74 65 72

Table 189: Telegram structure: sWA FREchoFilter

Example: sWa FREchoFilter

CoLa A	ASCII	<STX>sWA{SPC}FREchoFilter<ETX>
	Hex	02 73 57 41 20 46 52 45 63 68 6F 46 69 6C 74 65 72 03
CoLa B	Binary	02 02 02 02 00 00 00 11 73 57 41 20 46 52 45 63 68 6F 46 69 6C 74 65 72 20 70 Only available with firmware versions > V1.10 LMS5xx.

Table 190: Example: sWa FREchoFilter

4.5.5 Set and read fog filter

Set fog filter (LMS1xx)

Telegram structure: sWN MSsuppmode (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Set fog filter	String	10	All	MSsuppmode	4D 53 73 75 70 70 6D 6F 64 65
Status code	Code number	Bool_1	1	All	Glitch: 0 Fog: 1	Glitch: 00 Fog: 01

Table 191: Telegram structure: sWN MSsuppmode

Example: sWN MSsuppmode

CoLa A	ASCII	<STX>sWN{SPC}MSsuppmode{SPC}1<ETX>
	Hex	02 73 57 4E 20 4D 53 73 75 70 70 6D 6F 64 65 20 31 03
CoLa B	Binary	02 02 02 02 00 00 00 10 73 57 4E 20 4D 53 73 75 70 70 6D 6F 64 65 20 01 70

Table 192: Example: sWN MSsuppmode

Telegram structure: sWA MSsuppmode						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command	Set fog filter	String	10	All	MSsuppmode	4D 53 73 75 70 70 6D 6F 64 65

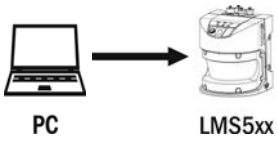
Table 193: Telegram structure: sWA MSsuppmode

Example: sWA MSsuppmode

CoLa A	ASCII	<STX>sWA{SPC}MSsuppmode<ETX>
	Hex	02 73 57 41 20 4D 53 73 75 70 70 6D 6F 64 65 03
CoLa B	Binary	02 02 02 02 00 00 00 0F 73 57 41 20 4D 53 73 75 70 70 6D 6F 64 65 7E

Table 194: Example: sWA MSsuppmode

Set fog filter (LMS5xx)



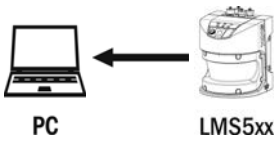
Telegram structure: sWN CLFogFilterEn (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Enable fog filter	String	13	All	CLFogFilterEn	43 4C 46 6F 67 46 69 6C 74 65 72 45 6E
Status code	Enable or disable fog filter	Bool_1	1	All	Disable: 0 Enable: 1	Disable: 00 Enable: 01

Table 195: Telegram structure: sWN CLFogFilterEn

Example: sWN CLFogFilterEn

CoLa A	ASCII	<STX>sWN{SPC}CLFogFilterEn{SPC}1<ETX>
	Hex	02 73 57 4E 20 43 4C 46 6F 67 46 69 6C 74 65 72 45 6E 20 31 03
CoLa B	Binary	02 02 02 02 00 00 00 13 73 57 4E 20 43 4C 46 6F 67 46 69 6C 74 65 72 45 6E 20 01 21

Table 196: Example: sWN CLFogFilterEn



Telegram structure: sWA CLFogFilterEn						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command	Enable fog filter	String	13	All	CLFogFilterEn	43 4C 46 6F 67 46 69 6C 74 65 72 45 6E

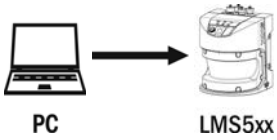
Table 197: Telegram structure: sWA CLFogFilterEn

Example: sWA CLFogFilterEn

CoLa A	ASCII	<STX>sWA{SPC}CLFogFilterEn<ETX>
	Hex	02 73 57 41 20 43 4C 46 6F 67 46 69 6C 74 65 72 45 6E 03
CoLa B	Binary	02 02 02 02 00 00 00 12 73 57 41 20 43 4C 46 6F 67 46 69 6C 74 65 72 45 6E 20 2F

Table 198: Example: sWA CLFogFilterEn

Read for enabled fog filter (LMS5xx)



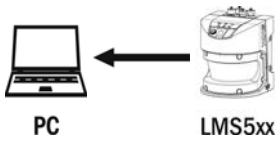
Telegram structure: sRN CLFogFilterEn						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sRN	73 52 4E
Command	Enabled fog filter	String	13	All	CLFogFilterEn	43 4C 46 6F 67 46 69 6C 74 65 72 45 6E

Table 199: Telegram structure: sRN CLFogFilterEn

Example: sRN CLFogFilterEn

CoLa A	ASCII	<STX>sRN{SPC}CLFogFilterEn<ETX>
	Hex	02 73 52 4E 20 43 4C 46 6F 67 46 69 6C 74 65 72 45 6E 03
CoLa B	Binary	02 02 02 02 00 00 00 11 73 52 4E 20 43 4C 46 6F 67 46 69 6C 74 65 72 45 6E 05

Table 200: Example: sRN CLFogFilterEn



Telegram structure: sRA CLFogFilterEn						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sRA	73 52 41
Command	Enabled fog filter	String	13	All	CLFogFilterEn	43 4C 46 6F 67 46 69 6C 74 65 72 45 6E
Status code	Fog filter enabled or disabled	Bool_1	1	All	Disabled: 0 Enabled: 1	Disabled: 00 Enabled: 01

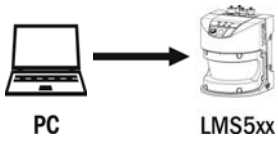
Table 201: Telegram structure: sRA CLFogFilterEn

Example: sRA CLFogFilterEn

CoLa A	ASCII	<STX>sRA{SPC}CLFogFilterEn{SPC}1<ETX>
	Hex	02 73 52 41 20 43 4C 46 6F 67 46 69 6C 74 65 72 45 6E 20 01 03
CoLa B	Binary	02 02 02 02 00 00 00 13 73 52 41 20 43 4C 46 6F 67 46 69 6C 74 65 72 45 6E 20 01 2B

Table 202: Example: sRA CLFogFilterEn

Set sensitivity fog filter (LMS5xx)



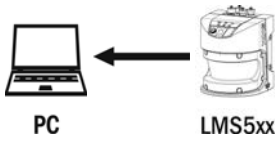
Telegram structure: sWN MCSenseLevel (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Sense level	String	12	All	MCSenseLevel	4D 43 53 65 6E 73 65 4C 65 76 65 6C
Sensitivity level		Uint_8	1	All	1 ... 6	01 ... 06

Table 203: Telegram structure: sWN MCSenseLevel

Example: sWN MCSenseLevel

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

Table 204: Example: sWN MCSenseLevel



Telegram structure: sWA MCSenseLevel						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command	Sense level	String	12	All	MCSenseLevel	4D 43 53 65 6E 73 65 4C 65 76 65 6C

Table 205: Telegram structure: sWA MCSenseLevel

Example: sWA MCSenseLevel

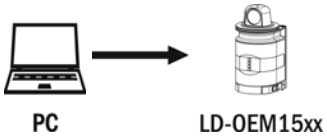
CoLa A	ASCII	<STX>sWA{SPC}MCSenseLevel<ETX>
	Hex	02 73 57 41 20 4D 43 53 65 6E 73 65 4C 65 76 65 6C 20 03
CoLa B	Binary	02 02 02 02 00 00 00 0F 73 57 41 20 4D 43 53 65 6E 73 65 4C 65 76 65 6C 20 7E

Table 206: Example: sWA MCSenseLevel

4.5.6 Enable/disable digital nearfield filter

Activates or deactivates the nearfield filter of the LD series.

Do not change the setting on LD-LRS XXXX !



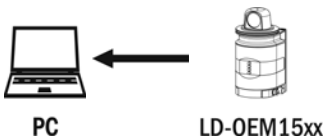
Telegram structure: sWN CLNFDigFilterEn (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Digital nearfield filter	String	15	All	CLNFDigFilterEn	43 4C 4E 46 44 69 67 46 69 6C 74 65 72 45 6E
Status code	Code number	Bool_1	1	All	Inactive: 0 Active: 1	Inactive: 00 Active: 01

Table 207: Telegram structure: sWN CLNFDigFilterEn

Example: sWN CLNFDigFilterEn

CoLa A	ASCII	<STX>sWN{SPC}CLNFDigFilterEn{SPC}1<ETX>
	Hex	02 73 57 4E 20 43 4C 4E 46 44 69 67 46 69 6C 74 65 72 45 6E 20 31 03
CoLa B	Binary	02 02 02 02 00 00 00 16 73 57 4E 20 43 4C 4E 46 44 69 67 46 69 6C 74 65 72 45 6E 20 01 51

Table 208: Example: sWN CLNFDigFilterEn



Telegram structure: sWA CLNFDigFilterEn						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command	Digital nearfield filter	String	15	All	CLNFDigFilterEn	43 4C 4E 46 44 69 67 46 69 6C 74 65 72 45 6E

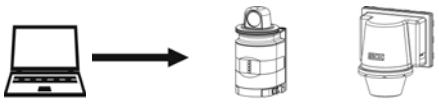
Table 209: Telegram structure: sWA CLNFDigFilterEn

Example: sWA CLNFDigFilterEn

CoLa A	ASCII	<STX>sWA{SPC}CLNFDigFilterEn<ETX>
	Hex	02 73 57 41 20 43 4C 4E 46 44 69 67 46 69 6C 74 65 72 45 6E 03
CoLa B	Binary	02 02 02 02 00 00 00 13 73 57 41 20 43 4C 4E 46 44 69 67 46 69 6C 74 65 72 45 6E 03

Table 210: Example: sWA CLNFDigFilterEn

4.5.7 Set digital nearfield filter sector selection



Do not change the setting on LD-LRS XXXX !

Telegram structure: sWN CLHWFilterSectEn (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Sector function	String	16	All	CLHWFilterSectEn	43 4C 48 57 46 69 6C 74 65 72 53 65 63 74 45 6E
Status code	Active sector vector	Bool_1	4 × 1	All	Active in none of the sectors: 0 0 0 0 Active in all sectors: 1 1 1 1	Active in none of the sectors: 00 00 00 00 Active in all sectors: 01 01 01 01

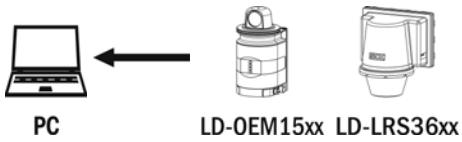
Table 211: Telegram structure: sWN CLHWFilterSectEn

Example: sWN CLHWFilterSectEn

Enable Nearfield Suppression for sector 1, disable for sectors 2, 3 and 4.

CoLa A	ASCII	<STX>sWN{SPC}CLHWFilterSectEn{SPC}1{SPC}0{SPC}0{SPC}0<ETX>
	Hex	02 73 57 4E 20 43 4C 48 57 46 69 6C 74 65 72 53 65 63 74 45 6E 20 31 20 30 20 30 20 30 03
CoLa B	Binary	02 02 02 02 00 00 00 19 73 57 4E 20 43 4C 48 57 46 69 6C 74 65 72 53 65 63 74 45 6E 20 31 30 30 30 51

Table 212: Example: sWN CLHWFilterSectEn 1 0 0 0



Telegram structure: sWA CLHWFilterSectEn						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command	Sector function	String	16	All	CLHWFilterSectEn	43 4C 48 57 46 69 6C 74 65 72 53 65 63 74 45 6E

Table 213: Telegram structure: sWA CLHWFilterSectEn

Example: sWA CLHWFilterSectEn

CoLa A	ASCII	<STX>sWA{SPC}CLHWFilterSectEn<ETX>
	Hex	02 73 57 41 20 43 4C 48 57 46 69 6C 74 65 72 53 65 63 74 45 6E 03
CoLa B	Binary	02 02 02 02 00 00 00 15 73 57 41 20 43 4C 48 57 46 69 6C 74 65 72 53 65 63 74 45 6E 20 5F

Table 214: Example: sWA CLHWFilterSectEn

4.5.8 Set Median Filter

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

Telegram structure: sWN LFPmedianfilter (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Sector function	String	16	All	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	All	Inactive: 0 Active: 1	00 01
Reserved	Always 3	Uint_16	2	All	3	00 03

Table 215: Telegram structure: sWN LFPmedianfilter

Example: sWN LFPmedianfilter

Enable Median filter

CoLa A	ASCII	<STX>sWN{SPC}LFPmedianfilter{SPC}1{SPC}3<ETX>
	Hex	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	Binary	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

Table 216: Example: sWN LFPmedianfilter 1 3

Telegram structure: sWA LFPmedianfilter						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command	Sector function	String	16	All	LFPmedianfilter	4C 46 50 6D 65 64 69 61 6E 66 69 6C 74 65 72

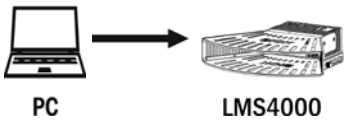
Table 217: Telegram structure: sWA LFPmedianfilter

Example: sWA LFPmedianfilter

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

Table 218: Example: sWA LFPmedianfilter

4.5.9 Set Edge Filter



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

Telegram structure: sWN LFPedgefilter (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E

Telegram structure: sWN LFPedgefilter (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command	Sector function	String	16	All	LFPedgefilter	4C 46 50 65 64 67 65 66 69 6C 74 65 72
Status code	Activate / deactivate edge filter	Bool_1	1	All	Inactive: 0 Active: 1	00 01

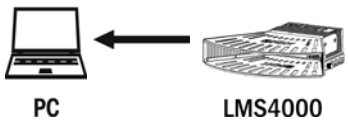
Table 219: Telegram structure: sWN LFPedgefilter

Example: sWN LFPedgefilter

Enable Median filter

CoLa A	ASCII	<STX>sWN{SPC} LFPedgefilter1<ETX>
	Hex	02 73 57 4E 20 4C 46 50 65 64 67 65 66 69 6C 74 65 72 20 31 03
CoLa B	Binary	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

Table 220: Example: sWN LFPedgefilter 1



Telegram structure: sWA LFPedgefilter						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command	Sector function	String	16	All	LFPedgefilter	4C 46 50 65 64 67 65 66 69 6C 74 65 72

Table 221: Telegram structure: sWA LFPmedianfilter

Example: sWA LFPedgefilter

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

Table 222: Example: sWA LFPedgefilter

4.6 Encoder

4.6.1 Set increment source

Telegram structure: sWN LICsrc (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Set increment source	String	6	All	LICsrc	4C 49 43 73 72 63
Increment source		Enum_8	1	All	Fixed speed: 0 Encoder: 1	Fixed speed: 00 Encoder: 01

Table 223: Telegram structure: sWN LICsrc

Example: sWN LICsrc

CoLa A	ASCII	<STX>sWN{SPC}LICsrc{SPC}0<ETX>
	Hex	02 73 57 4E 20 4C 49 43 73 72 63 20 30 03
CoLa B	Binary	02 02 02 02 00 00 00 0C 73 57 4E 20 4C 49 43 73 72 63 20 01 4F

Table 224: Example: sWN LICsrc

Telegram structure: sWA LICsrc						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command	Set increment source	String	6	All	LICsrc	4C 49 43 73 72 63

Table 225: Telegram structure: sWA LICsrc

Example: sWA LICsrc

CoLa A	ASCII	<STX>sWA{SPC}LICsrc<ETX>
	Hex	02 73 57 41 20 4C 49 43 73 72 63 03
CoLa B	Binary	02 02 02 02 00 00 00 0B 73 57 41 20 4C 49 43 73 72 63 41

Table 226: Example: sWA LICsrc

4.6.2 Set encoder settings

Telegram structure: sWN LICencset (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Encoder settings	String	9	All	LICencset	4C 49 43 65 6E 63 73 65 74
Encoder setting		Enum_8	1	All	Off: 0 Single increment/INC1: 1 Direction recognition (phase): 2 Direction recognition (level): 3	00 01 02 03
				LMS 4000 (+ above)	Fixed increment speed / ticks (1 kHz): 4	04

Table 227: Telegram structure: sWN LICencset

Example: sWN LICencset

CoLa A	ASCII	<STX>sWN{SPC}LICencset{SPC}0<ETX>
	Hex	02 73 57 4E 20 4C 49 43 65 6E 63 73 65 74 20 30 03
CoLa B	Binary	02 02 02 02 00 00 00 0F 73 57 4E 20 4C 49 43 65 6E 63 73 65 74 20 03 25

Table 228: Example: sWN LICencset

Telegram structure: sWA LICencset						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command	Encoder settings	String	9	All	LICencset	4C 49 43 65 6E 63 73 65 74

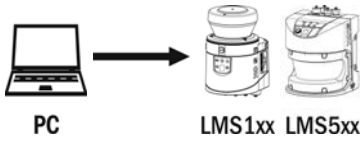
Table 229: Telegram structure: sWA LICencset

Example: sWA LICencset

CoLa A	ASCII	<STX>sWA{SPC}LICencset<ETX>
	Hex	02 73 57 41 20 4C 49 43 65 6E 63 73 65 74 03
CoLa B	Binary	02 02 02 02 00 00 00 0E 73 57 41 20 4C 49 43 65 6E 63 73 65 74 20 29

Table 230: Example: sWA LICencset

4.6.3 Set encoder resolution



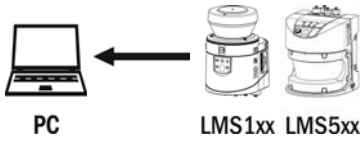
Telegram structure: sWN LICencres (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Set encoder resolution	String	9	All	LICencres	4C 49 43 65 6E 63 72 65 73
Encoder resolution	Resolution value in mm/Inc as float according to IEEE754	Real	4	All	+0.001d ... +2000d	3A 83 12 6F ... 44 FA 00 00 (see IEEE 754)

Table 231: Telegram structure: sWN LICencres

Example: sWN LICencres

CoLa A	ASCII	<STX>sWN{SPC}LICencres{SPC}+1000<ETX>
	Hex	02 73 57 4E 20 4C 49 43 65 6E 63 72 65 73 20 2B 31 30 30 30 03
CoLa B	Binary	02 02 02 02 00 00 00 10 73 57 4E 20 4C 49 43 65 6E 63 72 65 73 20 44 7A 00 00 1E

Table 232: Example: sWN LICencres



Telegram structure: sWA LICencres						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command	Set encoder resolution	String	9	All	LICencres	4C 49 43 65 6E 63 72 65 73

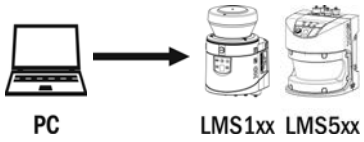
Table 233: Telegram structure: sWA LICencres

Example: sWA LICencres

CoLa A	ASCII	<STX>sWA{SPC}LICencres<ETX>
	Hex	02 73 57 41 20 4C 49 43 65 6E 63 72 65 73 03
CoLa B	Binary	02 02 02 02 00 00 00 0D 73 57 41 20 4C 49 43 65 6E 63 72 65 73 00

Table 234: Example: sWA LICencres

4.6.4 Set fixed speed



Telegram structure: sWN LICFixVel (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Set fixed speed	String	9	All	LICFixVel	4C 49 43 46 69 78 56 65 6C
Fixed speed	Speed in m/s as float according to IEEE754	Real	4	All	+0.001d ... +10.0d	3A 83 12 6F... 41 20 00 00

Table 235: Telegram structure: sWN LICFixVel

Example: sWN LICFixVel

CoLa A	ASCII	<STX>sWN{SPC}LICFixVel{SPC}+5<ETX>
	Hex	02 73 57 4E 20 4C 49 43 46 69 78 56 65 6C 20 2B 35 03
CoLa B	Binary	02 02 02 02 00 00 00 0B 73 57 4E 20 4C 49 43 46 69 78 56 65 6C 20 40 A0 00 00 C4

Table 236: Example: sWN LICFixVel

Telegram structure: sWA LICFixVel						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command	Set fixed speed	String	9	All	LICFixVel	4C 49 43 46 69 78 56 65 6C

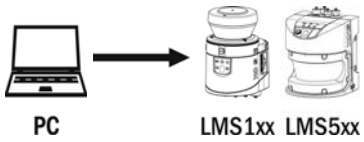
Table 237: Telegram structure: sWA LICFixVel

Example: sWA LICFixVel

CoLa A	ASCII	<STX>sWA{SPC}LICFixVel<ETX>
	Hex	02 73 57 41 20 4C 49 43 46 69 78 56 65 6C 03
CoLa B	Binary	02 02 02 02 00 00 00 0D 73 57 41 20 4C 49 43 46 69 78 56 65 6C 0B

Table 238: Example: sWA LICFixVel

4.6.5 Read speed threshold



Telegram structure: sRN LICSpTh						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sRN	73 52 4E
Command	Read speed threshold	String	7	All	LICSpTh	4C 49 43 53 70 54 68

Table 239: Telegram structure: sRN LICSpTh

Example: sRN LICSpTh

CoLa A	ASCII	<STX>sRN{SPC}LICSpTh<ETX>
	Hex	02 73 52 4E 20 4C 49 43 53 70 54 68 03
CoLa B	Binary	02 02 02 02 00 00 00 0D 73 52 4E 20 4C 49 43 53 70 54 68 16

Table 240: Example: sRN LICSpTh

Telegram structure: sRA LICSpTh						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sRA	73 52 41
Command	Speed threshold	String	7	All	LICSpTh	4C 49 43 53 70 54 68
Speed threshold	Speed threshold in %	Uint_8	2	All	+1d ... +20d (01h ... 14h)	01 ... 14

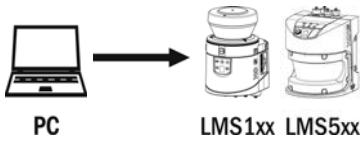
Table 241: Telegram structure: sRA LICSpTh

Example: sRA LICSpTh

CoLa A	ASCII	<STX>sRA{SPC}LICSpTh{SPC}5<ETX>
	Hex	02 73 52 41 20 4C 49 43 53 70 54 68 20 35 03
CoLa B	Binary	02 02 02 02 00 00 00 0D 73 52 41 20 4C 49 43 53 70 54 68 20 05 3C

Table 242: Example: sRA LICSpTh

4.6.6 Read encoder speed



Telegram structure: sRN LICencsp						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sRN	73 52 4E
Command	Read encoder speed	String	8	All	LICencsp	4C 49 43 65 6E 63 73 70

Table 243: Telegram structure: sRN LICencsp

Example: sRN LICencsp

CoLa A	ASCII	<STX>sRN{SPC}LICencsp <ETX>
	Hex	02 73 52 4E 20 4C 49 43 65 6C 63 73 70 03
CoLa B	Binary	02 02 02 02 00 00 00 11 73 52 4E 20 4C 49 43 65 6E 63 73 70 62

Table 244: Example: sRN LICencsp

Telegram structure: sRA LICencsp						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sRA	73 52 41
Command	Read encoder speed	String	8	All	LICencsp	4C 49 43 65 6E 63 73 70
Encoder speed	[Speed in m/s as float according to IEEE754]	Real	4	All	0h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF

Table 245: Telegram structure: sRA LICencsp

Example: sRA LICencsp (0 m/s)

CoLa A	ASCII	<STX>sRA{SPC}LICencsp{SPC}0<ETX>
	Hex	02 73 52 41 20 4C 49 43 65 6C 63 73 70 20 30 30 30 30 30 30 30 30 03
CoLa B	Binary	02 02 02 02 00 00 00 11 73 52 41 20 4C 49 43 65 6E 63 73 70 20 00 00 00 00 4D

Table 246: Example: sRA LICencsp

4.7 Inputs and Outputs

4.7.1 Port Configration of all I/Os

Telegram structure: sRN PortConfiguration						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sRN	73 52 4E
Command	Ask for port configuration	String	12	All	PortConfiguration	50 6F 72 74 43 6F 6E 66 69 67 75 72 61 74 69 6F 6E

Table 247: Telegram structure: sRN PortConfiguration

Example: sRN PortConfiguration

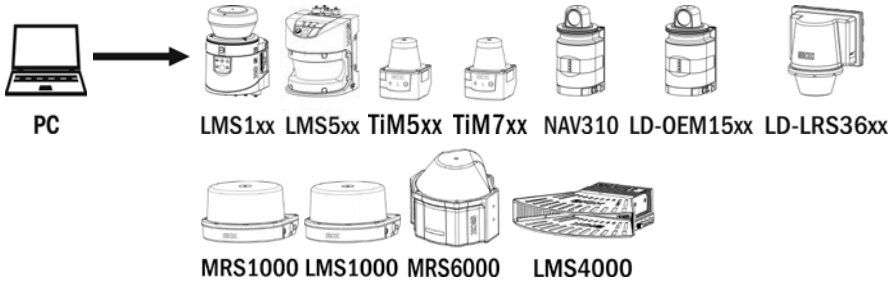
CoLa A	ASCII	<STX>sRN{SPC}PortConfiguration<ETX>
	Hex	02 73 52 4E 20 50 6F 72 74 43 6F 6E 66 69 67 75 72 61 74 69 6F 6E 03
CoLa B	Binary	02 02 02 02 00 00 00 15 73 52 4E 20 50 6F 72 74 43 6F 6E 66 69 67 75 72 61 74 69 6F 6E 26

Table 248: Example: sRN PortConfiguration

Telegram structure: sRA PortConfiguration						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sRA	73 52 41
Command	Configuration of all I/Os	String	12	All	PortConfiguration	50 6F 72 74 43 6F 6E 66 69 67 75 72 61 74 69 6F 6E
Start of loop, number of loops = amount of all current and future Inputs and Outputs of device family						
Port Type	Input or Output	Enum_8	1	All	Input: 0 Ouput: 1	00 01
Port Name	Amount of characters of the following port name	Uint_16	2	All	0h ... 20h	00 00 ... 00 20
	Port name	String	16	All	[Port name]	[Port name]

Telegram structure: sRA PortConfiguration							
Telegram part		**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Input Settings	Logic	Logic of the input	Bool_1	1	All	Active High: 0 Active low: 1	00 01
	Debouncing	Select debouncing time in ms	Uint_16	2	All	0h ... 27 10 (max. 10,000ms)	00 00 ... 27 10
	Sensitivity	Status change at Edge or Level	Enum_8	1	All	Edge: 0 Level: 1	00 01
	Reserved	Reserved value 1	Uint_16	2	All	0h	00 00
	Reserved	Reserved value 2	Uint_16	2	All	0h	00 00
Output Settings	Logic	Logic of the output	Bool_1	1	All	Active High: 0 Active low: 1	00 01
	Output Mode	PNP, NPN or Push-Pull	Enum_8	1	All	PNP: 0 NPN: 1 Push-Pull: 2	00 01 02
	Restart type	Restart behavior of output after event: immediatly or after specific time	Enum_8	1	All	Immediatly: 0 Time: 1	00 01
	Restart time	[Only with restart type = Time], time in ms	Uint_32	4	All	14h ... 927C0 (20 ms ... 600,000 ms)	00 00 00 14 ... 00 09 27 C0
	Combination	Combining multiple Events and/or Inputs	Enum_8	1	All	AND: 0 OR: 1 XOR: 2	00 01 02
				1	LMS 4000	Always: 0	00
	Reserved	Reserved value 3	Uint_16	2	All	0h	00 00
	Reserved	Reserved value 4	Uint_16	2	All	0h	00 00
	Sources	Amount (n) of combined sources	Uint_16	2	All	0h ... FFFFh	00 00 ... FF FF
					LMS 4000	0 ... 1	00 00 ... 00 01
		Start of source loop, number of loops = amount of combined sources					
		Source name	String	4	All	[Source]	[Source]
		Source Inverted or not	Bool_1	1	All	Not inverted: 0 Inverted: 1	00 01
		Reserved value 5	Uint_8	1	All	0h	00
		Reserved value 6	Uint_8	1	All	0h	00
		Stop of source loop					
Reserved		Reserved value 7	Uint_16	2	All	0h	00 00
Reserved		Reserved value 8	Uint_16	2	All	0h	00 00
Reserved		Reserved value 9	Uint_16	2	All	0h	00 00
Reserved		Reserved value 10	Uint_16	2	All	0h	00 00
Stop of loop							

4.7.2 Read state of the outputs



Telegram structure: sRN LIDoutputstate						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sRN	73 52 4E
Command	Output state	String	14	All	LIDoutputstate	4C 49 44 6F 75 74 70 75 74 73 74 61 74 65

Table 249: Telegram structure: sRN LIDoutputstate

Example: sRN LIDoutputstate

CoLa A	ASCII	<STX>sRN{SPC}LIDoutputstate<ETX>
	Hex	02 73 52 4E 20 4C 49 44 6F 75 74 70 75 74 73 74 61 74 65 03
CoLa B	Binary	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

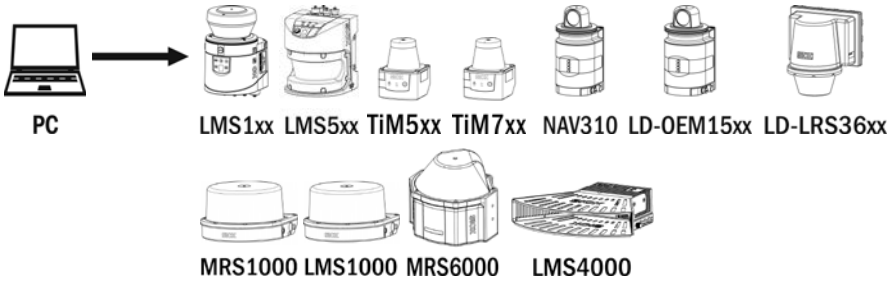
Table 250: Example: sRN LIDoutputstate

Telegram structure: sRA LIDoutputstate						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Find complete telegram structure of the answer in section 4.7.3 „Send outputstate by event“ on page 129.						

Table 251: Telegram structure: sRA LIDoutputstate

4.7.3 Send outputstate by event

Output telegram is sent every time an output state changes.



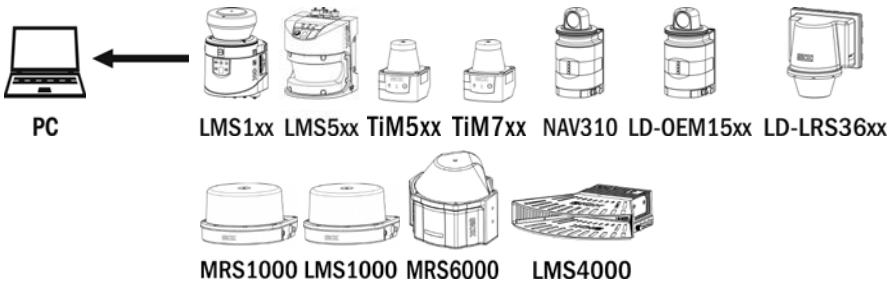
Telegram structure: sEN LIDoutputstate						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Event	String	3	All	sEN	73 45 4E
Command	Output state	String	14	All	LIDoutputstate	4C 49 44 6F 75 74 70 75 74 73 74 61 74 65
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

Table 252: Telegram structure: sEN LIDoutputstate

Example: sEN LIDoutputstate

CoLa A	ASCII	<STX>sEN{SPC}LIDoutputstate{SPC}1<ETX>
	Hex	02 73 45 4E 20 4C 49 44 6F 75 74 70 75 74 73 74 61 74 65 20 31 03
CoLa B	Binary	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

Table 253: Example: sEN LIDoutputstate



Telegram structure: sRA/sSN LIDoutputstate						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sRA/sSN	73 52 41 / 73 53 4E
Command	Output state	String	14	All	LIDoutputstate	4C 49 44 6F 75 74 70 75 74 73 74 61 74 65

Telegram structure: sRA/sSN LIDoutputstate						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Status code	Version number	Uint_16	2	All	0 … FFFFh	00 00 … FF FF
	System counter (time in µs since power up max. 71min then starting from 0 again)	Uint_32	4		0 … FFFFFFFFh	00 00 00 00 … FF FF FF FF
State of the outputs and count value in hex. (values of an example)	Out1 state	Enum_8	1	All	Not active: 0 Active: 1 Output not used: 2	Not active: 0 Active: 1 Output not used: 2
	Out1 count	Uint_32	4		0 … FFFFFFFFh	00 00 00 00 … FF FF FF FF
	Out2 state	Enum_8	1	LMS1xx LMS5xx LD-OEM15xx LD-LRS36xx MRS1000 MRS6000	Not active: 0 Active: 1 Output not used: 2	Not active: 0 Active: 1 Output not used: 2
	Out2 count	Uint_32	4		0 … FFFFFFFFh	00 00 00 00 … FF FF FF FF
	Out3 state	Enum_8	1		Not active: 0 Active: 1 Output not used: 2	Not active: 0 Active: 1 Output not used: 2
	Out3 count	Uint_32	4		0 … FFFFFFFFh	00 00 00 00 … FF FF FF FF
	Out4 state	Enum_8	1	LMS5xx LD-OEM15xx LD-LRS36xx MRS1000 MRS6000	Not active: 0 Active: 1 Output not used: 2	Not active: 0 Active: 1 Output not used: 2
	Out4 count	Uint_32	4		0 … FFFFFFFFh	00 00 00 00 … FF FF FF FF
	Out5 state	Enum_8	1		Not active: 0 Active: 1 Output not used: 2	Not active: 0 Active: 1 Output not used: 2
	Out5 count	Uint_32	4		0 … FFFFFFFFh	00 00 00 00 … FF FF FF FF
	Out6 state	Enum_8	1		Not active: 0 Active: 1 Output not used: 2	Not active: 0 Active: 1 Output not used: 2
	Out6 count	Uint_32	4		0 … FFFFFFFFh	00 00 00 00 … FF FF FF FF
	Out7 state	Enum_8	1	MRS1000 MRS6000	Not active: 0 Active: 1 Output not used: 2	Not active: 0 Active: 1 Output not used: 2
	Out7 count	Uint_32	4		0 … FFFFFFFFh	00 00 00 00 … FF FF FF FF
	Out8 state	Enum_8	1		Not active: 0 Active: 1 Output not used: 2	Not active: 0 Active: 1 Output not used: 2
	Out8 count	Uint_32	4		0 … FFFFFFFFh	00 00 00 00 … FF FF FF FF
	Ext.Out1 state	Enum_8	1	LMS1xx	Not active: 0	Not active: 0

Telegram structure: sRA/sSN LIDoutputstate						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
				LMS5xx	Active: 1 Output not used: 2	Active: 1 Output not used: 2
	Ext.Out1 count	Uint_32	4		0 ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
	Ext.Out2 state	Enum_8	1		Not active: 0 Active: 1 Output not used: 2	Not active: 0 Active: 1 Output not used: 2
	Ext.Out2 count	Uint_32	4		0 ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
	Ext.Out3 state	Enum_8	1		Not active: 0 Active: 1 Output not used: 2	Not active: 0 Active: 1 Output not used: 2
	Ext.Out3 count	Uint_32	4		0 ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
	Ext.Out4 state	Enum_8	1		Not active: 0 Active: 1 Output not used: 2	Not active: 0 Active: 1 Output not used: 2
	Ext.Out4 count	Uint_32	4		0 ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
	Ext.Out5 state	Enum_8	1		Not active: 0 Active: 1 Output not used: 2	Not active: 0 Active: 1 Output not used: 2
	Ext.Out5 count	Uint_32	4		0 ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
	Ext.Out6 state	Enum_8	1		Not active: 0 Active: 1 Output not used: 2	Not active: 0 Active: 1 Output not used: 2
	Ext.Out6 count	Uint_32	4		0 ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
	Ext.Out7 state	Enum_8	1		Not active: 0 Active: 1 Output not used: 2	Not active: 0 Active: 1 Output not used: 2
	Ext.Out7 count	Uint_32	4		0 ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
	Ext.Out8 state	Enum_8	1		Not active: 0 Active: 1 Output not used: 2	Not active: 0 Active: 1 Output not used: 2
	Ext.Out8 count	Uint_32	4		0 ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
Time	States code	Enum_8	1	All	No time data: 0 Time data: 1	No time data: 00 Time data: 01

Telegram structure: sRA/sSN LIDoutputstate						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Time Block (sensor-time from the last change of min. one of the outputs)	Year	Array	2	LMS1xx	e.g. 1970	e.g. 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 254: Telegram structure: sRA/sSN LIDoutputstate

Example: sRA LIDoutputstate

CoLa A	ASCII	<STX>sRA{SPC}LIDoutputstate{SPC}1{SPC}41F84EC5{SPC}1{SPC}5{SPC}1{SPC}5{SPC}1{SPC}5{SPC}2{SPC}0{SPC}2{SPC}0{SPC}2{SPC}0{SPC}2{SPC}0{SPC}2{SPC}0{SPC}2{SPC}0{SPC}2{SPC}0{SPC}2{SPC}0{SPC}1{SPC}7D9{SPC}2{SPC}12{SPC}C{SPC}29{SPC}E{SPC}975E0<ETX>
	Hex	02 73 52 41 20 4C 49 44 6F 75 74 70 75 74 73 74 61 74 65 20 31 20 41 F8 4E C5 20 31 20 35 20 31 20 35 20 31 20 35 20 32 20 30 20 32 20 30 20 32 20 30 20 32 20 30 20 32 20 30 20 32 20 30 20 32 20 30 20 32 20 30 20 31 20 07 D9 20 02 20 12 20 0C 20 29 20 0E 20 09 75 E0 03
CoLa B	Binary	02 02 02 02 00 00 00 5D 73 52 41 20 4C 49 44 6F 75 74 70 75 74 73 74 61 74 65 20 00 01 41 F8 4E C5 01 00 00 00 05 01 00 00 00 05 01 00 00 00 05 02 00 00 00 00 02 00 00 00 00 02 00 00 00 00 02 00 00 00 00 02 00 00 00 00 02 00 00 00 00 02 00 00 00 00 02 00 00 00 00 02 00 00 00 00 02 00 00 00 00 02 00 00 00 00 00 01 07 D9 02 12 0C 29 0E 00 09 75 E0 06

Table 255: Example: sRA LIDoutputstate

4.7.4 Set output state

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

Telegram structure: sMN mDOSetOutput						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3	All	sMN	73 4D 4E
Command	Set output state	String	12	All	mDOSetOutput	6D 44 4F 53 65 74 4F 75 74 70 75 74
Output number		Uint_8	1	LMS1xx	1 ... 3	01 ... 03
				LMS12x	1 ... 2	01 ... 02
				LMS 4000	1 ... 4	01 ... 04
				LMS5xx	1 ... 6	01 ... 06
				TiMxxx	1	01
Output state		Enum_8	1	All	Inactive: 0 Active: 1	Inactive: 00 Active: 01

Table 256: Telegram structure: sMN mDOSetOutput

Example: sMN mDOSetOutput

CoLa A	ASCII	<STX>sMN{SPC}mDOSetOutput{SPC}1{SPC}1<ETX>
	Hex	02 73 4D 4E 20 6D 44 4F 53 65 74 4F 75 74 70 75 74 20 31 20 31 03
CoLa B	Binary	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 69

Table 257: Example: sMN mDOSetOutput

Telegram structure: sAN mDOSetOutput						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sAN	73 41 4E
Command	Set output state	String	12	All	mDOSetOutput	6D 44 4F 53 65 74 4F 75 74 70 75 74
Status Code	Status code	Bool_1	1	All	Error: 0 Success: 1	Error: 00 Success: 01

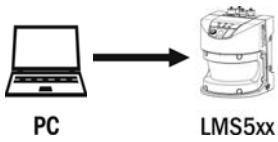
Table 258: Telegram structure: sAN mDOSetOutput

Example: sAN mDOSetOutput

CoLa A	ASCII	<STX>sAN{SPC}mDOSetOutput{SPC}1<ETX>
	Hex	02 73 41 4E 20 6D 44 4F 53 65 74 4F 75 74 70 75 74 20 31 03
CoLa B	Binary	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 67

Table 259: Example: sAN mDOSetOutput

4.7.5 Change output 6/3 function



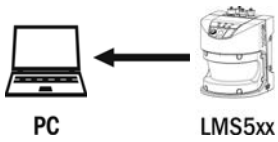
Telegram structure: sWN DO6Fnc/sWN DO3Fnc (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Output function	String	6	LMS5xx PRO	DO6Fnc	44 4F 36 46 6E 63
				LMS5xx Lite	DO3Fnc	44 4F 33 46 6E 63
Output state		Enum_8	1	All	No Function: 0 SOPAS command: 1 Device Ready: 2 Application: 3 Applic./Device Ready: 4 Dev.ready/Contamination: 5 Contamination: 6 Master Synchronisation: 7	Not available

Table 260: Telegram structure PRO: sWN DO6Fnc/Lite: sWN DO3Fnc

Example: sWN DO6Fnc ➜ Set Out6 to Master Synchronisation

CoLa A	ASCII	<STX>sWN{SPC}DO6Fnc{SPC}7<ETX>
	Hex	02 73 57 4E 20 44 4F 36 46 6E 63 20 37 03
CoLa B	Binary	Unavailable with current firmware.

Table 261: Example: sWN DO6Fnc ➜ Out6 to master sync



Telegram structure: sWA DO6Fnc						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command	Output function	String	6	LMS5xx PRO	DO6Fnc	44 4F 36 46 6E 63
				LMS5xx Lite	DO3Fnc	44 4F 33 46 6E 63

Table 262: Telegram structure: PRO: sWN DO6Fnc/Lite: sWN DO3Fnc

Example: sWA DO6Fnc

CoLa A	ASCII	<STX>sWA{SPC}DO6Fnc<ETX>
	Hex	02 73 57 41 20 44 4F 36 46 6E 63 03
CoLa B	Binary	Not available with firmware V1.10

Table 263: Example: sWA DO6Fnc

4.7.6 Change output 1 function

Telegram structure: sWN DO1Fnc (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Output function	String	6	All	DO1Fnc	44 4F 31 46 6E 63
Output 1 function	Selected function	Enum_8	1	All	No function: 0 Command: 1 Device ready: 2 Application dev. ready: 3 Sync pulse: 4 Sync index: 5	No function: 00 Command: 01 Device ready: 02 Application dev. ready: 03 Sync pulse: 04 Sync index: 05

Table 264: Telegram structure: sWN DO1Fnc

Example: sWN DO1Fnc ➜ Set Out1 to Device Ready

CoLa A	ASCII	<STX>sWN{SPC}DO 1Fnc{SPC}2<ETX>
	Hex	02 73 57 4E 20 44 4F 31 46 6E 63 20 32 03
CoLa B	Binary	02 02 02 02 00 00 00 0C 73 57 4E 20 44 4F 31 46 6E 63 20 02 19

Table 265: Example: sWN DO1Fnc ➜ Out1 to device ready

Telegram structure: sWA DO1Fnc						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command	Output function	String	6	All	DO1Fnc	44 4F 31 46 6E 63

Table 266: Telegram structure: sWA DO1Fnc

Example: sWA DO1Fnc

CoLa A	ASCII	<STX>sWA{SPC}DO1Fnc<ETX>
	Hex	02 73 57 41 20 44 4F 31 46 6E 63 03
CoLa B	Binary	02 02 02 02 00 00 00 0A 73 57 41 20 44 4F 31 46 6E 63 34

Table 267: Example: sWA DO1Fnc

Functions:

No function: 0

Command: 1

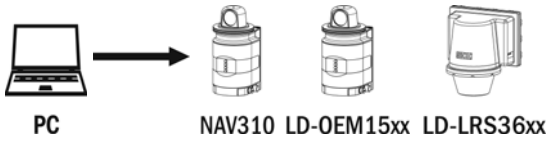
Device ready (for field application): 2

Application dev. ready: 3

Sync pulse (10 ms puls when timer register is read “sRN STIms”): 4

Sync index: 5
The output signal depends on the scanner head position
(high (+24 V): 0° ... 179°/low (0 V): 180° ... 360°).

4.7.7 Change output 1 logic state



Telegram structure: sWN DO1Logic (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Output function	String	8	All	DO1Logic	44 4F 31 4C 6F 67 69 63
Output 1 logic state	State of the output	Enum_8	1	All	Active_High: 0 Active_Low: 1	Active_High: 00 Active_Low: 01

Table 268: Telegram structure: sWN DO1Logic

Example: sWN DO1Logic ➜ Active_High

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

Table 269: Example: sWN DO1Logic ➜ Active_Low

Telegram structure: sWA DO1Logic						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command	Output logic	String	8	All	DO1Logic	44 4F 31 4C 6F 67 69 63

Table 270: Telegram structure: sWA DO1Logic

Example: sWA DO1Logic

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

Table 271: Example: sWA DO1Logic

4.7.8 Change output 2 function

Telegram structure: sWN DO2Fnc (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Output function	String	6	All	DO2Fnc	44 4F 32 46 6E 63
Output 2 function	Code number	Enum_8	1	All	No function: 0 Command: 1 Device ready: 2 Application dev. ready: 3	No function: 00 Command: 01 Device ready: 02 Application dev. ready: 03

Table 272: Telegram structure: sWN DO2Fnc

Example: sWN DO2Fnc ➜ Out2 to device ready

CoLa A	ASCII	<STX>sWN{SPC}DO2Fnc{SPC}2<ETX>
	Hex	02 73 57 4E 20 44 4F 32 46 6E 63 20 32 03
CoLa B	Binary	02 02 02 02 00 00 00 0C 73 57 4E 20 44 4F 32 46 6E 63 20 02 1A

Table 273: Example: sWN DO2Fnc ➜ Out2 to device ready

Telegram structure: sWA DO2Fnc						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command	Output function	String	6	All	DO2Fnc	44 4F 32 46 6E 63

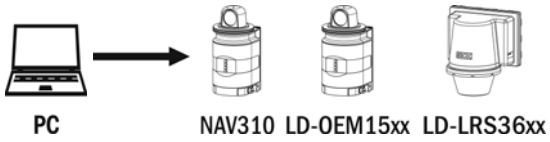
Table 274: Telegram structure: sWA DO2Fnc

Example: sWA DO2Fnc

CoLa A	ASCII	<STX>sWA{SPC}DO2Fnc<ETX>
	Hex	02 73 57 41 20 44 4F 32 46 6E 63 03
CoLa B	Binary	02 02 02 02 00 00 00 0A 73 57 41 20 44 4F 32 46 6E 63 37

Table 275: Example: sWA DO2Fnc

4.7.9 Change output 2 logic state



Telegram structure: sWN DO2Logic (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Output function	String	8	All	DO2Logic	44 4F 32 4C 6F 67 69 63
Output 2 logic state	State of the output	Enum_8	1	All	Active_High: 0 Active_Low: 1	Active_High: 00 Active_Low: 01

Table 276: Telegram structure: sWN DO2Logic

Example: sWN DO2Logic ➜ Active_High

CoLa A	ASCII	<STX>sWN{SPC}DO2Logic{SPC}0<ETX>
	Hex	02 73 57 4E 20 44 4F 32 4C 6F 67 69 63 20 30 03
CoLa B	Binary	02 02 02 02 00 00 00 0E 73 57 4E 20 44 4F 32 4C 6F 67 69 63 20 00 1C

Table 277: Example: sWN DO2Logic ➜ Active_High

Telegram structure: sWA DO2Logic						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command	Output logic	String	8	All	DO2Logic	44 4F 32 4C 6F 67 69 63

Table 278: Telegram structure: sWA DO2Logic

Example: sWA DO2Logic

CoLa A	ASCII	<STX>sWA{SPC}DO2Logic<ETX>
	Hex	02 73 57 41 20 44 4F 32 4C 6F 67 69 63 03
CoLa B	Binary	02 02 02 02 00 00 00 0C 73 57 41 20 44 4F 32 4C 6F 67 69 63 32

Table 279: Example: sWA DO2Logic

4.7.10 Set synchronization mode

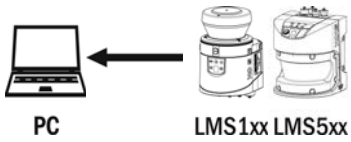
Telegram structure: sWN SYMode (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Set sync mode	String	6	All	SYMode	53 59 4D 6F 64 65
Sync mode data	Synchronization mode data	Bool_1	1	All	No sync = 0 Sync by wire = 1 Sync by CAN = 2	Not possible

Table 280: Telegram structure: sWN SYMode

Example: sWN SYMode

CoLa A	ASCII	<STX>sWN{SPC}SYMode{SPC}1<ETX>
	Hex	02 73 57 4E 20 53 59 4D 6F 64 65 20 31 03
CoLa B	Binary	Not possible

Table 281: Example: sWN SYMode



Telegram structure: sWA SYMode						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command	Set sync mode	String	6	All	SYMode	53 59 4D 6F 64 65

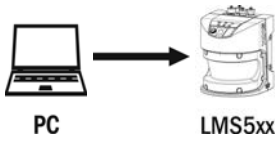
Table 282: Telegram structure: sWA SYMode

Example: sWA SYMode

CoLa A	ASCII	<STX>sWA{SPC}SYMode<ETX>
	Hex	02 73 57 41 20 53 59 4D 6F 64 65 03
CoLa B	Binary	Not possible

Table 283: Example: sWA SYMode

4.7.11 Set synchronization phase



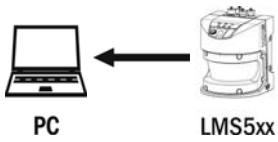
Telegram structure: sWN SYPhase (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	Not possible
Command	Set sync phase	String	7	All	SYPhase	Not possible
Sync phase data	Synchronization phase data	Int_16	2	All	-180d ... +180d (FF4Ch ... 00B4h)	Not possible

Table 284: Telegram structure: sWN SYPhase

Example: sWN SYPhase +90

CoLa A	ASCII	<STX>sWN{SPC}SYPhase{SPC}+90<ETX>
	Hex	02 73 57 4E 20 53 59 50 68 61 73 65 20 2B 39 30 03
CoLa B	Binary	Not possible

Table 285: Example: sWN SYPhase +90



Telegram structure: sWA SYPhase						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	Not possible
Command	Set sync phase	String	7	All	SYPhase	Not possible

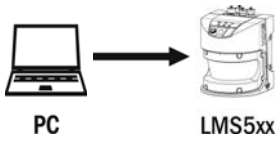
Table 286: Telegram structure: sWA SYPhase

Example: sWA SYPhase

CoLa A	ASCII	<STX>sWA{SPC}SYPhase<ETX>
	Hex	02 73 57 41 20 53 59 50 68 61 73 65 03
CoLa B	Binary	Not possible

Table 287: Example: sWA SYPhase

4.7.12 Change input 4 function



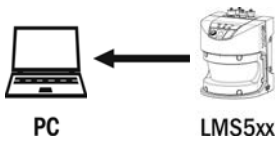
Telegram structure: sWN DO3And4Fnc (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Input function	String	10	All	DO3And4Fnc	44 4F 33 41 6E 64 34 46 6E 63
Input state	Code number	Enum_8	1	All	No function: 0 Encoder: 1 Slave sync: 2 Digital input: 3	

Table 288: Telegram structure: sWN DO3And4Fnc

Example: sWN In4 ➜ In3+4 to slave sync

CoLa A	ASCII	<STX>sWN{SPC}DO3And4Fnc{SPC}2<ETX>
	Hex	02 73 57 4E 20 44 4F 33 41 6E 64 34 46 6E 63 20 02 03
CoLa B	Binary	Not available with firmware V1.10

Table 289: Example: sWN In4 ➜ In3+4 to slave sync



Telegram structure: sWA DO3And4Fnc						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command	Input function	String	10	All	DO3And4Fnc	44 4F 33 41 6E 64 34 46 6E 63

Table 290: Telegram structure: sWA DO3And4Fnc

Example: sWA DO3And4Fnc

CoLa A	ASCII	<STX>sWA{SPC}DO3And4Fnc<ETX>
	Hex	02 73 57 41 20 44 4F 33 41 6E 64 34 46 6E 63 03
CoLa B	Binary	Not available with firmware V1.10

Table 291: Example: sWA DO3And4Fnc

4.7.13 Set debouncing time for input x

The telegram applies for the inputs 1 to 4 (DIxDebTim, x = 1 ... 4). The following tables show the data for input 3.

Telegram structure: sWN DI3DebTim (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Set debouncing time for input 3	String	9	All	DI3DebTim	44 49 33 44 65 62 54 69 6D
Debouncing time data	[ms]	Uint_16	2	All	0d ... +10000d (00h ... 2710h)	00 00 ... 27 10

Table 292: Telegram structure: sWN DI3DebTim

Example: sWN DI3DebTim

CoLa A	ASCII	<STX>sWN{SPC}DI3DebTim{SPC}+10<ETX>
	Hex	02 73 57 4E 20 44 49 33 44 65 62 54 69 6D 20 2B 31 30 03
CoLa B	Binary	02 02 02 02 00 00 00 10 73 57 4E 20 44 49 33 44 65 62 54 69 6D 20 00 0A 77

Table 293: Example: sWN DI3DebTim

Telegram structure: sWA DI3DebTim						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command	Set debouncing time for input 3	String	9	All	DI3DebTim	44 49 33 44 65 62 54 69 6D

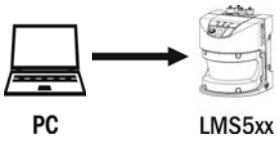
Table 294: Telegram structure: sWA DI3DebTim

Example: sWA DI3DebTim

CoLa A	ASCII	<STX>sWA{SPC}DI3DebTim<ETX>
	Hex	02 73 57 4E 20 44 49 33 44 65 62 54 69 6D 03
CoLa B	Binary	02 02 02 02 00 00 00 0E 73 57 41 20 44 49 33 44 65 62 54 69 6D 20 48

Table 295: Example: sWA DI3DebTim

4.7.14 Read status of external sync signal



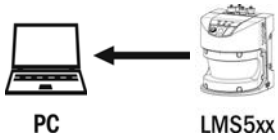
Telegram structure: sRN SYextmon						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sRN	73 52 4E
Command	Status of external sync signal	String	8	All	SYextmon	53 59 65 78 74 6D 6F 6E

Table 296: Telegram structure: sRN SYextmon

Example: sRN SYextmon

CoLa A	ASCII	<STX>sRN{SPC}SYextmon<ETX>
	Hex	02 73 52 4E 20 53 59 65 78 74 6D 6F 6E 03
CoLa B	Binary	02 02 02 02 00 00 00 0C 73 52 4E 20 53 59 65 78 74 6D 6F 6E 40

Table 297: Example: sRN SYextmon



Telegram structure: sRA SYextmon						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sRA	73 52 41
Command	Status of external sync signal	String	8	All	SYextmon	53 59 65 78 74 6D 6F 6E
Sync status data	Synchronization status data	Uint_8	1	All	None: 1 Too slow: 2 Good: 4 Too fast: 8	None: 01 Too slow: 02 Good: 04 Too fast: 08
Signal frequency	[Frequency in Hz as float according to IEEE754]	Real	4	All	0h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF

Table 298: Telegram structure: sRA SYextmon

Example: sRA SYextmon (49.9 Hz)

CoLa A	ASCII	<STX>sRA{SPC}SYextmon{SPC}4{SPC}4247BD87<ETX>
	Hex	02 73 52 41 20 53 59 65 78 74 6D 6F 6E 20 04 42 47 BD 87 03
CoLa B	Binary	02 02 02 02 00 00 00 12 73 52 41 20 53 59 65 78 74 6D 6F 6E 20 04 42 47 BD 87 54

Table 299: Example: sRA SYextmon

4.8 Status

4.8.1 Read contamination status of the LMS

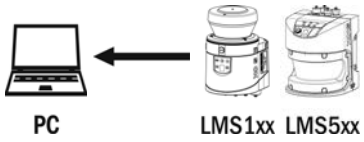
Telegram structure: sRN LCMstate						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sRN	73 52 4E
Command	Status of LMS	String	8	All	LCMstate	4C 43 4D 73 74 61 74 65

Table 300: Telegram structure: sRN LCMstate

Example: sRN LCMstate

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

Table 301: Example: sRN LCMstate



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

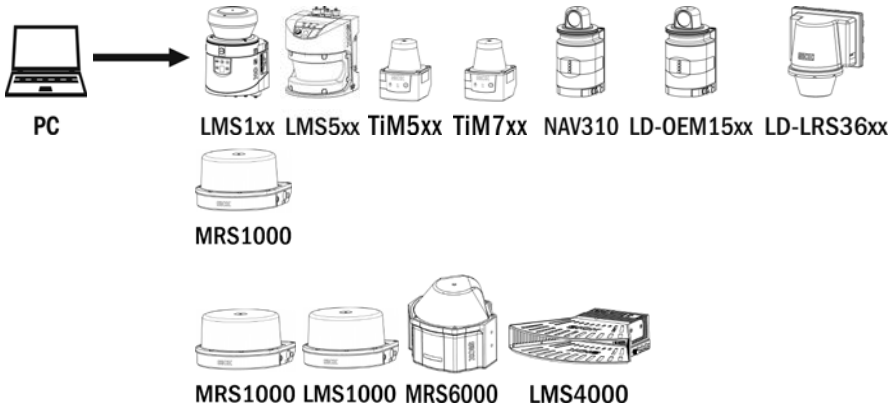
Table 302: Telegram structure: sRA LCMstate

Example for LMS100: sRA LCMstate

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

Table 303: Example for LMS100: sRA LCMstate

4.8.2 Read firmware version



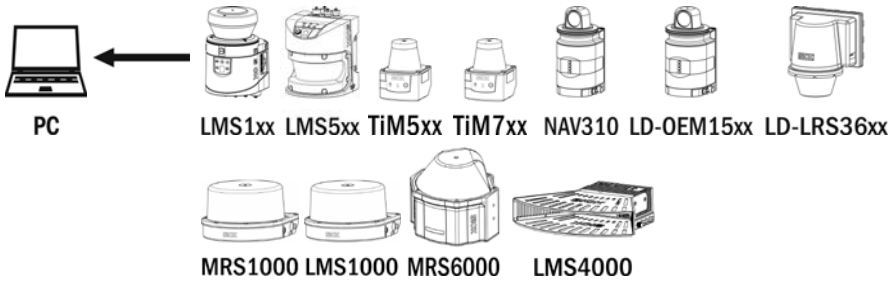
Telegram structure: sRN DeviceIdent						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sRN	73 52 4E
Command	Read firmware version	String	11	All	DeviceIdent	44 65 76 69 63 65 49 64 65 6E 74

Table 304: Telegram structure: sRN DeviceIdent

Example: sRN DeviceIdent

CoLa A	ASCII	<STX>sRN{SPC}DeviceIdent<ETX>
	Hex	02 73 52 4E 20 44 65 76 69 63 65 49 64 65 6E 74 03
CoLa B	Binary	02 02 02 02 00 00 00 0F 73 52 4E 20 44 65 76 69 63 65 49 64 65 6E 74 25

Table 305: Example: sRN DeviceIdent



Telegram structure: sRA DeviceIdent						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sRA	73 52 41
Command		String	11	All	DeviceIdent	44 65 76 69 63 65 49 64 65 6E 74
Value	Length of name of firmware version	Enum_16	1	All	0 ... 22h	0 ... 22h
Value	Name of firmware version	String		All	(See example)	(See example)
Value	Length of number of firmware version	Enum_16	1	All	0 ... 22h	0 ... 22h
Value	Number of firmware version	String		All	(See example)	(See example)

Table 306: Telegram structure: sRA DeviceIdent

Example: sRA DeviceIdent

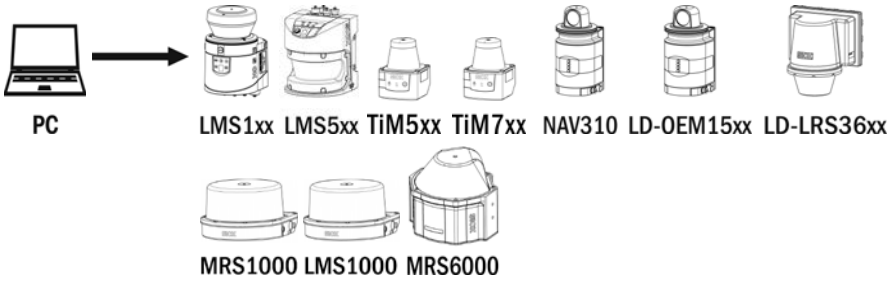
CoLa A	ASCII	<STX>sRA{SPC}DeviceIdent{SPC}10{SPC}LMS10x_FieldEval{SPC}10{SPC}V1.36-21.10.2010<ETX>
	Hex	Always ASCII answer
CoLa B	Binary	02 02 02 02 00 00 00 34 73 52 41 20 44 65 76 69 63 65 49 64 65 6E 74 20 00 10 4C 4D 53 31 30 78 5F 46 69 65 6C 64 45 76 61 6C 00 10 56 31 2E 33 36 2D 32 31 2E 31 30 2E 32 30 31 30 62

Table 307: Example: sRA DeviceIdent

4.8.3 Read the device state

This telegram reads the general device state.

Remark: The status of the measurement function of LMS1 and LMS5 can be read separately with the telegram STIms



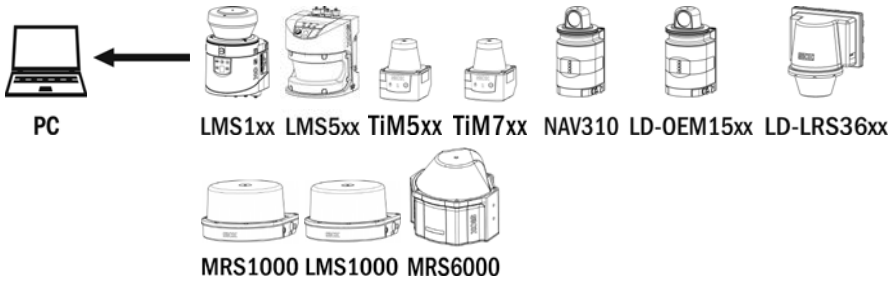
Telegram structure: sRN SCdevicestate						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sRN	73 52 4E
Command		String	13	All	SCdevicestate	53 43 64 65 76 69 63 65 73 74 61 74 65

Table 308: Telegram structure: sRN SCdevicestate

Example: sRN SCdevicestate

CoLa A	ASCII	<STX>sRN{SPC}SCdevicestate<ETX>
	Hex	02 73 52 4E 20 53 43 64 65 76 69 63 65 73 74 61 74 65 03
CoLa B	Binary	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

Table 309: Example: sRN SCdevicestate



Telegram structure: sRA SCdevicestate						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sRA	73 52 41
Command		String	13	All	SCdevicestate	53 43 64 65 76 69 63 65 73 74 61 74 65
Status code	Code number	Enum_8	1	LMS1xx NAV310 LD-OEM 15xx LD-LRS 36xx MRS6000	Busy: 0 Ready: 1 Error: 2	Busy: 00 Ready: 01 Error: 02
				LMS5xx TiMxxx MRS1000 LMS1000	Busy: 0 Ready: 1 Error: 2 Standby: 3	Busy: 00 Ready: 01 Error: 02 Standby: 03

Table 310: Telegram structure: sRA SCdevicestate

Example: sRA SCdevicestate

CoLa A	ASCII	<STX>sRA{SPC}SCdevicestate{SPC}0<ETX>
	Hex	02 73 52 41 20 53 43 64 65 76 69 63 65 73 74 61 74 65 20 00 03
CoLa B	Binary	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 00 1F

Table 311: Example: sRA SCdevicestate

4.8.4 Status commands for LD-XXX and NAV310

The following status commands will be explained in the subsequent sections:

- LMCmeasstate: Status of the internal Statemachine
- SCdevicestate: Status of the Sensors (actual measurement status)
- EMCustomerInfo: Additional error information
- LDMSenStat: Status of the state machine of the measurment core, Motor status

How status commands for for LD-XXX and NAV310 work together:

- If LMCmeasstate changes to “Idle” or an other status, although the measurement status “Measure2D” is expected, there is an error during the measurement (or during start up of the measurement).
- SCdevicestate is always "Ready", if the measurement is active.
  If “Busy” will be indicated the unit is not measuring (e.g, IDLE). If there is any failure “Error” will be indicated. (However LMCmeasstate could indicate “Measure2D”, if the failure occurs during the measurement, because it is only an indication of the status of the State machine).
- In case of a failure EMCustomerInfo can provide an information about the error.
  In case of an motor failure there are following condition visible:
  - Motor blocked during operation ➜ DEVICE_FAILURE
  - Motor blocked during spin up ➜ CHECK_PARAMETER
- It is also possible to read LDMSenStat (and to register as an event). This value equals the Sensorstatus of the NAV310/LD-XXX. A status “B1” of the measurement core means “Motor error and Idle”).
- During the measurement it is possible to monitor a deviation of the target rotation frequency. (If the device detects rotation values that are too slow, it will terminate the measurement.)

In case of an failure this value will not always be updated, therefore it is necessary to monitor LMCmeasstate and SCdevicestate in parallel.

NOTE

- In case of an failure (Scanner does not change to MEASURE2D or switches back to IDLE), it is necessary to send the command LMCstopmeas (even if the Status is indicated as IDLE)
- If at EMCustomerInfo the message CHECK_PARAMETER is indicated, a reset is only possible by a power cycle of the scanner.

Ask for Device Measurement State

Telegram structure: sRN LMCmeasstate						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sRN	73 52 4E
Command	Ask for measurement state	String	12	All	LMCmeasstate	4C 4D 43 6D 65 61 73 73 74 61 74 65

Table 312: Telegram structure: sRN LMCmeasstate

Example: sRN LMCmeasstate

CoLa A	ASCII	<STX>sRN{SPC}LMCmeasstate<ETX>
	Hex	02 73 52 4E 20 4C 4D 43 6D 65 61 73 73 74 61 74 65 03
CoLa B	Binary	02 02 02 02 00 00 00 11 73 52 4E 20 4C 4D 43 6D 65 61 73 73 74 61 74 65 30

Table 313: Example: sRN LMCmeasstate

Telegram structure: sRA LMCmeasstate						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sRA	73 52 41
Command	Report measurement state	String	12	All	LMCmeasstate	4C 4D 43 6D 65 61 73 73 74 61 74 65
Status code	Current measurement state	Enum_16	2	All	Idle: 3 Ready 2D: 6 Measure 2D: 7 Other state codes may show up during booting, firmware update or transition between states.	Idle: 0003 Ready 2D: 0006 Measure 2D: 0007

Table 314: Telegram structure: sRA LMCmeasstate

Example: sRA LMCmeasstate is Measure 2D

CoLa A	ASCII	<STX>sRA{SPC}LMCmeasstate{SPC}7<ETX>
	Hex	02 73 52 41 20 4C 4D 43 6D 65 61 73 73 74 61 74 65 20 00 07 03
CoLa B	Binary	02 02 02 02 00 00 00 13 73 52 41 20 4C 4D 43 6D 65 61 73 73 74 61 74 65 20 00 07 1F

Table 315: Example: sRA LMCmeasstate is Measure 2D

Ask for customer info of sensor

This telegram will provide additional error information.

Telegram structure: sRN EMCustomerInfo						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sRN	73 52 4E
Command	Ask for customer info	String	14	All	EMCustomerInfo	45 4D 43 75 73 74 6F 6D 65 72 49 6E 66 6F

Table 316: Telegram structure: sRN EMCustomerInfo

Example: sRN EMCustomerInfo

CoLa A	ASCII	<STX>sRN{SPC}EMCustomerInfo<ETX>
	Hex	02 73 52 4E 20 45 4D 43 75 73 74 6F 6D 65 72 49 6E 66 6F 03
CoLa B	Binary	02 02 02 02 00 00 00 12 73 52 4E 20 45 4D 43 75 73 74 6F 6D 65 72 49 6E 66 6F 4D

Table 317: Example: sRN EMCustomerInfo

Telegram structure: sRA EMCustomerInfo						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sRA	73 52 41
Command	Report customer info	String	14	All	EMCustomerInfo	45 4D 43 75 73 74 6F 6D 65 72 49 6E 66 6F
Status code	Customer info	Enum_16	2	All	0: DEVICE_OK 1: DEFECTIVE_DEVICE 2: DEVICE_TEMP_FAILURE 3: DEVICE_FAILURE 4: DEVICE_NOT_READY 5: CHECK_PARAMETER	0000: DEVICE_OK 0001: DEFECTIVE_DEVICE 0002: DEVICE_TEMP_FAILURE 0003: DEVICE_FAILURE 0004: DEVICE_NOT_READY 0005: CHECK_PARAMETER
					DEFECTIVE_DEVICE: Please return device to SICK DEVICE_TEMP_FAILURE D: Device failure. Please check temperature. DEVICE_FAILURE : Please switch off for 20 seconds and power up again. DEVICE_NOT_READY: Please wait. CHECK_PARAMETER: Warning – please check parametrization.	

Table 318: Telegram structure: sRA EMCustomerInfo

Example: sRA EMCustomerInfo = Device OK

CoLa A	ASCII	<STX>sRA{SPC}EMCustomerInfo{SPC}0<ETX>
	Hex	02 73 52 41 20 45 4D 43 75 73 74 6F 6D 65 72 49 6E 66 6F 20 00 03
CoLa B	Binary	02 02 02 02 00 00 00 14 73 52 4E 20 45 4D 43 75 73 74 6F 6D 65 72 49 6E 66 6F 20 00 6D

Table 319: Example: sRA EMCustomerInfo = Device OK

Ask for Sensorstatus

This telegram provides status information of the State Machine of measurement core and the Motor Status

Telegram structure: sRN LDMSenStat						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sRN	73 52 4E
Command	Ask for state	String	10	All	LDMSenStat	4C 44 4D 53 65 6E 53 74 61 74

Table 320: Telegram structure: sRN LDMSenStat

Example: sRN LDMSenStat

CoLa A	ASCII	<STX>sRN{SPC}LDMSenStat<ETX>
	Hex	02 73 52 4E 20 4C 44 4D 53 65 6E 53 74 61 74 03
CoLa B	Binary	02 02 02 02 00 00 00 0E 73 52 4E 20 4C 44 4D 53 65 6E 53 74 61 74 60

Table 321: Example: sRN LDMSenStat

Telegram structure: sRA LDMSenStat						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sRA	73 52 41
Command	Report state	String	10	All	LDMSenStat	4C 44 4D 53 65 6E 53 74 61 74

Telegram structure: sRA LDMSenStat						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Status code	Current state regarding ...	Uint_32	4	All	Idle: 3 Ready 2D: 6 Measure 2D: 7 Other state codes may show up during booting, firmware update or transition between states.	Idle: 0003 Ready 2D: 0006 Measure 2D: 0007
	Working mode		Bit 0...3		Idle: 1 Rotate: 2 Measure: 3 Error: 4 (Other bits: reserved)	Idle: 1 Rotate: 2 Measure: 3 Error: 4 (Other bits: reserved)
	Motor mode		Bit 4...7		Motor ok: 0 Motor spin to low: 4 Motor spin to high: 9 Motor stops or coder error: B (Other bits: reserved)	Motor ok: 0 Motor spin to low: 4 Motor spin to high: 9 Motor stop or coder error: B (Other bits: reserved)
	(Reserved)		Bit 8...31		(Reserved)	(Reserved)

Table 322: Telegram structure: sRA LDMSenStat

Example: sRA LDMSenStat Device in Idle mode

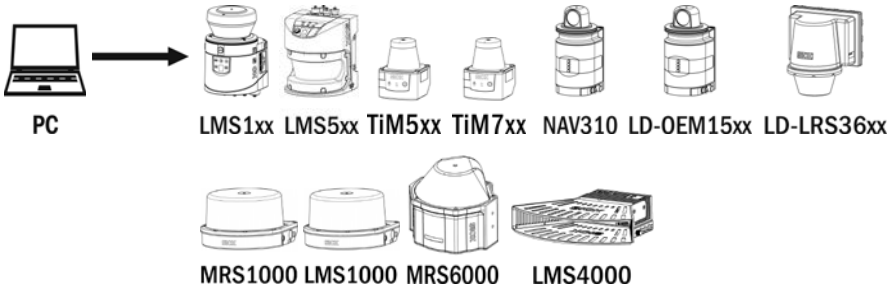
CoLa A	ASCII	<STX>sRA{SPC}LDMSenStat{SPC} 1<ETX>
	Hex	02 73 52 41 20 4C 44 4D 53 65 6E 53 74 61 74 20 31 03
CoLa B	Binary	02 02 02 02 00 00 00 13 73 52 41 20 4C 44 4D 53 65 6E 53 74 61 74 20 00 00 00 01 4E

Table 323: Example: sRA LDMSenStat Device is in Idle mode

4.8.5 Read device order number

Device order number

This telegram reads the device order number.



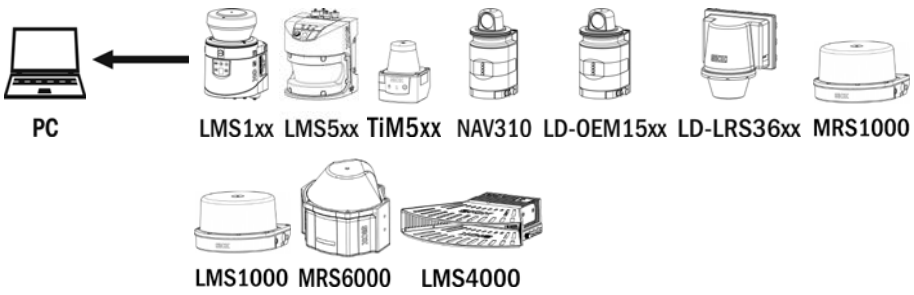
Telegram structure: sRN DIornr						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sRN	73 52 4E
Command	Read state	String	6	All	DIornr	44 49 6F 72 6E 72

Table 324: Telegram structure: sRN DIornr

Example: sRN DIornr

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

Table 325: Example: sRN DIornr



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

Table 326: Telegram structure: sRA DIornr

Example: sRA DIornr 1047782 (Order Number for LMS511-20100)

CoLa A	ASCII	<STX>sRA{SPC}DIornr{SPC}1047782<ETX>
	Hex	02 73 52 41 20 44 49 6F 72 6E 72 20 31 30 34 37 37 38 32 03
CoLa B	Binary	02 02 02 02 00 00 00 12 73 52 41 20 44 49 6F 72 6E 72 20 31 30 34 37 37 38 32 53

Table 327: Example for LMS511-20100: sRA DIornr

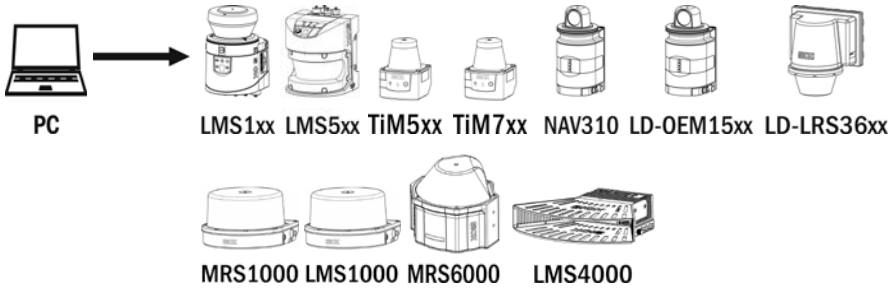
Example: sRA DIornr 1067299 (Order Number for TiM561-2050101)

CoLa A	ASCII	<STX>sRA{SPC}DIornr{SPC}1071419<ETX>
	Hex	02 73 52 41 20 44 49 6F 72 6E 72 20 31 30 37 31 34 31 39 03
CoLa B	Binary	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

Table 328: Example for TiM561-2050101: sRA DIornr

4.8.6 Read device type

This telegram asks for the device type.



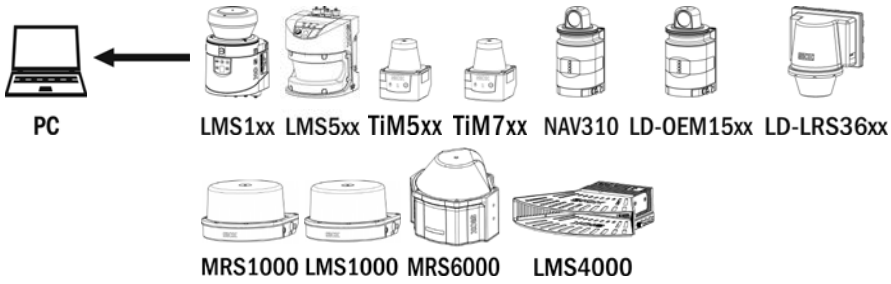
Telegram structure: sRN DItype						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sRN	73 52 4E
Command	Ask state	String	6	All	DItype	44 49 74 79 70 65

Table 329: Telegram structure: sRN DItype

Example: sRN DItype

CoLa A	ASCII	<STX>sRN{SPC}DItype<ETX>
	Hex	02 73 52 4E 20 44 49 74 79 70 65 03
CoLa B	Binary	02 02 02 02 00 00 00 0A 73 52 4E 20 44 49 74 79 70 65 5A

Table 330: Example: sRN DItype



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

Table 331: Telegram structure: sRA DItype

Example for LMS511-20100

CoLa A	ASCII	<STX>sRA{SPC}DItype{SPC}C{SPC}LMS511-20100<ETX>
	Hex	02 73 52 41 20 44 49 74 79 70 65 20 43 20 4C 4D 53 35 31 31 2D 32 30 31 30 30 03
CoLa B	Binary	02 02 02 02 00 00 00 18 73 52 41 20 44 49 74 79 70 65 20 0C 4C 4D 53 35 31 31 2D 32 30 31 30 30 00

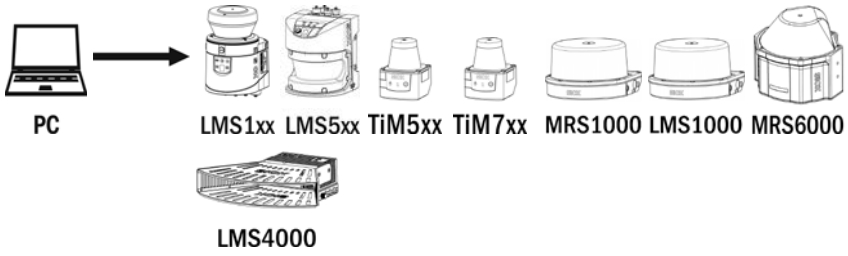
Table 332: Example for LMS511-20100: sRA DItype

Example for TiM561-2050101

CoLa A	ASCII	<STX>sRA{SPC}DItype{SPC}E{SPC}TIM561-2050101<ETX>
	Hex	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	Binary	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

Table 333: Example for TiM561-2050101: sRA DItype

4.8.7 Read operating hours



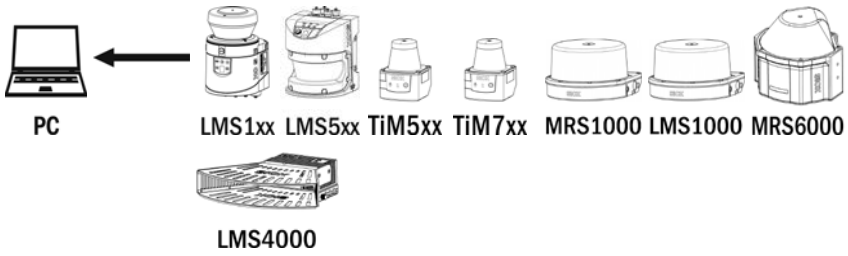
Telegram structure: sRN ODoprh						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sRN	73 52 4E
Command	Read operating hours	String	6	All	ODoprh	4F 44 6F 70 72 68

Table 334: Telegram structure: sRN ODoprh

Example: sRN ODoprh

CoLa A	ASCII	<STX>sRN{SPC}ODoprh<ETX>
	Hex	02 73 52 4E 20 4F 44 6F 70 72 68 03
CoLa B	Binary	02 02 02 02 00 00 00 0A 73 52 4E 20 4F 44 6F 70 72 68 41

Table 335: Example: sRN ODoprh



Telegram structure: sRA ODoprh						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sRA	73 52 41
Command	Read operating hours	String	6	All	ODoprh	4F 44 6F 70 72 68
Value	Operating hours in 1/10 h	Uint_32	4	All	0h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF

Table 336: Telegram structure: sRA ODoprh

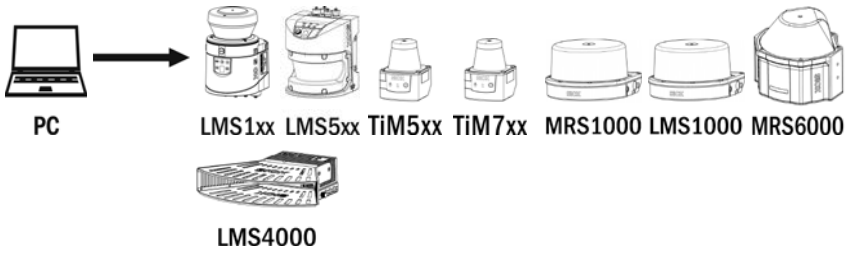
Example: sRA ODoprh

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

Table 337: Example: sRA ODoprh

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

4.8.8 Read power on counter



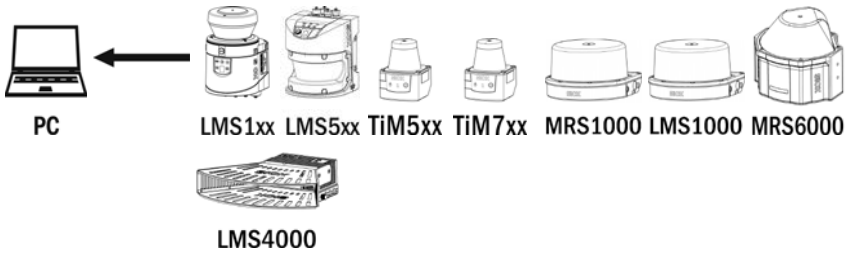
Telegram structure: sRN ODpwrc						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sRN	73 52 4E
Command	Read power on counter	String	6	All	ODpwrc	4F 44 70 77 72 63

Table 338: Telegram structure: sRN ODpwrc

Example: sRN ODpwrc

CoLa A	ASCII	<STX>sRN{SPC}ODpwrc<ETX>
	Hex	02 73 52 4E 20 4F 44 70 77 72 63 03
CoLa B	Binary	02 02 02 02 00 00 00 0A 73 52 4E 20 4F 44 70 77 72 63 52

Table 339: Example: sRN ODpwrc



Telegram structure: sRA ODpwrc						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sRA	73 52 41
Command	Read power on counter	String	6	All	ODpwrc	4F 44 70 77 72 63
Value	Power on counter	Uint_32	4	All	0h ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF

Table 340: Telegram structure: sRA ODpwrc

Example: sRA ODpwrc

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

Table 341: Example: sRA ODpwrc

4.8.9 Read temperature

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.

Telegram structure: sRN OPcurtmpdev						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sRN	73 52 4E
Command	Read temperature of the device	String	11	All	OPcurtmpdev	4F 50 63 75 72 74 6D 70 64 65 76

Table 342: Telegram structure: sRN OPcurtmpdev

Example: sRN OPcurtmpdev

CoLa A	ASCII	<STX>sRN{SPC}OPcurtmpdev<ETX>
	Hex	02 73 52 4E 20 4F 50 63 75 72 74 6D 70 64 65 76 03
CoLa B	Binary	02 02 02 02 00 00 00 0F 73 52 4E 20 4F 50 63 75 72 74 6D 70 64 65 76 2A

Table 343: Example: sRN OPcurtmpdev

Telegram structure: sRA OPcurtmpdev						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sRA	73 52 41
Command	Read temperature of the device	String	11	All	OPcurtmpdev	4F 50 63 75 72 74 6D 70 64 65 76
Tempera-ture data	[°C as float accord-ing to IEEE754]	Real	4	All	C2480000h ... 42C80000h (-50°C ... +100°C)	C2 48 00 00 ... 42 C8 00 00

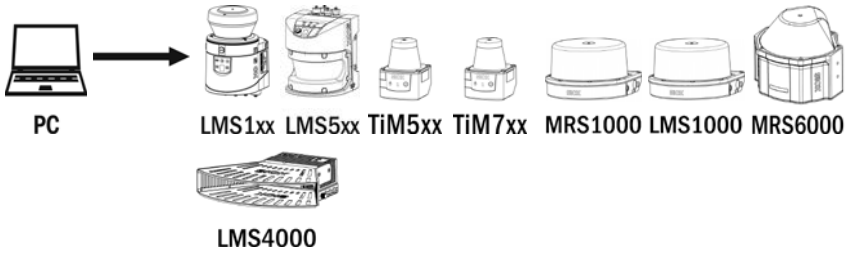
Table 344: Telegram structure: sRA OPcurtmpdev

Example: sRA OPcurtmpdev (-50°C)

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

Table 345: Example: sRA OPcurtmpdev

4.8.10 Set device name



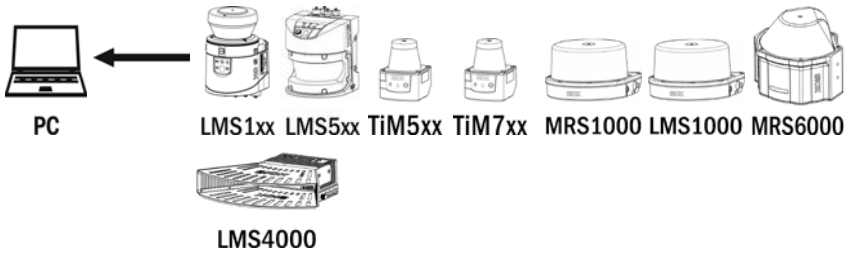
Telegram structure: sWN LocationName (Maintenance)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Set device name	String	12	All	LocationName	4C 6F 63 61 74 69 6F 6E 4E 61 6D 65
Value	Array of characters of the following device name	Uint_16	2	All	0d ... +16d (0h ... 10h)	00 00 ... 00 10
Value	Device name	String	16	All	[Device name]	[Device name]

Table 346: Telegram structure: sWN LocationName

Example: sWN LocationName +13 OutdoorDevice

CoLa A	ASCII	<STX>sWN{SPC}LocationName{SPC}+13{SPC}OutdoorDevice<ETX>
	Hex	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	Binary	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

Table 347: Example: sWN LocationName +13 OutdoorDevice



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

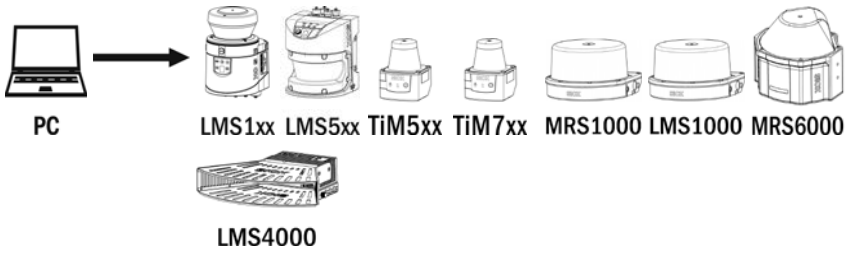
Table 348: Telegram structure: sWA LocationName

Example: sWA LocationName

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

Table 349: Example: sWA LocationName

4.8.11 Read device name



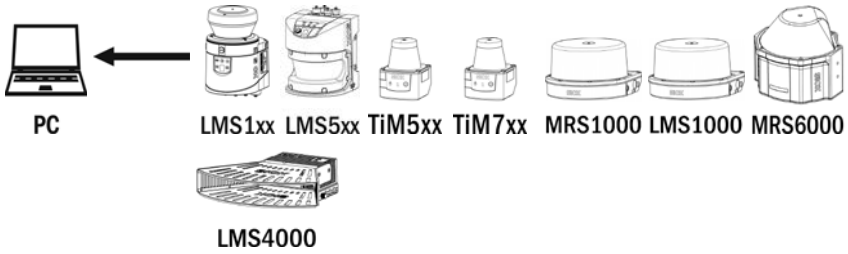
Telegram structure: sRN LocationName						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sRN	73 52 4E
Command	Read device name	String	12	All	LocationName	4C 6F 63 61 74 69 6F 6E 4E 61 6D 65

Table 350: Telegram structure: sRN LocationName

Example: sRN LocationName

CoLa A	ASCII	<STX>sRN{SPC}LocationName<ETX>
	Hex	02 73 52 4E 20 4C 6F 63 61 74 69 6F 6E 4E 61 6D 65 03
CoLa B	Binary	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

Table 351: Example: sRN LocationName



Telegram structure: sRA LocationName						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sRA	73 52 41
Command	Read device name	String	12	All	LocationName	4C 6F 63 61 74 69 6F 6E 4E 61 6D 65
Value	Array of characters of the following device name	Uint_16	2	All	0d ... +16d (0h ... 10h)	00 00 ... 00 10
Value	Device name	String	16	All	[Device name]	[Device name]

Table 352: Telegram structure: sRA LocationName

Example: sRA LocationName

CoLa A	ASCII	<STX>sRA{SPC}LocationName{SPC}D{SPC}OutdoorDevice<ETX>
	Hex	02 73 52 41 20 4C 6F 63 61 74 69 6F 6E 4E 61 6D 65 20 44 20 4F 75 74 64 6F 6F 72 44 65 76 69 63 65 03
CoLa B	Binary	02 02 02 02 00 00 00 17 73 52 41 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 20

Table 353: Example: sRA LocationName

4.8.12 Read angle compensation sine

Telegram structure: sRN MCAngleCompSin						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sRN	73 52 4E
Command	Read angle compensation sine	String	14	All	MCAngleCompSin	4D 43 41 6E 67 6C 65 43 6F 6D 70 53 69 6E

Table 354: Telegram structure: sRN MCAngleCompSin

Example: sRN MCAngleCompSin

CoLa A	ASCII	<STX>sRN{SPC}MCAngleCompSin<ETX>
	Hex	02 73 52 4E 20 4D 43 41 6E 67 6C 65 43 6F 6D 70 53 69 6E 03
CoLa B	Binary	02 02 02 02 00 00 00 12 73 52 4E 20 4D 43 41 6E 67 6C 65 43 6F 6D 70 53 69 6E 65

Table 355: Example: sRN MCAngleCompSin

Telegram structure: sRA MCAngleCompSin						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sRA	73 52 41
Command	Angle compensation sine	String	14	All	MCAngleCompSin	4D 43 41 6E 67 6C 65 43 6F 6D 70 53 69 6E
Amplitude	[1/10000°]	Int_16	2	All	-10000d ... +10000d (D8F0h ... 2710h)	D8 F0 ... 27 10
Phase	[1/10000°]	Int_32	4	All	-3600000d ... +3600000d (FFC91180h ... 36EE80h)	FF C9 11 80 ... 00 36 EE 80
Offset	[1/10000°]	Int_16	2	All	-10000d ... +10000d (D8F0h ... 2710h)	D8 F0 ... 27 10

Table 356: Telegram structure: sRA MCAngleCompSin

Example: sRA MCAngleCompSin

CoLa A	ASCII	<STX>sRA{SPC}MCAngleCompSin{SPC}0{SPC}0{SPC}0<ETX>
	Hex	02 73 52 41 20 4D 43 41 6E 67 6C 65 43 6F 6D 70 53 69 6E 20 30 20 30 20 30 03
CoLa B	Binary	02 02 02 02 00 00 00 18 73 52 41 20 4D 43 41 6E 67 6C 65 43 6F 6D 70 53 69 6E 20 00 00 00 00 00 00 00 00 4A

Table 357: Example: sRA MCAngleCompSin

The values of the angular compensation could be retrieved form the memory of the NAV310 to improve the angular measurement accuracy.

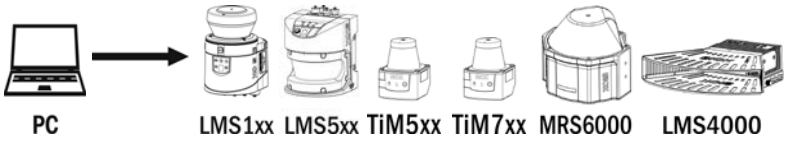
The applied formula is:

AngleComp =AngleRaw + (AngleCompAmp * sin(AngleRaw - AngleCompPhase) + AngleCompOffset

Example (C coded):

```
angleRaw:Raw angle as float in degrees (0.000 ... 359999)
angleComp:Compensated angle as float in degrees (0.000 ... 359999)
AngleCompAmp
AngleCompPhase
AngleCompOffset: Compensation parameters as int in 1/1000 degrees
float compensateAngle(float angleRaw)
{
    float angleComp;
    angleRaw += ((float) AngleCompOffset)/1000.0;
    angleRaw += (((float) AngleCompAmp)/1000.0) *
        sin((DEGTORAD * (angle - ((float) AngleCompPhase)/1000.0)));
    return angleComp;
}
```

4.8.13 Reset output counter



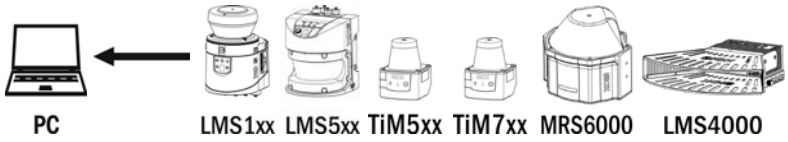
Telegram structure: sMN LIDrstoutpcnt (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3	All	sMN	73 4D 4E
Command	Reset output counter	String	13	All	LIDrstoutpcnt	4C 49 44 72 73 74 6F 75 74 70 63 6E 74

Table 358: Telegram structure: sMN LIDrstoutpcnt

Example: sMN LIDrstoutpcnt

CoLa A	ASCII	<STX>sMN{SPC}LIDrstoutpcnt<ETX>
	Hex	02 73 4D 4E 20 4C 49 44 72 73 74 6F 75 74 70 63 6E 74 03
CoLa B	Binary	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

Table 359: Example: sMN LIDrstoutpcnt



Telegram structure: sAN LIDrstoutpcnt						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sAN	73 41 4E
Command	Reset state	String	13	All	LIDrstoutpcnt	4C 49 44 72 73 74 6F 75 74 70 63 6E 74
Status code	Code number	Bool_1	1	All	Success: 0 Error: 1	Success: 00 Error: 01

Table 360: Telegram structure: sAN LIDrstoutpcnt

Example: sAN LIDrstoutpcnt

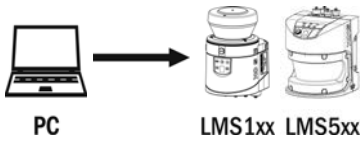
CoLa A	ASCII	<STX>sAN{SPC}LIDrstoutpcnt{SPC}0<ETX>
	Hex	02 73 41 4E 20 4C 49 44 72 73 74 6F 75 74 70 63 6E 74 20 30 03
CoLa B	Binary	02 02 02 02 00 00 00 13 73 41 4E 20 4C 49 44 72 73 74 6F 75 74 70 63 6E 74 20 00 2F

Table 361: Example: sAN LIDrstoutpcnt

4.8.14 Read heating state

Note

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



Telegram structure: sRN OPheatstateext (no required user level)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sRN	73 52 4E
Command		String	14	All	OPheatstateext	4F 50 68 65 61 74 73 74 61 74 65 65 78 74

Table 362: Telegram structure: sRN OPheatstateext

Example: sRN OPheatstateext

CoLa A	ASCII	<STX>sRN{SPC}OPheatstateext<ETX>
	Hex	02 73 52 4E 20 4F 50 68 65 61 74 73 74 61 74 65 65 78 74 03
CoLa B	Binary	02 02 02 02 00 00 00 12 73 52 4E 20 4F 50 68 65 61 74 73 74 61 74 65 65 78 74 56

Table 363: Example: sRN OPheatstateext

Telegram structure: sRA OPheatstateext (no required user level)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sRA	73 52 41
Command		String	14	All	OPheatstateext	4F 50 68 65 61 74 73 74 61 74 65 65 78 74

Telegram structure: sRA OPheatstateext (no required user level)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Heating state	Heating state	Enum_8	1	All	0d = Heating active 1d = Current for heating too low 2d = Heating inactive 3d = Heating not connected	00 01 02 03

Table 364: Telegram structure: sRA OPheatstateext

Example: sRA OPheatstateext 2

CoLa A	ASCII	<STX>sRA{SPC}OPheatstateext{SPC}2<ETX>
	Hex	02 73 52 41 20 4F 50 68 65 61 74 73 74 61 74 65 65 78 74 20 32 03
CoLa B	Binary	02 02 02 02 00 00 00 14 73 52 41 20 4F 50 68 65 61 74 73 74 61 74 65 65 78 74 20 02 7B

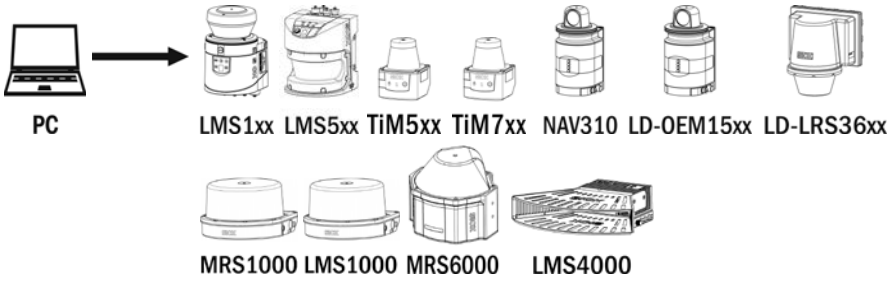
Table 365: Example: sRA OPheatstateext 2

4.9 Interfaces

4.9.1 Set IP address

IMPORTANT

- 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.



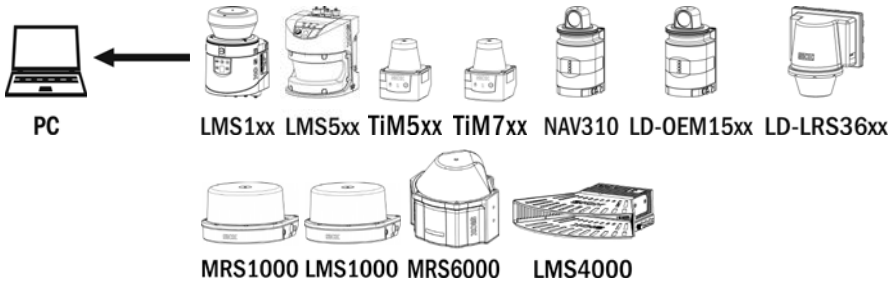
Telegram structure: sWN EIIpAddr (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Set IP address	String	8	All	EIIpAddr	45 49 49 50 41 64 64 72
IP address	Set values in hex	Uint_32	4	All	00 00 00 00h ... FF FF FF FFh (decimal values unwieldy)	00 00 00 00 ... FF FF FF FF

Table 366: Telegram structure: sWN EIIpAddr

Example: sWN EIIpAddr 192.168.0.2

CoLa A	ASCII	<STX>sWN{SPC}EIIpAddr{SPC}C0{SPC}A8{SPC}0{SPC}2<ETX>
	Hex	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	Binary	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 05

Table 367: Example: sWN EIIpAddr 192.168.0.2



Telegram structure: sWA EIIpAddr						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command	Set IP address	String	8	All	EIIpAddr	45 49 49 50 41 64 64 72

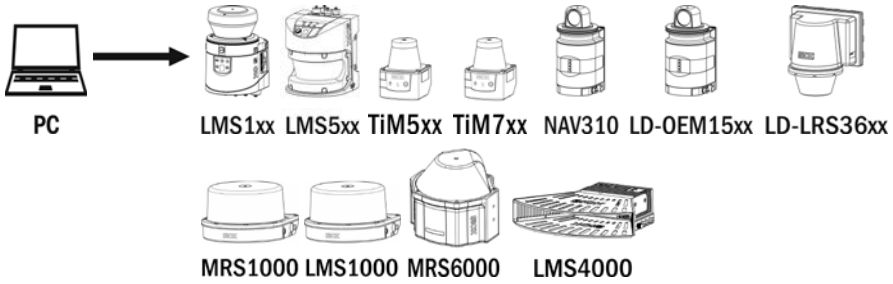
Table 368: Telegram structure: sWA EIIpAddr

Example: sWA EIIpAddr

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

Table 369: Example: sWA EIIpAddr

4.9.2 Read IP address



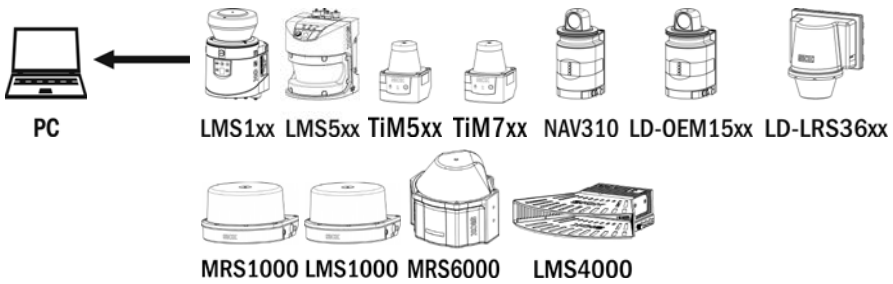
Telegram structure: sRN EIIpAddr						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sRN	73 52 4E
Command	Read IP address	String	8	All	EIIpAddr	45 49 49 50 41 64 64 72

Table 370: Telegram structure: sRN EIIpAddr

Example: srN EIIpAddr

CoLa A	ASCII	<STX>sRN{SPC}EIIpAddr<ETX>
	Hex	02 73 57 4E 20 45 49 49 70 41 64 64 72 03
CoLa B	Binary	02 02 02 02 00 00 00 0C 73 52 4E 20 45 49 49 70 41 64 64 72 49

Table 371: Example: srN EIIpAddr



Telegram structure: sRA EIIpAddr						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sRA	73 52 41
Command	Set IP address	String	8	All	EIIpAddr	45 49 49 50 41 64 64 72
IP address	Set values in hex	Uint_32	4	All	00 00 00 00h ... FF FF FF FFh (decimal values unwieldy)	00 00 00 00 ... FF FF FF FF

Table 372: Telegram structure: sRA EIIpAddr

Example: sRA EIIpAddr 192.168.0.2

CoLa A	ASCII	<STX>sRA{SPC}EIIpAddr{SPC}C0{SPC}A8{SPC}00{SPC}02<ETX>
	Hex	02 73 57 41 20 45 49 49 70 41 64 64 72 03
CoLa B	Binary	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

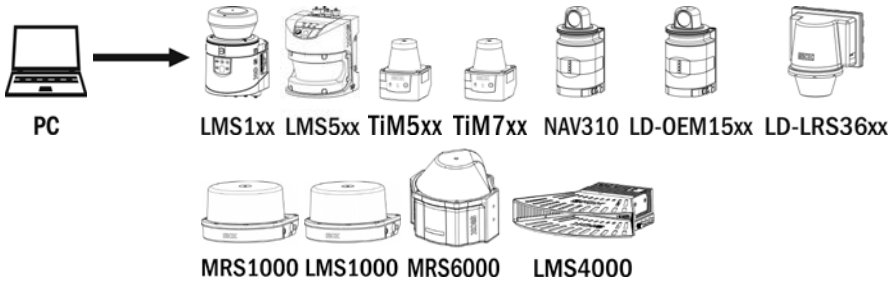
Table 373: Example: sRA EIIpAddr 192.168.0.2

4.9.3 Set Ethernet gateway

Change Ethernet gateway IP address (TCP/IP)

IMPORTANT

- 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.



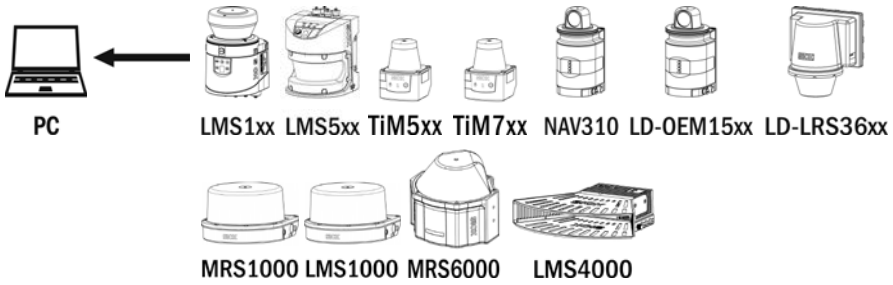
Telegram structure: sWN EIgate						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Set gateway adress	String	6	All	EIgate	45 49 67 61 74 65
Gateway address	Set values in hex	Uint_32	4	All	00 00 00 00h ... FF FF FF FFh (decimal values unwieldy)	00 00 00 00 ... FF FF FF FF

Table 374: Telegram structure: sWN EIgate

Example: sWN EIgate 192.168.0.1

CoLa A	ASCII	<STX>sWN{SPC}EIgate{SPC}C0{SPC}A8{SPC}00{SPC}01<ETX>
	Hex	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	Binary	02 02 02 02 00 00 00 0F 73 57 4E 20 45 49 67 61 74 65 20 C0 A8 00 01 5A

Table 375: Example: sWN EIgate 192.168.0.1



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

Table 376: Telegram structure: sWA Elgate

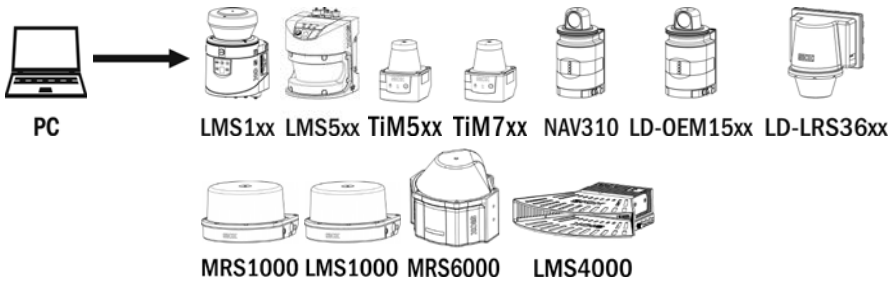
Example: sWA Elgate

CoLa A	ASCII	<STX>sWA{SPC}Elgate<ETX>
	Hex	02 73 57 41 20 45 49 67 61 74 65 03
CoLa B	Binary	02 02 02 02 00 00 00 0A 73 57 41 20 45 49 67 61 74 65 5E

Table 377: Example: sWA Elgate

4.9.4 Read Ethernet gateway

Read for the Ethernet gateway (TCP/IP)



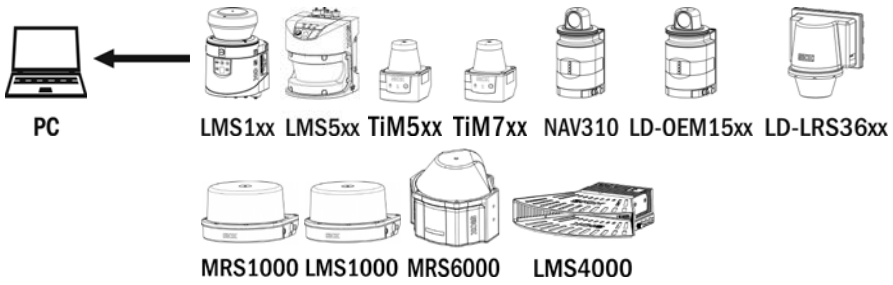
Telegram structure: sRN Elgate						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sRN	73 52 4E
Command	Read gateway address	String	6	All	Elgate	45 49 67 61 74 65

Table 378: Telegram structure: sRN Elgate

Example: sRN Elgate

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

Table 379: Example: sRN Elgate



Telegram structure: sRA Elgate						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sRA	73 52 41
Command	Read gateway address	String	6	All	Elgate	45 49 67 61 74 65
Gateway address	Values in hex	Uint_32	4	All	00 00 00 00h ... FF FF FF FFh (decimal values unwieldy)	00 00 00 00 ... FF FF FF FF

Table 380: Telegram structure: sRA Elgate

Example: sRA Elgate 192.168.0.1

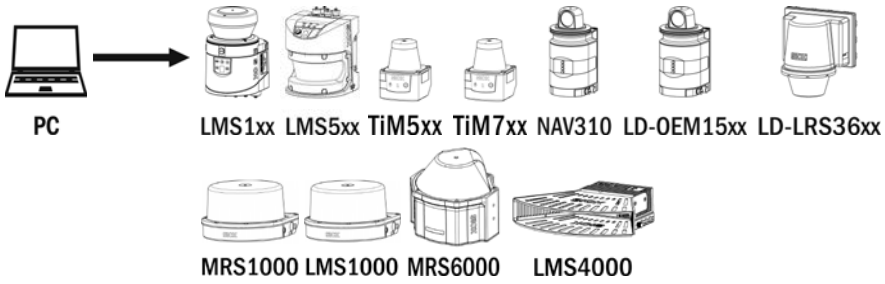
CoLa A	ASCII	<STX>sRA{SPC}Elgate{SPC}C0{SPC}A8{SPC}00{SPC}01<ETX>
	Hex	02 73 52 41 20 45 49 67 61 74 65 20 C0 A8 00 01 03
CoLa B	Binary	02 02 02 02 00 00 00 0F 73 52 41 20 45 49 67 61 74 65 20 C0 A8 00 01 12

Table 381: Example: sRA Elgate 192.168.0.1

4.9.5 Set IP mask

IMPORTANT

- 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.



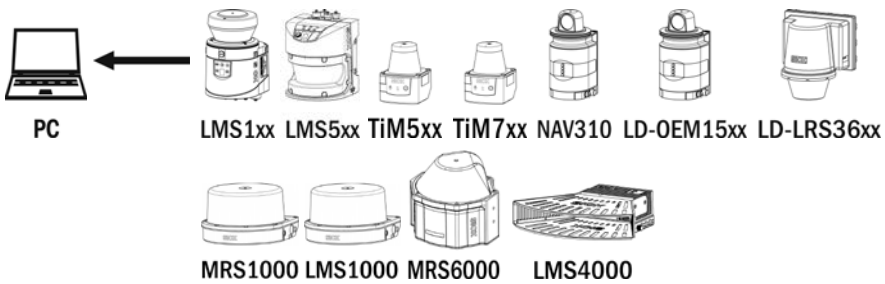
Telegram structure: sWN EImask						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Set IP mask	String	6	All	EImask	45 49 6D 61 73 6B
IP mask	Set values in hex	Uint_32	4	All	00 00 00 00h ... FF FF FF FFh (decimal values unwieldy)	00 00 00 00 ... FF FF FF FF

Table 382: Telegram structure: sWN EImask

Example: sWN EImask 255.255.254.0

CoLa A	ASCII	<STX>sWN{SPC}EImask{SPC}FF{SPC}FF{SPC}FE{SPC}00<ETX>
	Hex	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	Binary	02 02 02 02 00 00 00 0F 73 57 4E 20 45 49 6D 61 73 6B 20 FF FF FE 00 8C

Table 383: Example: sWN EImask 255.255.254.0



Telegram structure: sWA EImask						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command	Set IP mask	String	6	All	EImask	45 49 6D 61 73 6B

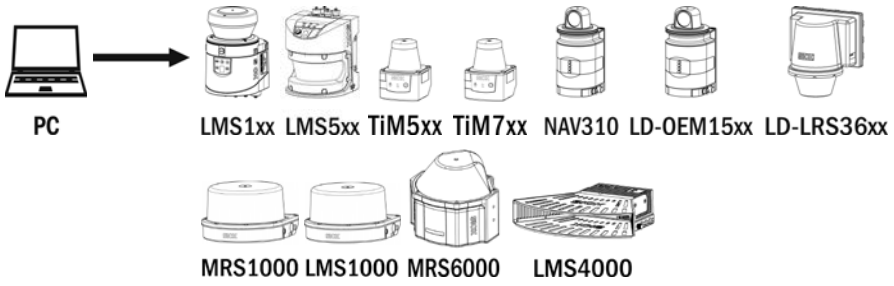
Table 384: Telegram structure: sWA EImask

Example: sWA EImask

CoLa A	ASCII	<STX>sWA{SPC}EImask<ETX>
	Hex	02 73 57 41 20 45 49 6D 61 73 6B 03
CoLa B	Binary	02 02 02 02 00 00 00 0A 73 57 41 20 45 49 6D 61 73 6B 63

Table 385: Example: sWA EImask

4.9.6 Read IP mask



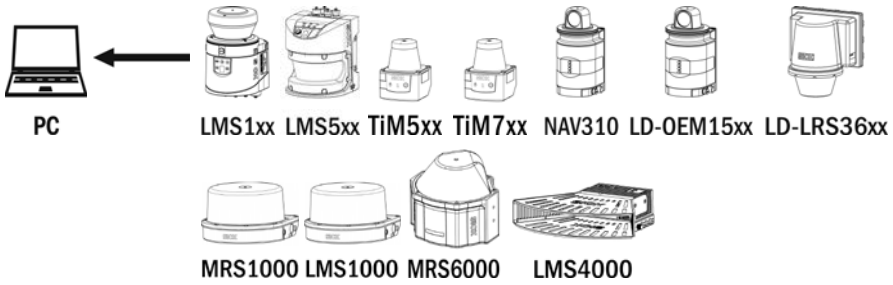
Telegram structure: sRN EImask						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sRN	73 52 4E
Command	Read IP mask	String	6	All	EImask	45 49 6D 61 73 6B

Table 386: Telegram structure: sRN EImask

Example: sRN EImask

CoLa A	ASCII	<STX>sRN{SPC}EImask<ETX>
	Hex	02 73 52 4E 20 45 49 6D 61 73 6B 03
CoLa B	Binary	02 02 02 02 00 00 00 0A 73 52 4E 20 45 49 6D 61 73 6B 57

Table 387: Example: sRN EImask



Telegram structure: sRA Elmask						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sRA	73 52 41
Command	Read IP mask	String	6	All	Elmask	45 49 6D 61 73 6B
IP mask	Values in hex	Uint_32	4	All	00 00 00 00h ... FF FF FF FFh (decimal values unwieldy)	00 00 00 00 ... FF FF FF FF

Table 388: Telegram structure: sRA Elmask

Example: sRA Elmask 255.255.254.0

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

Table 389: Example: sRA Elmask 255.255.254.0

4.9.7 Set baud rate for host interface

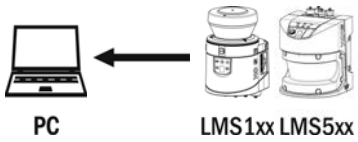
Telegram structure: sWN SIHstBaud (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Set baud rate for host interface	String	9	All	SIHstBaud	53 49 48 73 74 42 61 75 64
Baud rate data	Baud rate data for host interface	Enum_8	1	All	9600: +5d (05h) 19200: +6d (06h) 38400: +7d (07h) 57600: +8d (08h) 115200: +9d (09h)	9600: 05 19200: 06 38400: 07 57600: 08 115200: 09
				LMS1xx, LMS5xx	250000: +10d (0Ah) 500000: +11d (0Bh)	250000: 0A 500000: 0B

Table 390: Telegram structure: sWN SIHstBaud

Example: sWN SIHstBaud

CoLa A	ASCII	<STX>sWN{SPC}SIHstBaud{SPC}+8<ETX>
	Hex	02 73 57 4E 20 53 49 48 73 74 42 61 75 64 20 08 03
CoLa B	Binary	02 02 02 02 00 00 00 0F 73 57 4E 20 53 49 48 73 74 42 61 75 64 20 08 05

Table 391: Example: sWN SIHstBaud



Telegram structure: sWA SIHstBaud						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command	Set baud rate for host interface	String	9	All	SIHstBaud	53 49 48 73 74 42 61 75 64

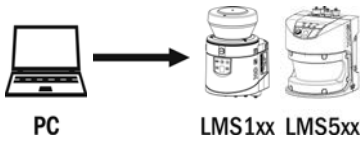
Table 392: Telegram structure: sWA SIHstBaud

Example: sWA SIHstBaud

CoLa A	ASCII	<STX>sWA{SPC}SIHstBaud<ETX>
	Hex	02 73 57 41 20 53 49 48 73 74 42 61 75 64 03
CoLa B	Binary	02 02 02 02 00 00 00 0E 73 57 41 20 53 49 48 73 74 42 61 75 64 20 02

Table 393: Example: sWA SIHstBaud

4.9.8 Read baud rate of host interface



Telegram structure: sRN SIHstBaud						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sRN	73 52 4E
Command	Read baud rate of host interface	String	9	All	SIHstBaud	53 49 48 73 74 42 61 75 64

Table 394: Telegram structure: sRN SIHstBaud

Example: sRN SIHstBaud

CoLa A	ASCII	<STX>sRN{SPC}SIHstBaud<ETX>
	Hex	02 73 52 4E 20 53 49 48 73 74 42 61 75 64 03
CoLa B	Binary	02 02 02 02 00 00 00 0D 73 52 4E 20 53 49 48 73 74 42 61 75 64 28

Table 395: Example: sRN SIHstBaud

Telegram structure: sRA SIHstBaud						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sRA	73 52 41
Command	Read baud rate of host interface	String	9	All	SIHstBaud	53 49 48 73 74 42 61 75 64
Baud rate data	Baud rate data of host interface	Enum_8	1	All	9600: 5d (05h) 19200: 6d (06h) 38400: 7d (07h) 57600: 8d (08h) 115200: 9d (09h)	9600: 05 19200: 06 38400: 07 57600: 08 115200: 09
				LMS1xx, LMS5xx	250000: 10d (0Ah) 500000: 11d (0Bh)	250000: 0A 500000: 0B

Table 396: Telegram structure: sRA SIHstBaud

Example: sRA SIHstBaud

CoLa A	ASCII	<STX>sRA{SPC}SIHstBaud{SPC}8<ETX>
	Hex	02 73 52 41 20 53 49 48 73 74 42 61 75 64 20 08 03
CoLa B	Binary	02 02 02 02 00 00 00 0F 73 52 41 20 53 49 48 73 74 42 61 75 64 20 08 0F

Table 397: Example: sRA SIHstBaud

4.9.9 Set interface type

Telegram structure: sWN SIHstHw (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Set hardware settings for host interface	String	7	All	SIHstHw	53 49 48 73 74 48 77
Interface type data	Hardware settings data for host interface	Enum_8	1	All	TX_RS232: 0 TX_RS485_2WIRE: 1 TX_RS422_485_4WIRE: 2	TX_RS232: 00 TX_RS485_2WIRE: 01 TX_RS422_485_4WIRE: 02

Table 398: Telegram structure: sWN SIHstHw

Example: sWN SIHstHw

CoLa A	ASCII	<STX>sWN{SPC}SIHstHw{SPC}0<ETX>
	Hex	02 73 57 4E 20 53 49 48 73 74 48 77 20 00 03
CoLa B	Binary	02 02 02 02 00 00 00 0D 73 57 4E 20 53 49 48 73 74 48 77 20 00 00

Table 399: Example: sWN SIHstHw

Telegram structure: sWA SIHstHw						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command	Set hardware settings for host interface	String	7	All	SIHstHw	53 49 48 73 74 48 77

Table 400: Telegram structure: sWA SIHstHw

Example: sWA SIHstHw

CoLa A	ASCII	<STX>sWA{SPC}SIHstHw<ETX>
	Hex	02 73 57 41 20 53 49 48 73 74 48 77 03
CoLa B	Binary	02 02 02 02 00 00 00 0C 73 57 41 20 53 49 48 73 74 48 77 20 0F

Table 401: Example: sWA SIHstHw

4.9.10 Read interface type

Telegram structure: sRN SIHstHw						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sRN	73 52 4E
Command	Read hardware settings of host interface	String	7	All	SIHstHw	53 49 48 73 74 48 77

Table 402: Telegram structure: sRN SIHstHw

Example: sRN SIHstHw

CoLa A	ASCII	<STX>sRN{SPC}SIHstHw<ETX>
	Hex	02 73 52 4E 20 53 49 48 73 74 48 77 03
CoLa B	Binary	02 02 02 02 00 00 00 0B 73 52 4E 20 53 49 48 73 74 48 77 25

Table 403: Example: sRN SIHstHw

Telegram structure: sRA SIHstHw						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sRA	73 52 41
Command	Read hardware settings of host interface	String	7	All	SIHstHw	53 49 48 73 74 48 77
Interface type data	Hardware settings data of host interface	Enum_8	1	All	TX_RS232: 0 TX_RS485_2WIRE: 1 TX_RS422_485_4WIRE: 2	TX_RS232: 00 TX_RS485_2WIRE: 01 TX_RS422_485_4WIRE: 02

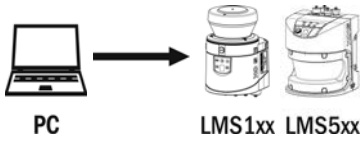
Table 404: Telegram structure: sRA SIHstHw

Example: sRA SIHstHw

CoLa A	ASCII	<STX>sRA{SPC}SIHstHw{SPC}0<ETX>
	Hex	02 73 57 41 20 53 49 48 73 74 48 77 20 00 03
CoLa B	Binary	02 02 02 02 00 00 00 0D 73 52 41 20 53 49 48 73 74 48 77 20 00 0A

Table 405: Example: sRA SIHstHw

4.9.11 Set function front panel



Telegram structure: sWN LMLfpFcn (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Set function of the front panel	String	8	All	LMLfpFcn	4C 4D 4C 66 70 46 63 6E
Reserved	Reserved	Bool_1	1	All	1	01
LED function Q1/Q2	Code number	Enum_8	1	All	No function: 0 Application: 1 Command: 2	No function: 00 Application: 01 Command: 02
LED function OK/Stop	Code number	Enum_8	1	All	Application: 0 Command: 1	Application: 00 Command: 01
Display function	Code number	Enum_8	1	All	No function: 0 Application: 1 Command: 2	No function: 00 Application: 01 Command: 02

Table 406: Telegram structure: sWN LMLfpFcn

Example: sWN LMLfpFcn

CoLa A	ASCII	<STX>sWN{SPC}LMLfpFcn{SPC}1{SPC}1{SPC}0{SPC}1<ETX>
	Hex	02 73 57 4E 20 4C 4D 4C 66 70 46 63 6E 20 31 20 31 20 30 20 31 03
CoLa B	Binary	02 02 02 02 00 00 00 11 73 57 4E 20 4C 4D 4C 66 70 46 63 6E 20 01 01 00 01 7B

Table 407: Example: sWN LMLfpFcn

Telegram structure: sWA LMLfpFcn						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command	Front panel function	String	8	All	LMLfpFcn	4C 4D 4C 66 70 46 63 6E

Table 408: Telegram structure: sWA LMLfpFcn

Example: sWA LMLfpFcn

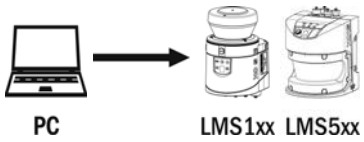
CoLa A	ASCII	<STX>sWA{SPC}LMLfpFcn<ETX>
	Hex	02 73 57 41 20 4C 4D 4C 66 70 46 63 6E 03
CoLa B	Binary	02 02 02 02 00 00 00 0D 73 57 41 20 4C 4D 4C 66 70 46 63 6E 75

Table 409: Example: sWA LMLfpFcn

4.9.12 Set front LEDs

To use this command, it is necessary to set the function of the LED to “Command” (use sWN LMLfpFcn), otherwise this command will have no influence to the LEDs.

OK and Stop LED can only alternate, if one is switched on, the other will turn automatically off.



Telegram structure: sMN mLMLSetLed						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3	All	sMN	73 4D 4E
Command	Set front LED	String	10	All	mLMLSetLed	6D 4C 4D 4C 53 65 74 4C 65 64
LED	LED to turn on/off	Int_8	1	All	Stop: 1 OK: 2 Q1: 3 Q2: 4	Stop: 01 OK: 02 Q1: 03 Q2: 04
Status	On or Off	Int_8	1	All	On: 1 Off: 0	On: 01 Off: 00

Table 410: Telegram structure: sMN mLMLSetLed

Example: sMN mLMLSetLed 1 1 (Stop LED)

CoLa A	ASCII	<STX>sMN{SPC}mLMLSetLed{SPC}1{SPC}1<ETX>
	Hex	02 73 4D 4E 20 6D 4C 4D 4C 53 65 74 4C 65 64 20 31 20 31 03
CoLa B	Binary	02 02 02 02 00 00 00 12 73 4D 4E 20 6D 4C 4D 4C 53 65 74 4C 65 64 20 01 20 01 7F

Table 411: Example: sMN mLMLSetLed 1 1 (Stop LED)

Telegram structure: sAN mLMLSetLed						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sAN	73 41 4E
Command	Front LED	String	10	All	mLMLSetLed	6D 4C 4D 4C 53 65 74 4C 65 64
Status code	Code number	Bool_1	1	All	Error: 0 Success: 1	Error: 00 Success: 01

Table 412: Telegram structure: sAN mLMLSetLed

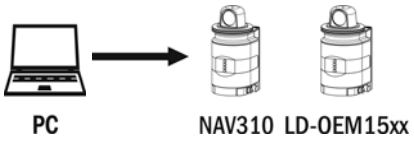
Example: sAN mLMLSetLed

CoLa A	ASCII	<STX>sAN{SPC}mLMLSetLed{SPC}0<ETX>
	Hex	02 73 41 4E 20 6D 4C 4D 4C 53 65 74 4C 65 64 20 00 03
CoLa B	Binary	02 02 02 02 00 00 00 10 73 41 4E 20 6D 4C 4D 4C 53 65 74 4C 65 64 20 00 53

Table 413: Example: sAN mLMLSetLed

4.9.13 Set function of LED1

With this command the operation of LED1 can be defined. Either it has no function (00), it flashes when output Q1 or application is active (01) or it can be turned on and off (02) by another telegram command (sMN mHMISetLed).



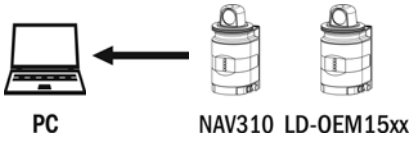
Telegram structure: sWN HMIfpFcn_Y1 (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Set function of the front panel LED1	String	11	All	HMIfpFcn_Y1	48 4D 49 66 70 46 63 6E 5F 59 31
LED1 function Q1	Code number	Enum_8	1	All	No function: 0 Application: 1 Command: 2	No function: 00 Application: 01 Command: 02

Table 414: Telegram structure: sWN HMIfpFcn_Y1

Example: sWN HMIfpFcn_Y1 = Command

CoLa A	ASCII	<STX>sWN{SPC}HMIfpFcn_Y1{SPC}2<ETX>
	Hex	02 73 57 4E 20 48 4D 49 66 70 46 63 6E 5F 59 31 20 32 03
CoLa B	Binary	02 02 02 02 00 00 00 11 73 57 4E 20 48 4D 49 66 70 46 63 6E 5F 59 31 20 02 4E

Table 415: Example: sWN HMIfpFcn_Y1 = Command



Telegram structure: sWA HMIfpFcn_Y1						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command	LED1 function	String	11	All	HMIfpFcn_Y1	48 4D 49 66 70 46 63 6E 5F 59 31

Table 416: Telegram structure: sWA HMIfpFcn_Y1

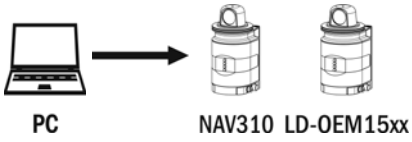
Example: sWA HMIfpFcn_Y1

CoLa A	ASCII	<STX>sWA{SPC}HMIfpFcn_Y1<ETX>
	Hex	02 73 57 41 20 48 4D 49 66 70 46 63 6E 5F 59 31 03
CoLa B	Binary	02 02 02 02 00 00 00 0F 73 57 41 20 48 4D 49 66 70 46 63 6E 5F 59 31 63

Table 417: Example: sWA HMIfpFcn_Y1

4.9.14 Set function of LED2

With this command the operation of LED2 can be defined. Either it has no function (00), it flashes when output Q2 or application is active (01) or it can be turned on and off (02) by another telegram command (sMN mHMISetLed).



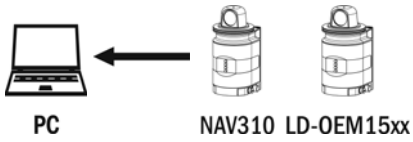
Telegram structure: sWN HMIfpFcn_Y2 (Authorized client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Write	String	3	All	sWN	73 57 4E
Command	Set function of the front panel LED2	String	11	All	HMIfpFcn_Y2	48 4D 49 66 70 46 63 6E 5F 59 32
LED2 function Q2	Code number	Enum_8	1	All	No function: 0 Application: 1 Command: 2	No function: 00 Application: 01 Command: 02

Table 418: Telegram structure: sWN HMIfpFcn_Y2

Example: sWN HMIfpFcn_Y2 = Command

CoLa A	ASCII	<STX>sWN{SPC}HMIfpFcn_Y2{SPC}2<ETX>
	Hex	02 73 57 4E 20 48 4D 49 66 70 46 63 6E 5F 59 32 20 32 03
CoLa B	Binary	02 02 02 02 00 00 00 11 73 57 4E 20 48 4D 49 66 70 46 63 6E 5F 59 32 20 02 7D

Table 419: Example: sWN HMIfpFcn_Y2 = Command



Telegram structure: sWA HMIfpFcn_Y2						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command	LED2 function	String	11	All	HMIfpFcn_Y2	48 4D 49 66 70 46 63 6E 5F 59 32

Table 420: Telegram structure: sWA HMIfpFcn_Y2

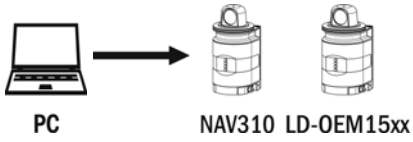
Example: sWA HMIfpFcn_Y2

CoLa A	ASCII	<STX>sWA{SPC}HMIfpFcn_Y2<ETX>
	Hex	02 73 57 41 20 48 4D 49 66 70 46 63 6E 5F 59 32 03
CoLa B	Binary	02 02 02 02 00 00 00 0F 73 57 41 20 48 4D 49 66 70 46 63 6E 5F 59 32 60

Table 421: Example: sWA HMIfpFcn_Y2

4.9.15 Switch on/off LED1 or LED2

With this command the LEDs can be switched on and off (e.g. to locate the sensor or test the conncection). As a prerequisite, the operation of LED1 and LED2 must have been set to the right function (sWN HMIfpFcn_).



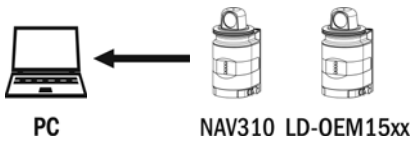
Telegram structure: sMN mHMISetLed						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Method	String	3	All	sMN	73 4D 4E
Command	Set function of the front panel	String	10	All	mHMISetLed	6D 48 4D 49 53 65 74 4C 65 64
LED number 1/2	LED number	Uint_8	1	All	LED 1: 3 LED 2: 4	LED 1: 03 LED 2: 04
LED function off/on	Code number	Uint_8	1	All	Off: 0 On: 1	Off: 00 On: 01

Table 422: Telegram structure: sMN mHMISetLed

Example: sMN mHMISetLed 1 = On

CoLa A	ASCII	<STX>sMN{SPC}mHMISetLed{SPC}3{SPC}1<ETX>
	Hex	02 73 4D 4E 20 6D 48 4D 49 53 65 74 4C 65 64 20 33 20 31 03
CoLa B	Binary	02 02 02 02 00 00 00 12 73 4D 4E 20 6D 48 4D 49 53 65 74 4C 65 64 20 03 20 01 7C

Table 423: Example: sMN mHMISetLed 1 = On



Telegram structure: sAN mHMISetLed						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sAN	73 41 4E
Command	LED status	String	10	All	mHMISetLed	6D 48 4D 49 53 65 74 4C 65 64
Result	Code number	Bool_1	1	All	No success: 0 Success: 1	No success: 00 Success: 01

Table 424: Telegram structure: sAN mHMISetLed

Example: sAN mHMISetLed 01

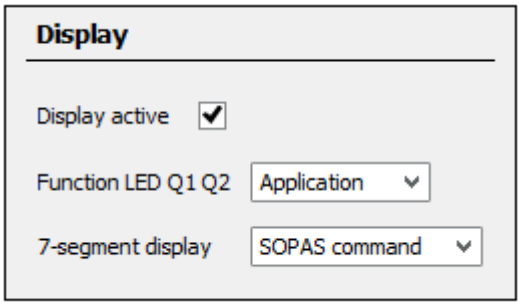
CoLa A	ASCII	<STX>sAN{SPC}mHMISetLed{SPC}1<ETX>
	Hex	02 73 41 4E 20 6D 48 4D 49 53 65 74 4C 65 64 20 31 03
CoLa B	Binary	02 02 02 02 00 00 00 10 73 41 4E 20 6D 48 4D 49 53 65 74 4C 65 64 20 01 53

Table 425: Example: sAN mHMISetLed 01

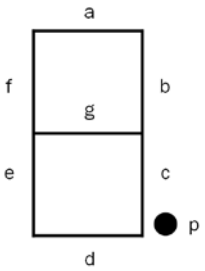
4.9.16 Set 7-segment display to specific symbol or number

Precondition

It is mandatory to define that the 7-segment display should react to SOPAS commands. This option needs to be activated using the engineering tool “SOPAS”. Choose “SOPAS command” in the drop-down list for the 7-segment display (as shown in the figure below). Afterwards the segments of the display can be set via telegram.



The display is consisting of the segments A-P:



The segments A-G are operated using the bits 0 to 6, the segment P using bit 7:

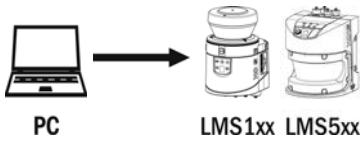
Segment	p	g	f	e	d	c	b	a
Related bit	7	6	5	4	3	2	1	0

Example: Showing the number “7” on the display:

Segment	p	g	f	e	d	c	b	a
On/Off for showing the required symbol	off	off	off	off	off	on	on	on
Binary	0	0	0	0	0	1	1	1

Transfer binary into ASCII:

00000111 bin ≙ 07 ASCII (30 37 hex)



Telegram structure: sMN mLMLSetDisp (no required user level)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Request (SOPAS method by name)	String	3	All	sMN	73 4D 4E
Command	Set 7-segment display	String	11	All	mLMLSetDisp	6D 4C 4D 4C 53 65 74 44 69 73 70
Display	7-segment display in the display of the LMS. The segments A-G are operated using the bits 0 to 6, the point P using bit 7.	Uint_8	1	All	00h display off ... 07h display 7 ... FFh display completely on	LMS1xx: 00 ... 07 ... FF LMS5xx: 30 30 ... 30 37 ... 46 46

Table 426: Telegram structure: sMN mLMLSetDisp

Example: sMN mLMLSetDisp 07 (Showing the number “7” on the display)

CoLa A	ASCII	<STX>sMN{SPC}mLMLSetDisp{SPC}07<ETX>
	Hex	02 73 4D 4E 20 6D 4C 4D 4C 53 65 74 44 69 73 70 20 30 37 03
CoLa B	Binary	LMS1xx: 02 02 02 02 00 00 00 11 73 4D 4E 20 6D 4C 4D 4C 53 65 74 44 69 73 70 20 07 3B LMS5xx: 02 02 02 02 00 00 00 12 73 4D 4E 20 6D 4C 4D 4C 53 65 74 44 69 73 70 20 30 37 3B

Table 427: Example: sMN mLMLSetDisp 07 (Showing the number “7” on the display)

Telegram structure: sAN mLMLSetDisp (no required user level)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer (SOPAS answer)	String	3	All	sAN	73 41 4E
Command	Set 7-segment display	String	11	All	mLMLSetDisp	6D 4C 4D 4C 53 65 74 44 69 73 70
ErrorCode	The command has been accepted if the error code 1 is returned.	Enum8	1	All	0 error 1 no error	LMS1xx: 00 error 01 no error LMS5xx: 30 error 31 no error

Table 428: Telegram structure: sAN mLMLSetDisp

Example: sAN mLMLSetDisp 1

CoLa A	ASCII	<STX>sAN{SPC}mLMLSetDisp{SPC}1<ETX>
	Hex	02 73 41 4E 20 6D 4C 4D 4C 53 65 74 44 69 73 70 20 31 03
CoLa B	Binary	02 02 02 02 00 00 00 11 73 41 4E 20 6D 4C 4D 4C 53 65 74 44 69 73 70 20 01 31

Table 429: Example: sAN mLMLSetDisp 1

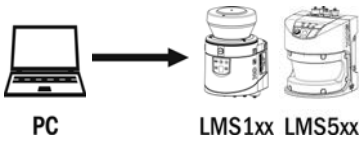
4.9.17 Switching CoLa dialect on ethernet host port

Note

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

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

Binary CRC32 is available since the firmware version V1.80.0



Telegram structure: sWN EIHstCola (user level 'authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sWN	73 57 4E
Command		String	9	All	EIHstCola	45 49 48 73 74 43 6F 6C 61
CoLa dialect	CoLa dialect	Enum_8	1	All	0d = CoLa ASCII 1d = CoLa Binary 2d = Binary CRC32	00 01 02

Table 430: Telegram structure: sMN EIHstCola

Example: sWN EIHstCola 0

CoLa A	ASCII	<STX>sWN{SPC}EIHstCola{SPC}0<ETX>
	Hex	02 73 57 4E 20 45 49 48 73 74 43 6F 6C 61 20 30 03
CoLa B	Binary	02 02 02 02 00 00 00 0F 73 57 4E 20 45 49 48 73 74 43 6F 6C 61 20 00 08

Table 431: Example: sWN EIHstCola 0

Telegram structure: sWA EIHstCola (user level 'authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Answer	String	3	All	sWA	73 57 41
Command		String	9	All	EIHstCola	45 49 48 73 74 43 6F 6C

Telegram structure: sWA EIHstCola (user level 'authorized client' required)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
						61

Table 432: Telegram structure: sWA EIHstCola

Example: sWA EIHstCola

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

Table 433: Example: sWA EIHstCola

4.10 Application

4.10.1 Request status change of monitoring fields on event

Precondition

Necessary sensor setup:

1. Setup detection fields
2. Setup evaluation cases and assign outputs to the evaluation cases

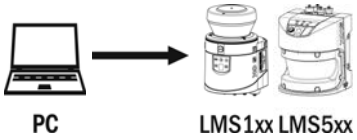
Advantage of this telegram:

Status of the evaluation case (1=field free, 2=detecting or 3=infringement) is transmitted if there is a status change of the monitoring field within the application.

For example if the object size in the field exceeds the set object size parameter but the duration of the object inside the field is below the set time parameter. Then the status change from “field free” to “detecting” will be given out via telegram.

Necessary procedure after each power up of the sensor:

1. Establish Ethernet connection to LMS
2. Activate Ethernet output



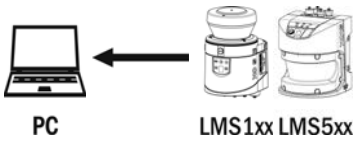
Telegram structure: sEN ECRChangeArr (no required user level)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sEN	73 45 4E
Command	Data telegram	String	12	All	ECRChangeArr	45 43 52 43 68 61 6E 67 65 41 72 72
Reporting	Start/stop	Enum_8	1	LM1xx LMS5XX	Stop: 0 Start: 1	LMS1xx: Stop: 00 Start: 01 LMS5xx: Stop: 30 Start: 31

Table 434: Telegram structure: sEN ECRChangeArr

Example: sEN ECRChangeArr 1

CoLa A	ASCII	<STX>sEN{SPC}ECR ChangeArr{SPC}1<ETX>
	Hex	02 73 45 4E 20 45 43 52 43 68 61 6E 67 65 41 72 72 20 31 03
CoLa B	Binary	02 02 02 02 00 00 00 12 73 45 4E 20 45 43 52 43 68 61 6E 67 65 41 72 72 20 01 7A

Table 435: Example: sEN ECRChangeArr 1



Telegram structure: sEA ECRChangeArr (no required user level)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sEA	73 45 41
Command	Data telegram	String	12	All	ECRChangeArr	45 43 52 43 68 61 6E 67 65 41 72 72
Reporting	Start/stop	Enum_8	1	LM1xx LMS5XX	Stop: 0 Start: 1	LMS1xx: Stop: 00 Start: 01 LMS5xx: Stop: 30 Start: 31

Table 436: Telegram structure: sEA ECRChangeArr

Example: sEA ECRChangeArr 1

CoLa A	ASCII	<STX>sEA{SPC}ECRChangeArr{SPC}1<ETX>
	Hex	02 73 45 41 20 45 43 52 43 68 61 6E 67 65 41 72 72 20 31 03
CoLa B	Binary	02 02 02 02 00 00 00 12 73 45 41 20 45 43 52 43 68 61 6E 67 65 41 72 72 20 01 75

Table 437: Example: sEA ECRChangeArr 1

NOTE

The answer to the telegram will be followed by data that is sent on event.

The sensor only sends the following answer if there is a status change of the evaluation case within the application.

Telegram structure: sSN ECRChangeArr (no required user level)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command	Read	String	3	All	sSN	73 53 4E

Telegram structure: sSN ECRChangeArr (no required user level)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
type						
Command	Data telegram	String	12	All	ECRChangeArr	45 43 52 43 68 61 6E 67 65 41 72 72
System counter	Time in µs since power up max. 71min then starting from 0 again	Uint_32	4	LM1xx LMS5XX	0 ... FFFFFFFFh	00 00 00 00 ... FF FF FF FF
Array	1-10	UInt_8	1	LM1xx LMS5XX	0h ... Ah (0d ...10d)	01 ... 0A
EVC number	1-10	UInt_8	1	LM1xx LMS5XX	0h ... Ah (0d ...10d)	01 ... 0A
Object detection		Enum_8	1	LM1xx LMS5XX	1= field free 2= detecting 3= infrigement	LMS1xx: 01= field free 02= detecting 03= infrigement LMS5xx: 31= field free 32= detecting 33= infrigement
Year		Uint_16	2	LM1xx LMS5XX	0000h ... 270Fh (0d...9999d)	00 00 ... 27 0F
Month	1 to 12	Uint_8	1	LM1xx LMS5XX	00h ... 0Ch (0d ... 12d)	00 ... 0C
Day	Day of month 1 to 31	Uint_8	1	LM1xx LMS5XX	00h ... 1Fh (0d ... 31d)	00 ... 1F
Hour	0 to 23	Uint_8	1	LM1xx LMS5XX	00h ... 17h (0d ... 23d)	00 ... 17
Minute	0 to 59	Uint_8	1	LM1xx LMS5XX	00h ... 3Bh 0d ... 59d	00 ... 3B
Second	0 to 59	Uint_8	1	LM1xx LMS5XX	00h ... 3Bh (0d ... 59d)	00 ... 3B
µSecond	0 to 9999999	Uint_32	4	LM1xx LMS5XX	00000000h ... 000F423Fh (0d ... 999999d)	00 00 00 00 ... 00 0F 42 3F

Table 438: Telegram structure: sSN ECRChangeArr

Example: sSN ECRChangeArr 1

CoLa A	ASCII	<STX>sSN{SPC}ECRChangeArr{SPC}97F8C2E3{SPC}1{SPC}1{SPC}1{SPC}7B2{SPC}1{SPC}1{SPC}0{SPC}2A{SPC}1D{SPC}63DA8<ETX>
	Hex	02 73 53 4E 20 45 43 52 43 68 61 6E 67 65 41 72 72 20 39 37 46 38 43 32 45 33 20 31 20 31 20 31 20 37 42 32 20 31 20 31 20 30 20 32 41 20 31 44 20 36 33 44 41 38 03
CoLa B	Binary	02 02 02 02 00 00 00 24 73 53 4E 20 45 43 52 43 68 61 6E 67 65 41 72 72 20 CF 09 10 99 00 01 01 02 07 B2 01 01 02 09 1C 00 03 47 D8 2E

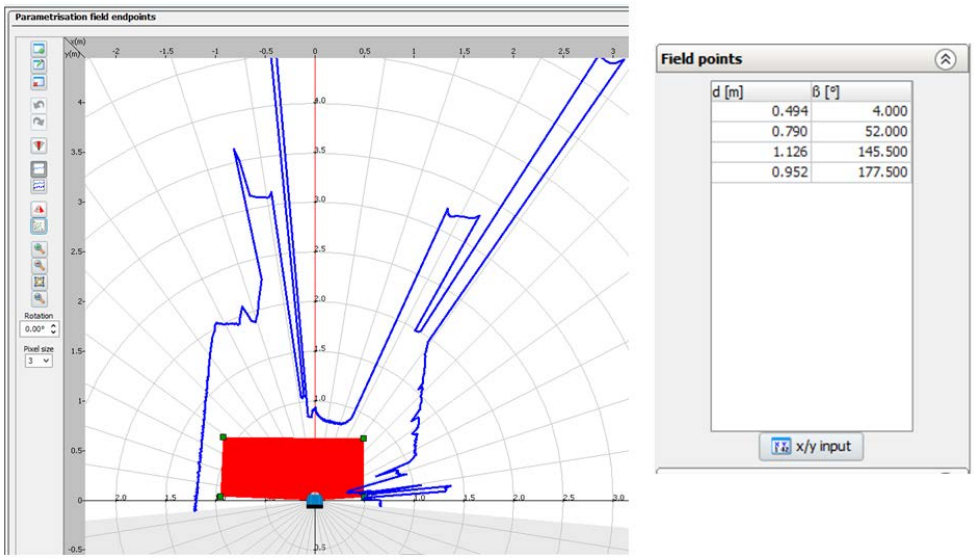
Table 439: Example: sSN ECRChangeArr 1

4.10.2 Request SOPAS field data structure

Note

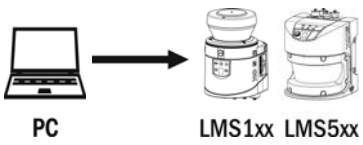
The SOPAS telegram “mLFEgetField” requests a field number as parameter and returns the corresponding SOPAS field data structure. If the field number is number is not configured, the answer telegram will be filled with 0.

Example – Request the field data structure of an evaluation field (field number: 1) that has been parameterized in the engineering tool SOPAS.



The answer telegram will include information regarding the field data structure of the requested evaluation field (type: segmented, number of field points etc.)

Please note: The sensor will switch to the state “Stop measurement” during read out. After the read out you have to switch the sensor back to “Run measurement”.



Telegram structure: sMN mLFEgetField (required user level: Authorized Client)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Request	String	3	LMS1xx LMS5XX	sMN	73 4D 4E
Command	Only one telegram	String	14	LMS1xx LMS5XX	mLFEgetField	6D 4C 46 45 67 65 74 46 69 65 6C 64
Command		UInt_32	4	LMS1xx LMS5XX	1...10	00 00 00 01...00 00 01 01

Table 440: Telegram structure: sMN mLFEgetField

Example: sMN mLFEgetField 1

CoLa A	ASCII	<STX>sMN{SPC}mLFEgetField{SPC}1<ETX>
	Hex	02 73 4D 4E 20 6D 4C 46 45 67 65 74 46 69 65 6C 64 20 31 03
CoLa B	Binary	02 02 02 02 00 00 00 15 73 4D 4E 20 6D 4C 46 45 67 65 74 46 69 65 6C 64 20 00 00 00 01 67

Table 441: Example: sMN mLFEgetField 1

Telegram structure: sAN mLFEgetField							
Telegram part		**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Read			String	3	All	sAN	73 41 4E
Data telegram			String	12	LMS1xx LMS5xx	mLFEgetField	6D 4C 46 45 67 65 74 46 69 65 6C 64
Field header	Distance Scale Factor	Scale factor or factor of the measurement values (for the LMS5xx this depends on the angular resolution)	Real as float according to IEEE754	4	LMS1xx LMS5xx	Factor × 1: 3F800000h Factor × 2: 40000000h	3F 80 00 00 40 00 00 00
	Distance Scale Offset	Sets starting point of measurement	Real as float according to IEEE754	4	LMS1xx LMS5xx	00000000	00 00 00 00
	Angle Scale Factor	Angle resolution	UInt_32	4	LMS1xx LMS5xx	+50000d (1388h)	00 00 13 88
	Angle Scale Offset	Starting angle	Int_32	4	LMS1xx LMS5xx	-50000d (FFFF3CB0h)	FF FF 3C B0
	Field type		Enum_8	1	LMS1xx LMS5xx	0 = Radial 1 = Rectangle 2 = Segmented 3 = Dynamic	00 01 02 03
	Field number		UInt_8	1	LMS1xx LMS5xx	0 .. 10	00...0A

Segmented field	Segmented field configured			UInt_16	2	LMS1xx LMS5xx	0 = no data for segmented field available, i.e. field type is not segmented field 1 = data for segmented field available, i.e. field type is segemented field	00 00...00 01
	Only shown if field type is configured	Number of field points		UInt_16	2	LMS1xx LMS5xx	0 .. 571	00 00... 02 3B
		Angle index		UInt_16	2	LMS1xx LMS5xx	LMS1xx: 0 .. 1081 LMS5xx: 0 .. 1140	00 00 ... 04 39 00 00 ... 04 74
		Start distance		UInt_16	2	LMS1xx LMS5xx	0 .. 65535 (FFFFh) FFFFh = not set	00 00...FF FF
		End distance		UInt_16	2	LMS1xx LMS5xx	0 .. 65535 (FFFFh) FFFFh = not set	00 00...FF FF
		...		...	...	...	...	...
		Angle index		UInt_16	2	LMS1xx LMS5xx	LMS1xx: 0 .. 1081 LMS5xx: 0 .. 1140	00 00 ... 04 39 00 00 ... 04 74
		Start distance		UInt_16	2	LMS1xx LMS5xx	0 .. 65535 (FFFFh) FFFFh = not set	00 00...FF FF
		End distance		UInt_16	2	LMS1xx LMS5xx	0 .. 65535 (FFFFh) FFFFh = not set	00 00...FF FF
Rectangular field	Rectangular field configured			UInt_16	2	LMS1xx LMS5xx	0 = no data for rectangular field available, i.e. field type is not rectangular field 1 = data for rectangular field available, i.e. field type is rectangular field	00 00...00 01
	Only shown if field type is configured	Angle of reference point		Int_32	4	LMS1xx LMS5xx	LMS1xx: FFF92230h ... 225510h (-450000d ... 225,0000d) LMS5xx: -50000 .. 1850000	FF F9 22 30 ... 00 22 55 10

		Distance of reference point		UInt_16	2	LMS1xx LMS5xx	LMS1xx: LMS100, LMS111: 0 h ...4E20h (0d..20000d) LMS151: 0h...C350h (0d...50000d) LMS5xx 0 .. 65535	00 00 00 00 ... 00 00 4E 20 00 00 00 00 ... 00 00 C3 50
		Rotation angle		Int_32	4	LMS1xx LMS5xx	-1800000 .. 1800000	FF E4 88 C0...00 1B 77 40
		Width		UInt_32	4	LMS1xx LMS5xx	0 .. 4294967295	00 00 00 00 ... 7F FF FF FF
		Length		UInt_32	4	LMS1xx LMS5xx	0 .. 4294967295	00 00 00 00 ... 7F FF FF FF
Radial field			Radial field configured	UInt_16	2	LMS1xx LMS5xx	0 (radial fields are not available)	00 00
Dynamic field	Dynamic field configured			UInt_16	2	LMS1xx LMS5xx	0 = no data for dynamic field available, i.e. field type is not dynamic field 1 = data for dynamic field available, i.e. field type is dynamic field	00 00...00 01
	Only shown if field type is configured	Angle of reference point		Int_32	4	LMS1xx LMS5xx	LMS1xx: -450000 ... 2250000 LMS5xx: -50000 .. 1850000	FF F9 22 30 ... 00 22 55 10
		Distance of reference point		UInt_16	2	LMS1xx LMS5xx	LMS100, LMS111: 0 h ...4E20h (0d..20000d) LMS151: 0h...C350h (0d...50000d) LMS5xx: 0 .. 65535	00 00 00 00 ... 00 00 4E 20 00 00 00 00 ... 00 00 C3 50
		Rotation angle		Int_32	4	LMS1xx LMS5xx	-1800000 .. 1800000	FF E4 88 C0...00 1B 77 40
		Width		UInt_32	4	LMS1xx LMS5xx	0 .. 4294967295	00 00 00 00 ... 7F FF FF FF

		Length		UInt_32	4	LMS1xx LMS5xx	0 .. 4294967295	00 00 00 00 ... 7F FF FF FF
		Maximum speed		Int_16	2	LMS1xx LMS5xx	0 .. 30000	00 00 ... 7530
		Maximum length		UInt_32	4	LMS1xx LMS5xx	0 .. 4294967295	00 00 00 00 ... 7F FF FF FF
Version number				UInt_16	2	LMS1xx LMS5xx	0 .. 65535	00 00 ... FF FF
Length of field name				UInt_16	2	LMS1xx LMS5xx	0 .. 32	00 00..00 20
Field name				String	0...32	LMS1xx LMS5xx	FIELD1	46 49 45 4c 44 31
Length of comment			If no comment, also no length with a string type ...	UInt_16	2	LMS1xx LMS5xx	0 .. 128	00 00..00 80
Comment				String	0...128	LMS1xx LMS5xx		

Table 442: Example: sAN mLFEgetField

Example: sAN mLFEgetField

CoLa A	ASCII	<STX>sAN{SPC}mLFEgetField{SPC}40000000{SPC}00000000{SPC}1388{SPC}FFFF3CB0{SPC}2{SPC}1{SPC}1{SPC}4{SPC}12{SPC}FFFF{SPC}F7{SPC}72{SPC}FFFF{SPC}18B{SPC}12D{SPC}FFFF{SPC}233{SPC}16D{SPC}FFFF{SPC}1DC{SPC}0{SPC}0{SPC}0{SPC}1{SPC}6{SPC}FIELD1{SPC}0<ETX>
	Hex	02 73 41 4E 20 6D 4C 46 45 67 65 74 46 69 65 6C 64 20 34 30 30 30 30 30 30 30 20 30 30 30 30 30 30 30 30 30 30 20 31 33 38 38 20 46 46 46 46 33 43 42 30 20 32 20 31 20 31 20 34 20 31 32 20 46 46 46 46 20 46 37 20 37 32 20 46 46 46 46 20 31 38 42 20 31 32 44 20 46 46 46 46 20 32 33 33 20 31 36 44 20 46 46 46 46 20 31 44 43 20 30 20 30 20 30 20 31 20 36 20 46 49 45 4C 44 31 20 30 03
CoLa B	Binary	02 02 02 02 00 00 00 51 73 41 4E 20 6D 4C 46 45 67 65 74 46 69 65 6C 64 20 40 00 00 00 00 00 00 00 00 00 13 88 FF FF 3C B0 02 01 00 01 00 04 00 12 FF FF 00 F7 00 72 FF FF 01 8B 01 2D FF FF 02 33 01 6D FF FF 01 DC 00 00 00 00 00 00 00 01 00 06 46 49 45 4C 44 31 00 00 FE

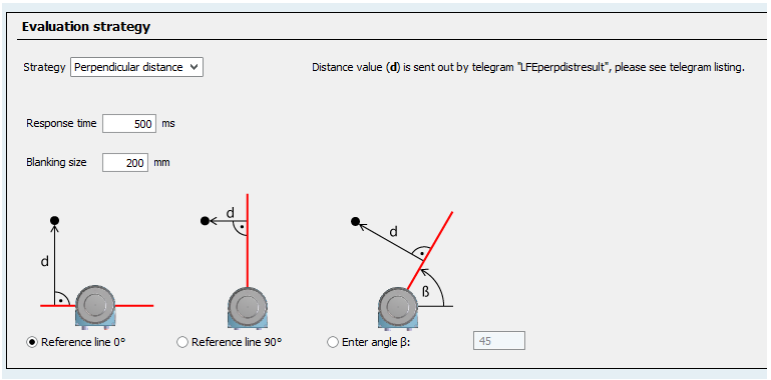
Table 443: Example: sAN mLFEgetField

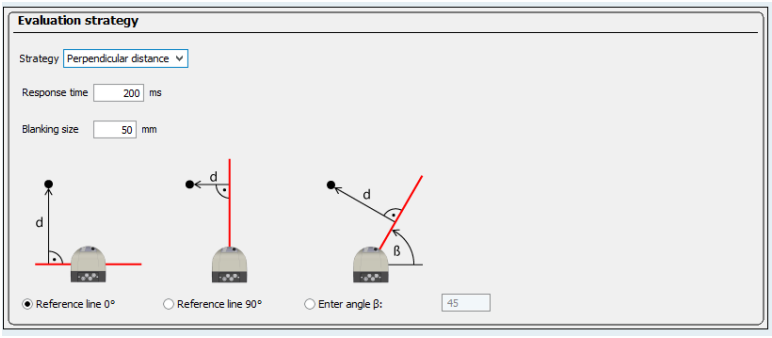
4.10.3 Request perpendicular distance once

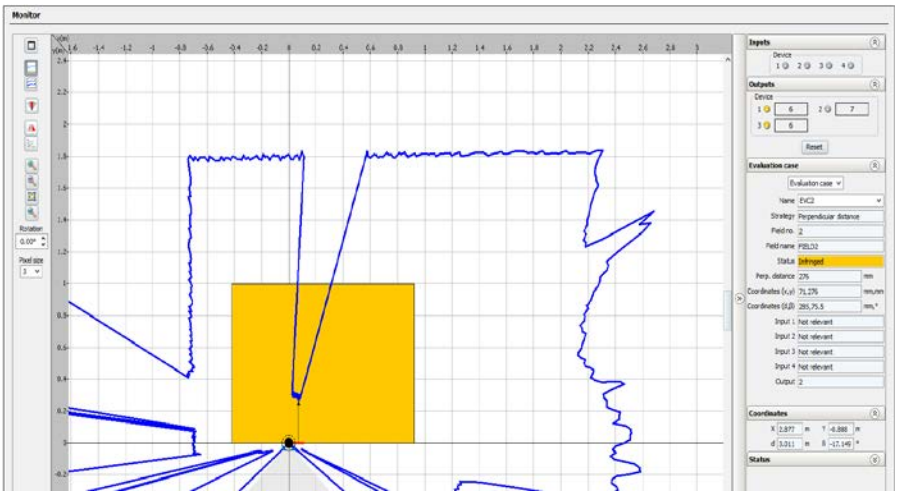
Precondition

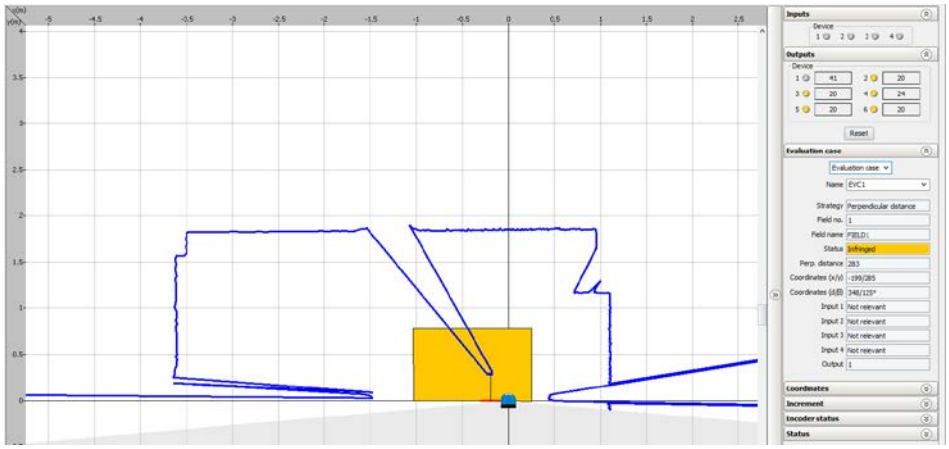
The evaluation stategy “Perpendicular distance” has to be activated in the engineering tool SOPAS. After you get the perpendicular distance displayed in SOPAS as shown below, you are able to read out the telegram.

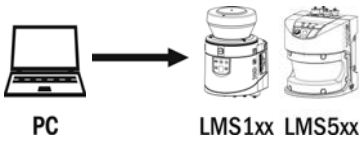
Set the EVC to evaluation strategy “Perpendicular distance” and choose the field which you like to read out. Also keep in mind, that the field evalutation will only be activated by choosing an output.











Telegram structure: sRN LFEperpdistresult (no required user level)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sRN	73 52 4E
Command	Only one telegram	String	14	All	LFEperpdistresult	4C 46 45 70 65 72 70 64 69 73 74 72 65 73 75 6C 74

Table 444: Telegram structure: sRN LFEperpdistresult

Example: sRN LFEperpdistresult

CoLa A	ASCII	<STX>sRN{SPC}LFEperpdistresult<ETX>
	Hex	02 73 52 4E 20 4C 46 45 70 65 72 70 64 69 73 74 72 65 73 75 6C 74 03
CoLa B	Binary	02 02 02 02 00 00 00 15 73 52 4E 20 4C 46 45 70 65 72 70 64 69 73 74 72 65 73 75 6C 74 14

Table 445: Example: sRN LFEperpdistresult

Telegram structure: sRA LFEperpdistresult (no required user level)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sRA	73 52 41
Command	Data telegram	String	14	LMS1xx LMS5XX	LFEperpdistresult	4C 46 45 70 65 72 70 64 69 73 74 72 65 73 75 6C 74
Array	1-10	UInt_8	1	LMS1xx LMS5XX	0h ... Ah (0d ...10d)	01 ... 0A
EVC number	1-10	UInt_8	1	LMS1xx LMS5XX	0h ... Ah (0d ...10d)	01 ... 0A
Perpendicular distance	LMS100,LMS111 0 m ... 20 m LMS151 0 m ... 50 m LMS5xx: 0 m ... 80 m	UInt_32	4	LMS1xx LMS5XX	LMS100, LMS111: 0h ...4E20h (0..20000d) LMS151: 0h...C350h (0...50000d) LMS5xx: 0h ... 13880h (0d ... 80000d)	LMS100, LMS111: 00 00 00 00 ...00 00 4E 20 LMS151: 00 00 00 00...00 00 C3 50 LMS5xx: 00 00 00 00 ... 00 01 38 80
Reserved	default 0	UInt_32	4	LMS1xx LMS5XX	30	00 00 00 00
Reserved	default 0	UInt_32	4	LMS1xx LMS5XX	30	00 00 00 00
X-Pos [mm]	LMS100,LMS111 0 m ... 20 m LMS151 0 m ... 50 m LMS5xx: 0 m ... 80 m	Int_32	4	LMS1xx LMS5XX	LMS100, LMS111: FFFFB1E0 h ...4E20h (-20000d..20000d) LMS151: FFFF3CB0h...C350h (-50000d...50000d) LMS5xx: FFFEC780h ... 13880h (-80000d ... 80000d)	FF FF B1 E0 ... 00 00 4E 20 FF FF 3C B0 ... 00 00 C3 50 FF FE C7 80 ... 00 01 38 80
Y-Pos [mm]	LMS100,LMS111	Int_32	4	LMS1xx	LMS100, LMS111:	FF FF B1 E0 ... 00 00 4E 20

Telegram structure: sRA LFEperpdistresult (no required user level)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
	0 m ... 20 m LMS151 0 m ... 50 m LMS5xx: 0 m ... 80 m			LMS5XX	FFFFB1E0 h ...4E20h (-20000d..20000d) LMS151: FFFF3CB0h...C350h (-50000d...50000d) LMS5xx: FFFEC780h ... 13880h (-80000d ... 80000d)	FF FF 3C B0 ... 00 00 C3 50 FF FE C7 80 ... 00 01 38 80

Telegram structure: sRA LFEperpdistresult (no required user level)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Reserved	default 0	Int_32	4	LMS1xx LMS5XX	30	00 00 00 00
Radial distance	LMS100,LMS111 0 m … 20 m LMS151 0 m … 50 m LMS5xx: 0 m … 80 m	UInt_32	4	LMS1xx LMS5XX	LMS100, LMS111: 0h …4E20h (0..20000d) LMS151: 0h…C350h (0…50000d) LMS5xx: 0h … 13880h (0d … 80000d)	LMS100, LMS111: 00 00 00 00 …00 00 4E 20 LMS151: 00 00 00 00…00 00 C3 50 LMS5xx: 00 00 00 00 … 00 01 38 80
Radial distance angle	LMS1xx: -45,0000°… 225,0000° LMS5xx: -5,0000°… 185,0000°	Int_32	4	LMS1xx LMS5XX	LMS1xx: FFF92230h … 225510h (-450000d … 225,0000d) LMS5xx: FFFF3CB0h … 001C3A90h 50000d … 1850000d	LMS1xx: FF F9 22 30 … 00 22 55 10 LMS5xx: FF FF 3C B0 … 00 1C 3A 90
Reserved	default 0	Int_32	4	LMS1xx LMS5XX	30	00 00 00 00
Reference line angle	LMS1xx: -45,0000°… 225,0000° LMS5xx: -5,0000°… 185,0000°	Int_32	4	LMS1xx LMS5XX	LMS1xx: FFF92230h … 225510h (-450000d … 225,0000d) LMS5xx: FFFF3CB0h … 001C3A90h 50000d … 1850000d	LMS1xx: FF F9 22 30 … 00 22 55 10 LMS5xx: FF FF 3C B0 … 00 1C 3A 90
Reserved	default 0	Int_32	4	LMS1xx LMS5XX	30	00 00 00 00
Reserved	default 0	Int_32	4	LMS1xx LMS5XX	30	00 00 00 00
Reserved	default 0	Int_32	4	LMS1xx LMS5XX	30	00 00 00 00
Reserved	default 0	Int_32	4	LMS1xx LMS5XX	30	00 00 00 00
Reserved	default 0	Int_32	4	LMS1xx LMS5XX	30	00 00 00 00
Reserved	default 0	Int_32	4	LMS1xx LMS5XX	30	00 00 00 00
Reserved	default 0	Int_32	4	LMS1xx	30	00 00 00 00

Telegram structure: sRA LFEperpdistresult (no required user level)						
Telegram part	Description	Variable	Length	Sensor	Values CoLa A (ASCII)	Values CoLa B (Binary)
				LMS5XX		
Reserved	default 0	Int_32	4	LMS1xx LMS5XX	30	00 00 00 00

Table 446: Telegram structure: sRA LFEperpdistresult

Example: sRA LFEperpdistresult

CoLa A	ASCII	<STX>sRA{SPC}LFEperpdistresult{SPC}1{SPC}2{SPC}21A{SPC}0{SPC}0{SPC}0{SPC}21A{SPC}0{SPC}21A{SPC}DBBA0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0<ETX>
	Hex	02 73 52 41 20 4C 46 45 70 65 72 70 64 69 73 74 72 65 73 75 6C 74 20 31 20 32 20 32 31 41 20 30 20 30 20 30 20 32 31 41 20 30 20 32 31 41 20 44 42 42 41 30 20 30 20 30 20 30 20 30 20 30 20 30 20 30 20 30 20 30 20 30 03
CoLa B	Binary	02 02 02 02 00 00 00 61 73 52 41 20 4C 46 45 70 65 72 70 64 69 73 74 72 65 73 75 6C 74 20 00 01 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 38

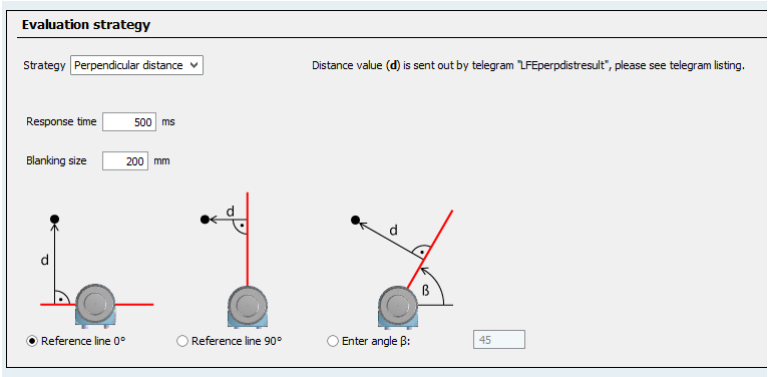
Table 447: Example: sRA LFEperpdistresult

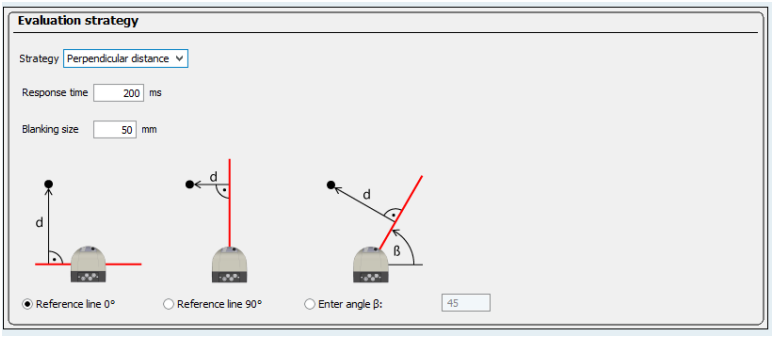
4.10.4 Request perpendicular distance continiously on event

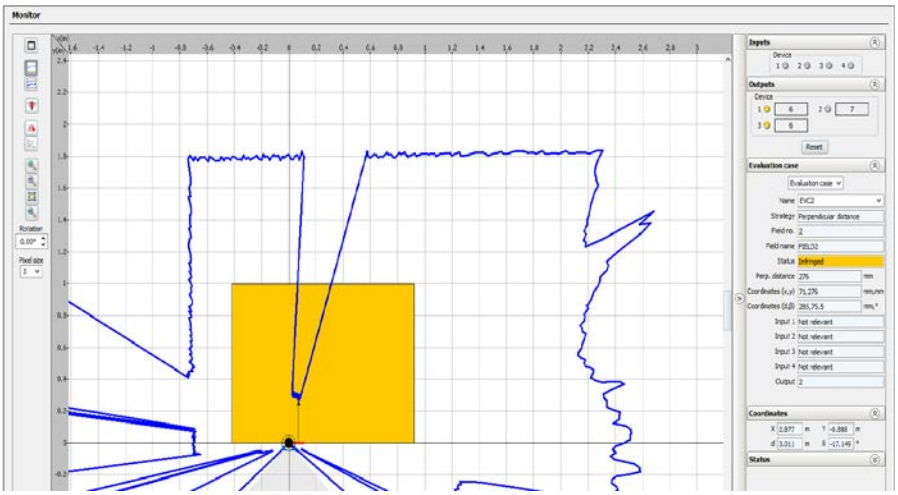
Precondition

The evaluation stategy “Perpendicular distance” has to be activated in the engineering tool SOPAS. After you get the perpendicular distance displayed in SOPAS as shown below, you are able to read out the telegram.

Set the EVC to evaluation strategy “Perpendicular distance” and choose the field which you like to read out. Also keep in mind, that the field evalutation will only be activated by choosing an output.







Telegram structure: sEN LFEperpdistresult (no required user level)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sEN	73 45 4E
Command	Only one telegram	String	14	All	LFEperpdistresult	4C 46 45 70 65 72 70 64 69 73 74 72 65 73 75 6C 74
Reporting	Start/stop	Enum_8	1	LMS1xx LMS5XX	Stop: 0 Start: 1	Stop: 00 Start: 01

Table 448: Telegram structure: sEN LFEperpdistresult

Example: sEN LFEperpdistresult 1

CoLa A	ASCII	<STX> sEN{SPC}LFEperpdistresult{SPC}1<ETX>
	Hex	02 73 45 4E 20 4C 46 45 70 65 72 70 64 69 73 74 72 65 73 75 6C 74 20 31 03
CoLa B	Binary	LMS1xx: 02 02 02 02 00 00 00 17 73 45 4E 20 4C 46 45 70 65 72 70 64 69 73 74 72 65 73 75 6C 74 20 01 22 LMS5xx: 02 02 02 02 00 00 00 17 73 45 4E 20 4C 46 45 70 65 72 70 64 69 73 74 72 65 73 75 6C 74 20 31 12

Table 449: Example: sEN LFEperpdistresult 1

Telegram structure: sEA LFEperpdistresult (no required user level)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sEA	73 45 41
Command	Only one telegram	String	14	All	LFEperpdistresult	4C 46 45 70 65 72 70 64 69 73 74 72 65 73 75 6C 74
Reporting	Start/stop	Enum_8	1	LMS1xx LMS5XX	Stop: 0 Start: 1	Stop: 00 Start: 01

Table 450: Telegram structure: sEA LFEperpdistresult

Example: sEA LFEperpdistresult 1

CoLa A	ASCII	<STX> sEA{SPC}LFEperpdistresult{SPC}1<ETX>
	Hex	02 73 45 41 20 4C 46 45 70 65 72 70 64 69 73 74 72 65 73 75 6C 74 20 31 03
CoLa B	Binary	LMS1xx: 02 02 02 02 00 00 00 17 73 45 41 20 4C 46 45 70 65 72 70 64 69 73 74 72 65 73 75 6C 74 20 01 2D LMS5xx: 02 02 02 02 00 00 00 17 73 45 41 20 4C 46 45 70 65 72 70 64 69 73 74 72 65 73 75 6C 74 20 31 1D

Table 451: Example: sEA LFEperpdistresult 1

NOTE

The answer to the telegram will be followed by data that is sent on event.

The sensor only sends the following answer if there are perpendicular distance values calculated within the application.

Telegram structure: sSN LFEperpdistresult (no required user level)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sSN	73 53 4E
Command	Data telegram	String	14	LMS1xx LMS5XX	LFEperpdistresult	4C 46 45 70 65 72 70 64 69 73 74 72 65 73 75 6C 74
Array	1-10	UInt_8	1	LMS1xx LMS5XX	0h ... Ah (0d ...10d)	01 ... 0A
EVC number	1-10	UInt_8	1	LMS1xx LMS5XX	0h ... Ah (0d ...10d)	01 ... 0A
Perpendicular distance	LMS100,LMS111 0 m ... 20 m LMS151 0 m ... 50 m LMS5xx: 0 m ... 80 m	UInt_32	4	LMS1xx LMS5XX	LMS100, LMS111: 0h ...4E20h (0..20000d) LMS151: 0h...C350h (0...50000d) LMS5xx: 0h ... 13880h (0d ... 80000d)	LMS100, LMS111: 00 00 00 00 ...00 00 4E 20 LMS151: 00 00 00 00...00 00 C3 50 LMS5xx: 00 00 00 00 ... 00 01 38 80
Reserved	default 0	UInt_32	4	LMS1xx LMS5XX	30	00 00 00 00
Reserved	default 0	UInt_32	4	LMS1xx LMS5XX	30	00 00 00 00
X-Pos [mm]	LMS100,LMS111 0 m ... 20 m LMS151 0 m ... 50 m LMS5xx: 0 m ... 80 m	Int_32	4	LMS1xx LMS5XX	LMS100, LMS111: FFFFB1E0 h ...4E20h (-20000d..20000d) LMS151: FFFF3CB0h...C350h (-50000d...50000d) LMS5xx: FFFEC780h ... 13880h (-80000d ... 80000d)	FF FF B1 E0 ... 00 00 4E 20 FF FF 3C B0 ... 00 00 C3 50 FF FE C7 80 ... 00 01 38 80
Y-Pos [mm]	LMS100,LMS111	Int_32	4	LMS1xx	LMS100, LMS111:	FF FF B1 E0 ... 00 00 4E 20

Telegram structure: sSN LFEperpdistresult (no required user level)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
	0 m ... 20 m LMS151 0 m ... 50 m LMS5xx: 0 m ... 80 m			LMS5XX	FFFFB1E0 h ...4E20h (-20000d..20000d) LMS151: FFFF3CB0h...C350h (-50000d...50000d) LMS5xx: FFFEC780h ... 13880h (-80000d ... 80000d)	FF FF 3C B0 ... 00 00 C3 50 FF FE C7 80 ... 00 01 38 80
Reserved	default 0	Int_32	4	LMS1xx LMS5XX	30	00 00 00 00
Radial distance	LMS100,LMS111 0 m ... 20 m LMS151 0 m ... 50 m LMS5xx: 0 m ... 80 m	UInt_32	4	LMS1xx LMS5XX	LMS100, LMS111: 0h ...4E20h (0..20000d) LMS151: 0h...C350h (0...50000d) LMS5xx: 0h ... 13880h (0d ... 80000d)	LMS100, LMS111: 00 00 00 00 ...00 00 4E 20 LMS151: 00 00 00 00...00 00 C3 50 LMS5xx: 00 00 00 00 ... 00 01 38 80
Radial distance angle	LMS1xx: -45,0000°... 225,0000° LMS5xx: -5,0000°... 185,0000°	Int_32	4	LMS1xx LMS5XX	LMS1xx: FFF92230h ... 225510h (-450000d ... 225,0000d) LMS5xx: FFFF3CB0h ... 001C3A90h 50000d ... 1850000d	LMS1xx: FF F9 22 30 ... 00 22 55 10 LMS5xx: FF FF 3C B0 ... 00 1C 3A 90
Reserved	default 0	Int_32	4	LMS1xx LMS5XX	30	00 00 00 00
Reference line angle	LMS1xx: -45,0000°... 225,0000° LMS5xx: -5,0000°... 185,0000°	Int_32	4	LMS1xx LMS5XX	LMS1xx: FFF92230h ... 225510h (-450000d ... 225,0000d) LMS5xx: FFFF3CB0h ... 001C3A90h 50000d ... 1850000d	LMS1xx: FF F9 22 30 ... 00 22 55 10 LMS5xx: FF FF 3C B0 ... 00 1C 3A 90
Reserved	default 0	Int_32	4	LMS1xx LMS5XX	30	00 00 00 00
Reserved	default 0	Int_32	4	LMS1xx LMS5XX	30	00 00 00 00
Reserved	default 0	Int_32	4	LMS1xx	30	00 00 00 00

Telegram structure: sSN LFEperpdistresult (no required user level)						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
				LMS5XX		
Reserved	default 0	Int_32	4	LMS1xx LMS5XX	30	00 00 00 00
Reserved	default 0	Int_32	4	LMS1xx LMS5XX	30	00 00 00 00
Reserved	default 0	Int_32	4	LMS1xx LMS5XX	30	00 00 00 00
Reserved	default 0	Int_32	4	LMS1xx LMS5XX	30	00 00 00 00
Reserved	default 0	Int_32	4	LMS1xx LMS5XX	30	00 00 00 00

Table 452: Telegram structure: sSN LFEperpdistresult

Example: sSN LFEperpdistresult

CoLa A	ASCII	<STX>sSN{SPC}LFEperpdistresult{SPC}1{SPC}2{SPC}B0{SPC}0{SPC}0{SPC}FFFFFFE8{SPC}B0{SPC}0{SPC}B2{SPC}EF420{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0<ETX>
	Hex	02 73 53 4E 20 4C 46 45 70 65 72 70 64 69 73 74 72 65 73 75 6C 74 20 31 20 32 20 42 30 20 30 20 30 20 46 46 46 46 46 46 45 38 20 42 30 20 30 20 42 32 20 45 46 34 32 30 20 30 20 30 20 30 20 30 20 30 20 30 20 30 20 30 20 30 20 30 03
CoLa B	Binary	02 02 02 02 00 00 00 61 73 53 4E 20 4C 46 45 70 65 72 70 64 69 73 74 72 65 73 75 6C 74 20 00 01 01 00 00 01 5E 00 00 00 00 00 00 00 00 00 00 00 4A 00 00 01 5E 00 00 00 00 00 00 01 66 00 0B E6 E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 15

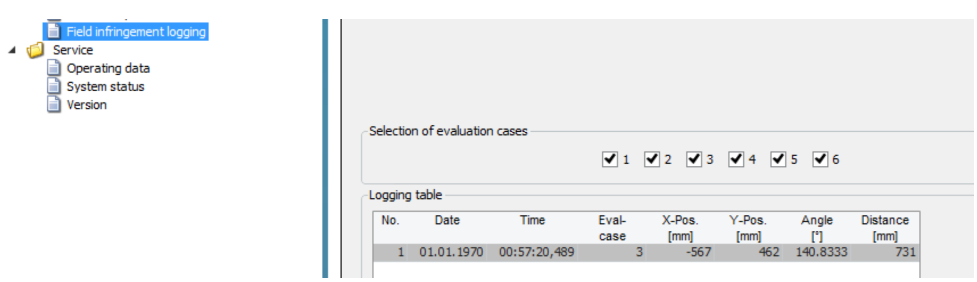
Table 453: Example: sSN LFEperpdistresult

4.10.5 Request latest field infringement info

Note

The command is used to request entries from the "field infringement logging" via telegram. Using this command, only the latest field infringement is given out. The answer telegram includes information regarding date and time of the infringement as well as the associated EVC and infringement position.

Please note: Only activated EVCs will be considered. An EVC is only valid, when an output is defined in the EVC. Using this telegram, an EVC without a defined output is not taken into considaration.



Telegram structure: sRN LFEinfringementinfo						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Read	String	3	All	sRN	73 52 4E
Command	Only one telegram	String	19	LMS1xx LMS5XX	LFEinfringementinfo	4C 46 45 69 6E 66 72 69 6E 67 65 6D 65 6E 74 69 6E 66 6F

Table 454: Telegram structure: sRN LFEinfringementinfo

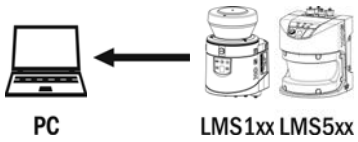
Example: sRN LFEinfringementinfo

CoLa A	ASCII	<STX>sRN{SPC}LFEinfringementinfo<ETX>
	Hex	02 73 52 4E 20 4C 46 45 69 6E 66 72 69 6E 67 65 6D 65 6E 74 69 6E 66 6F 03
CoLa B	Binary	02 02 02 02 00 00 00 17 73 52 4E 20 4C 46 45 69 6E 66 72 69 6E 67 65 6D 65 6E 74 69 6E 66 6F 0A

Table 455: Example: sEN LFEinfringementinfo

NOTE

The answer telegram refers to the latest recorded field infringement. Therefore the device outputs the latest field infringement entry from the logging table.



Telegram structure: sRA LFEinfringementinfo							
Telegram part		**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type		Read	String	3	All	sRA	73 52 41
Command		Data telegram	String	19	LMS1XX LMS5XX	LFEinfringementinfo	4C 46 45 69 6E 66 72 69 6E 67 65 6D 65 6E 74 69 6E 66 6F
Time Info	Counter	continious counter of infrigements			LMS1XX LMS5XX	0000h ... 270Fh (0d...9999d)	00 00 ... 27 0F
	Year		Uint_16	2	LMS1XX LMS5XX	0000h ... 270Fh (0d...9999d)	00 00 ... 27 0F
	Month	1 to 12	Uint_8	1	LMS1XX LMS5XX	00h ... 0Ch (0d ... 12d)	00 ... 0C

Telegram structure: sRA LFEinfringementinfo							
Telegram part		Description	Variable	Length	Sensor	Values CoLa A (ASCII)	Values CoLa B (Binary)
	Day	Day of month 1 to 31	Uint_8	1	LMS1XX LMS5XX	00h ... 1Fh (0d ... 31d)	00 ... 1F
	Hour	0 to 23	Uint_8	1	LMS1XX LMS5XX	00h ... 17h (0d ... 23d)	00 ... 17
	Minute	0 to 59	Uint_8	1	LMS1XX LMS5XX	00h ... 3Bh 0d ... 59d	00 ... 3B
	Second	0 to 59	Uint_8	1	LMS1XX LMS5XX	00h ... 3Bh (0d ... 59d)	00 ... 3B
	µSecond	0 to 999999	Uint_32	4	LMS1XX LMS5XX	00000000h ... 000F423Fh (0d ... 999999d)	00 00 00 00 ... 00 0F 42 3F
Infrigment-Info	EVC number	1-10	UInt_8	1	LMS1XX LMS5XX	0h ... Ah (0d ...10d)	01 ... 0A
	X-Pos [mm]	LMS100,LMS111 -20 m ... 20 m LMS151 -50 m ... 50 m LMS5xx -80 m ... 80 m	Int_32	4	LMS1XX LMS5XX	LMS100, LMS111: FFFFB1E0 h ...4E20h (-20000d..20000d) LMS151: FFFF3CB0h...C350h (-50000d...50000d) LMS5xx FFFEC780h ... 13880h (-80000d ... 80000d)	FF FF B1 E0 ... 00 00 4E 20 FF FF 3C B0 ... 00 00 C3 50 FF FE C7 80 ... 00 01 38 80
	Y-Pos [mm]	LMS100,LMS111 -20 m ... 20 m LMS151 -50 m ... 50 m LMS5xx -80 m ... 80 m	Int_32	4	LMS1XX LMS5XX	LMS100, LMS111: FFFFB1E0 h ...4E20h (-20000d..20000d) LMS151: FFFF3CB0h...C350h (-50000d...50000d) LMS5xx FFFEC780h ... 13880h (-80000d ... 80000d)	FF FF B1 E0 ... 00 00 4E 20 FF FF 3C B0 ... 00 00 C3 50 FF FE C7 80 ... 00 01 38 80

Telegram structure: sRA LFEinfringementinfo							
Telegram part		Description	Variable	Length	Sensor	Values CoLa A (ASCII)	Values CoLa B (Binary)
	Distance [mm]	LMS100,LMS111 0 m ... 20 m LMS151 0 m ... 50 m LMS5xx 0 m ... 80 m	UInt_32	4	LMS1XX LMS5XX	LMS100, LMS111: 0 h ...4E20h (0d..20000d) LMS151: 0h...C350h (0d...50000d) LMS5xx 0h ... 13880h (0d ... 80000d)	00 00 00 00 ... 00 00 4E 20 00 00 00 00 ... 00 00 C3 50 00 00 00 00 ... 00 01 38 80
	Angle [°]	LMS1xx -450000 ... 2250000 LMS5xx -5,0000°... 185,0000°	Int_32	4	LMS1XX LMS5XX	LMS1xx FFF92230 h... 00225510 h (-450000d ... 2250000d) LMS5xx FFFF3CB0h ... 001C3A90h 50000d ... 1850000d	FF F9 22 30 ... 00 22 55 10 FF FF 3C B0 ... 00 1C 3A 90
	Angle of Layer [°]	LMS1xx -450000 ... 2250000 LMS5xx -5,0000°... 185,0000°	Int_32	4	LMS1XX LMS5XX	LMS1xx FFF92230 h... 00225510 h (-450000d ... 2250000d) LMS5xx FFFF3CB0h ... 001C3A90h 50000d ... 1850000d	FF F9 22 30 ... 00 22 55 10 FF FF 3C B0 ... 00 1C 3A 90
	Object Size [mm]	LMS100,LMS111 0 m ... 20 m LMS151 0 m ... 50 m LMS5xx 0 m ... 80 m	UInt_32	4	LMS1XX LMS5XX	LMS100, LMS111: 0 h ...4E20h (0d..20000d) LMS151: 0h...C350h (0d...50000d) LMS5xx 0h ... 13880h (0d ... 80000d)	00 00 00 00 ... 00 00 4E 20 00 00 00 00 ... 00 00 C3 50 00 00 00 00 ... 00 01 38 80

Table 456: Example: sRA LFEinfringementinfo

Example: sRA LFEinfringementinfo

CoLa A	ASCII	<STX>sRA{SPC}LFEinfringementinfo{SPC}4{SPC}7B2{SPC}1{SPC}1{SPC}2{SPC}32{SPC}1F{SPC}3A1B0{SPC}1{SPC}FFFFFDAE{SPC}D6{SPC}277{SPC}187083{SPC}4A<ETX>
	Hex	02 73 52 41 20 4C 46 45 69 6E 66 72 69 6E 67 65 6D 65 6E 74 69 6E 66 6F 20 34 20 37 42 32 20 31 20 31 20 32 20 33 32 20 31 46 20 33 41 31 42 30 20 31 20 46 46 46 46 46 44 41 45 20 44 36 20 32 37 37 20 31 38 37 30 38 33 20 34 41 03
CoLa B	Binary	02 02 02 02 00 00 00 3A 73 52 41 20 4C 46 45 69 6E 66 72 69 6E 67 65 6D 65 6E 74 69 6E 66 6F 20 00 04 07 B2 01 01 02 32 1F 00 03 A1 B0 01 FF FF FD AE 00 00 00 D6 00 00 02 77 00 18 70 83 00 00 00 4A F9

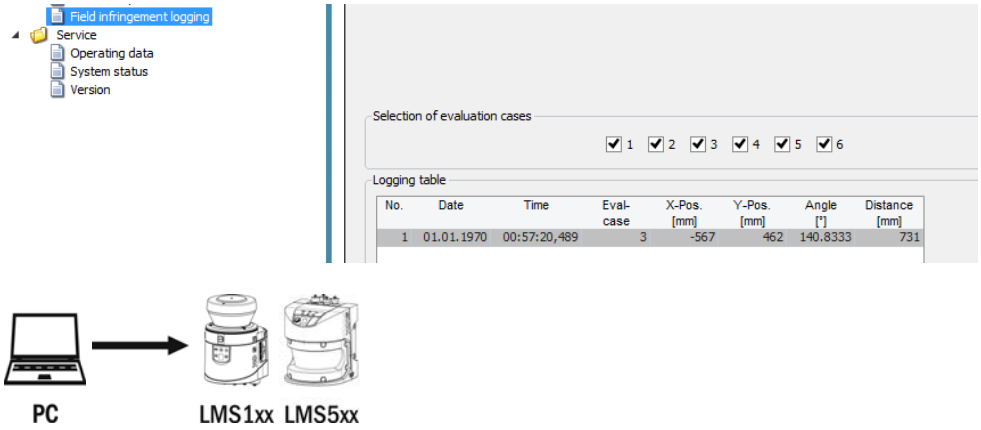
Table 457: Example: sRA LFEinfringementinfo

4.10.6 Request field infringement info continiously on event

Note

The command is used to request entries from the “field infringement logging” via telegram. Using this command, a telegram will be sent from the sensor on event (in case of a new field infringement). The answer telegram includes information regarding date and time of the infringement as well as the associated EVC and infringement position.

Please note: Only activated EVCs will be considered. An EVC is only valid, when an output is defined in the EVC. Using this telegram, an EVC without a defined output is not taken into considaration.



Telegram structure: sEN LFEinfringementinfo						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Event	String	3	All	sEN	73 45 4E
Command	Data telegram	String	19	LMS1XX LMS5XX	LFEinfringementinfo	4C 46 45 69 6E 66 72 69 6E 67 65 6D 65 6E 74 69 6E 66 6F
Reporting	Start/stop	Enum_8	1	LMS1XX LMS5XX	Stop: 0 Start: 1	LMS1xx: Stop: 00 Start: 01 LMS5xx: Stop: 30 Start: 31

Table 458: Telegram structure: sEN LFEinfringementinfo

Example: sEN LFEinfringementinfo 1

CoLa A	ASCII	<STX>sEN{SPC}LFEinfringementinfo{SPC}1<ETX>
	Hex	02 73 45 4E 20 4C 46 45 69 6E 66 72 69 6E 67 65 6D 65 6E 74 69 6E 66 6F 20 31 03
CoLa B	Binary	02 02 02 02 00 00 00 19 73 45 4E 20 4C 46 45 69 6E 66 72 69 6E 67 65 6D 65 6E 74 69 6E 66 6F 20 01 3C

Table 459: Example: sEN LFEinfringementinfo 1

Telegram structure: sEA LFEinfringementinfo						
Telegram part	**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type	Event	String	3	All	sEA	73 45 41
Command	Data telegram	String	19	LMS1XX LMS5XX	LFEinfringementinfo	4C 46 45 69 6E 66 72 69 6E 67 65 6D 65 6E 74 69 6E 66 6F
Reporting	Start/stop	Enum_8	1	LMS1XX LMS5XX	Stop: 0 Start: 1	LMS1xx: Stop: 00 Start: 01 LMS5xx: Stop: 30 Start: 31

Table 460: Telegram structure: sEA LFEinfringementinfo

Example: sEA LFEinfringementinfo 1

CoLa A	ASCII	<STX>sEA{SPC}LFEinfringementinfo{SPC}1<ETX>
	Hex	02 73 45 41 20 4C 46 45 69 6E 66 72 69 6E 67 65 6D 65 6E 74 69 6E 66 6F 20 31 03
CoLa B	Binary	02 02 02 02 00 00 00 19 73 45 41 20 4C 46 45 69 6E 66 72 69 6E 67 65 6D 65 6E 74 69 6E 66 6F 20 01 33

Table 461: Example: sEA LFEinfringementinfo 1

NOTE

The answer to the telegram will be followed by data that is sent on event.

The sensor only sends the following data if there is a new field infringement detected and noted in the field infringement logging.

Telegram structure: sSN LFEinfringementinfo							
Telegram part		**Description**	**Variable**	**Length**	**Sensor**	**Values CoLa A (ASCII)**	**Values CoLa B (Binary)**
Command type		Read	String	3	All	sSN	73 53 4E
Command		Data telegram	String	19	LMS1XX LMS5XX	LFEinfringementinfo	4C 46 45 69 6E 66 72 69 6E 67 65 6D 65 6E 74 69 6E 66 6F
Time Info	Counter	continious counter of infrigements			LMS1XX LMS5XX	0000h ... 270Fh (0d...9999d)	00 00 ... 27 0F
	Year		Uint_16	2	LMS1XX LMS5XX	0000h ... 270Fh (0d...9999d)	00 00 ... 27 0F
	Month	1 to 12	Uint_8	1	LMS1XX LMS5XX	00h ... 0Ch (0d ... 12d)	00 ... 0C
	Day	Day of month 1 to 31	Uint_8	1	LMS1XX LMS5XX	00h ... 1Fh (0d ... 31d)	00 ... 1F
	Hour	0 to 23	Uint_8	1	LMS1XX LMS5XX	00h ... 17h (0d ... 23d)	00 ... 17
	Minute	0 to 59	Uint_8	1	LMS1XX LMS5XX	00h ... 3Bh 0d ... 59d	00 ... 3B
	Second	0 to 59	Uint_8	1	LMS1XX LMS5XX	00h ... 3Bh (0d ... 59d)	00 ... 3B
	µSecond	0 to 999999	Uint_32	4	LMS1XX LMS5XX	00000000h ... 000F423Fh (0d ... 999999d)	00 00 00 00 ... 00 0F 42 3F
Infrigment-Info	EVC number	1-10	UInt_8	1	LMS1XX LMS5XX	0h ... Ah (0d ...10d)	01 ... 0A
	X-Pos [mm]	LMS100,LMS111 -20 m ... 20 m LMS151 -50 m ... 50 m LMS5xx -80 m ... 80 m	Int_32	4	LMS1XX LMS5XX	LMS100, LMS111: FFFFB1E0 h ...4E20h (-20000d..20000d) LMS151: FFFF3CB0h...C350h (-50000d...50000d) LMS5xx FFFEC780h ... 13880h (-80000d ... 80000d)	FF FF B1 E0 ... 00 00 4E 20 FF FF 3C B0 ... 00 00 C3 50 FF FE C7 80 ... 00 01 38 80

Telegram structure: sSN LFEinfringementinfo							
Telegram part		Description	Variable	Length	Sensor	Values CoLa A (ASCII)	Values CoLa B (Binary)
	Y-Pos [mm]	LMS100,LMS111 -20 m ... 20 m LMS151 -50 m ... 50 m LMS5xx -80 m ... 80 m	Int_32	4	LMS1XX LMS5XX	LMS100, LMS111: FFFFB1E0 h ...4E20h (-20000d..20000d) LMS151: FFFF3CB0h...C350h (-50000d...50000d) LMS5xx FFFEC780h ... 13880h (-80000d ... 80000d)	FF FF B1 E0 ... 00 00 4E 20 FF FF 3C B0 ... 00 00 C3 50 FF FE C7 80 ... 00 01 38 80
	Distance [mm]	LMS100,LMS111 0 m ... 20 m LMS151 0 m ... 50 m LMS5xx 0 m ... 80 m	UInt_32	4	LMS1XX LMS5XX	LMS100, LMS111: 0 h ...4E20h (0d..20000d) LMS151: 0h...C350h (0d...50000d) LMS5xx 0h ... 13880h (0d ... 80000d)	00 00 00 00 ... 00 00 4E 20 00 00 00 00 ... 00 00 C3 50 00 00 00 00 ... 00 01 38 80
	Angle [°]	LMS1xx -450000 ... 2250000 LMS5xx -5,0000°... 185,0000°	Int_32	4	LMS1XX LMS5XX	LMS1xx FFF92230 h... 00225510 h (-450000d ... 2250000d) LMS5xx FFFF3CB0h ... 001C3A90h 50000d ... 1850000d	FF F9 22 30 ... 00 22 55 10 FF FF 3C B0 ... 00 1C 3A 90
	Angle of Layer [°]	LMS1xx -450000 ... 2250000 LMS5xx -5,0000°... 185,0000°	Int_32	4	LMS1XX LMS5XX	LMS1xx FFF92230 h... 00225510 h (-450000d ... 2250000d) LMS5xx FFFF3CB0h ... 001C3A90h 50000d ... 1850000d	FF F9 22 30 ... 00 22 55 10 FF FF 3C B0 ... 00 1C 3A 90
	Object Size [mm]	LMS100,LMS111 0 m ... 20 m LMS151 0 m ... 50 m LMS5xx 0 m ... 80 m	UInt_32	4	LMS1XX LMS5XX	LMS100, LMS111: 0 h ...4E20h (0d..20000d) LMS151: 0h...C350h (0d...50000d) LMS5xx 0h ... 13880h (0d ... 80000d)	00 00 00 00 ... 00 00 4E 20 00 00 00 00 ... 00 00 C3 50 00 00 00 00 ... 00 01 38 80

Table 462: Example: sSN LFEinfringementinfo

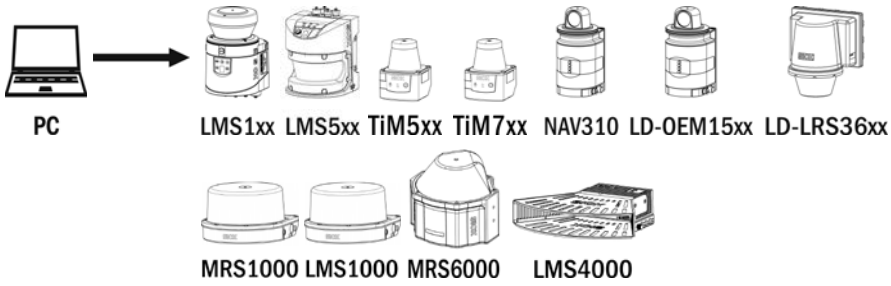
Example: sSN LFEinfringementinfo

CoLa A	ASCII	<STX>sSN{SPC}LFEinfringementinfo{SPC}6{SPC}7B2{SPC}1{SPC}1{SPC}0{SPC}19{SPC}21{SPC}9D788{SPC}1{SPC}41{SPC}C3{SPC}0{SPC}CE{SPC}AE8F8{SPC}0{SPC}A8<ETX>
	Hex	02 73 53 4E 20 4C 46 45 69 6E 66 72 69 6E 67 65 6D 65 6E 74 69 6E 66 6F 20 36 20 37 42 32 20 31 20 31 20 30 20 31 39 20 32 31 20 39 44 37 38 38 20 31 20 34 31 20 43 33 20 30 20 43 45 20 41 45 38 46 38 20 30 20 41 38 03
CoLa B	Binary	02 02 02 02 00 00 00 40 73 53 4E 20 4C 46 45 69 6E 66 72 69 6E 67 65 6D 65 6E 74 69 6E 66 6F 20 00 06 07 B2 01 01 00 19 21 00 09 D7 88 01 00 00 00 41 00 00 00 C3 00 00 00 00 00 00 00 CE 00 0A E8 F8 00 00 00 00 00 A8 09

Table 463: Example: sSN LFEinfringementinfo

5 Diagnostics

5.1 SOPAS error codes



sFA ErrorCode

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
Sopas_Error_FLEX_OUT_OF_BOUNDS	An array was accessed over its maximum length.	14	E

Telegram structure: sFA ErrorCode			
Error code	**Description**	**Dec.**	**Hex.**
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 464: SOPAS error codes

Example: sFA ErrorCode Wrong userlevel

CoLa A	ASCII	<STX>sFA{SPC}01<ETX>
	Hex	02 73 46 41 20 30 31 03
CoLa B	Binary	02 02 02 02 00 00 00 05 73 46 41 20 01 55

Table 465: Example: sFA ErrorCode Wrong userlevel

5.2 Additional information

Every response telegram starts with a separat framed string:

<STX>sSI 2 1<ETX><STX>“Answer”<ETX>

If it is an event from SOPAS, send command: <STX>sEN SCParmChngd 0<ETX> to deactivate that event.

6 List of tables

Table 1: Example: Binary telegram 7
Table 2: Example: ASCII telegram 8
Table 3: Telegram structure: sMN SetAccessMode 12
Table 4: Example: sMN SetAccessMode 12
Table 5: Telegram structure: sAN SetAccessMode 13
Table 6: Example for LMS100: sAN SetAccessMode 13
Table 7: Telegram structure: sMN mLMPsetscancfg 16
Table 8: Example: sMN mLMPsetscancfg for LMS1xx with 1 measurement sector of 270 ° 17
Table 9: Example: sMN mLMPsetscancfg for LD-XXX###1 with 1 measurement sector of 360 ° 18
Table 10: Example: sMN mLMPsetscancfg for LD-XXX###1 with 1 measurement sector of 270 ° 18
Table 11: Example: sMN mLMPsetscancfg for LD-XXX###1 with 2 measurement sectors 19
Table 12: Example: sMN mLMPsetscancfg for LD-XXX###1 with 4 measurement sectors 19
Table 13: Example: sMN mLMPsetscancfg for LD-XXX###0 with 1 measurement sector of 360 ° 20
Table 14: Example: sMN mLMPsetscancfg for LD-XXX###0 with 1 measurement sector of 270 ° 20
Table 15: Example: sMN mLMPsetscancfg for LD-XXX###0 with 2 measurement sectors 21
Table 16: Example: sMN mLMPsetscancfg for LD-XXX###0 with 4 measurement sectors 21
Table 17: Telegram structure: sAN mLMPsetscancfg 23
Table 18: Example: sAN mLMPsetscancfg 23
Table 19: Telegram structure: sRN LMPscancfg 24
Table 20: Example for LMS100: sRN LMPscancfg 24
Table 21: Telegram structure: sRA LMPscancfg 26
Table 22: Example: sRA LMPscancfg 26
Table 23: Telegram structure: sWN MMAlignmentMode 27
Table 24: Example: sWN MMAlignmentMode 27
Table 25: Telegram structure: sWA MMAlignmentMode 27
Table 26: Example: sWA MMAlignmentMode 28
Table 27: Telegram structure: sMN mCLsetscancfglist 28
Table 28: Interlace mode for sMN mCLsetscancfglist 29
Table 29: Example: Set scan configuration 1: sMN mCLsetscancfglist 1 29
Table 30: Telegram structure: sAN mCLsetscancfglist 30
Table 31: Example: sAN mCLsetscancfglist Ok 30

Table 32: Telegram structure: sMN mCLsetscancfglist 31
Table 33: Example: sMN mCLsetscancfglist 5 31
Table 34: Telegram structure: sAN mCLsetscancfglist 32
Table 35: Example: sAN mCLsetscancfglist 0 32
Table 36: Telegram structure: sMN LMCstandby 33
Table 37: Example: sMN LMCstandby 33
Table 38: Telegram structure: sAN LMCstandby 33
Table 39: Example: sAN LMCstandby 33
Table 40: Telegram structure: sMN LMCstartmeas 34
Table 41: Example: sMN LMCstartmeas 34
Table 42: Telegram structure: sAN LMCstartmeas 35
Table 43: Example: sAN LMCstartmeas 35
Table 44: Telegram structure: sMN LMCstopmeas 35
Table 45: Example: sMN LMCstopmeas 36
Table 46: Telegram structure: sAN LMCstopmeas 36
Table 47: Example: sAN LMCstopmeas 36
Table 48: Telegram structure: sWN LMPautostartmeas 37
Table 49: Example: sWN LMPautostartmeas 1 37
Table 50: Telegram structure: sWA LMDautostartmeas 38
Table 51: Example: sWA LMPautostartmeas 38
Table 52: Telegram structure: sWN IOlasc 39
Table 53: Example: sWN IOlasc with Software Trigger and 0.5 s delay on start 39
Table 54: Telegram structure: sWA IOlasc 39
Table 55: Example: sWA IOlasc 39
Table 56: Telegram structure: sWN CLApplication 40
Table 57: Example: Activate the field application: sWN CLApplication 11 40
Table 58: Telegram structure: sWA CLApplication 40
Table 59: Example: sWA CLApplication correct and accepted 41
Table 60: Telegram structure: sWN SetActiveApplications 41
Table 61: Example: Activate the field application: : sWN SetActiveApplications 1 FEVL 1 41
Table 62: Telegram structure: sWA SetActiveApplications 42
Table 63: Example: sWA SetActiveApplications correct and accepted 42
Table 64: Telegram structure: sRN SetActiveApplications 42
Table 65: Example for MRS1000: sRN SetActiveApplications 42
Table 66: Telegram structure: sMN mSCloadfacdef 43
Table 67: Example: sMN mSCloadfacdef 43
Table 68: Telegram structure: sAN mSCloadfacdef 44

Table 69: Example: sAN mSCloadfacdef 44
Table 70: Telegram structure: sMN mSCloadappdef 45
Table 71: Example: sMN mSCloadappdef 45
Table 72: Telegram structure: sAN mSCloadappdef 45
Table 73: Example: sAN mSCloadappdef 46
Table 74: Telegram structure: sMN SetPassword 46
Table 75: Example: sMN SetPassword 46
Table 76: Telegram structure: sAN SetPassword 47
Table 77: Example: sAN SetPassword 47
Table 78: Telegram structure: sMN CheckPassword 48
Table 79: Example: sMN CheckPassword 48
Table 80: Telegram structure: sAN CheckPassword 48
Table 81: Example: sAN CheckPassword 49
Table 82: Telegram structure: sMN mSCreboot 49
Table 83: Example: sMN mSCreboot 49
Table 84: Telegram structure: sAN mSCreboot 50
Table 85: Example: sAN mSCreboot 50
Table 86: Telegram structure: sWN LCMcfg 51
Table 87: Example: sWN LCMcfg 51
Table 88: Telegram structure: sWA LCMcfg 52
Table 89: Example: sWA LCMcfg 52
Table 90: Telegram structure: sRN LCMcfg 52
Table 91: Example: sRN LCMcfg 52
Table 92: Telegram structure: sRA LCMcfg 53
Table 93: Example: sRA LCMcfg 53
Table 94: Telegram structure: sRN CMContLvlM 53
Table 95: Example: sRN CMContLvlM 54
Table 96: Telegram structure: sRA CMContLvlM 54
Table 97: Example for LMS5xx: sRA CMContLvlM 54
Table 98: Telegram structure: sMN mEEwriteall 55
Table 99: Example: sMN mEEwriteall 55
Table 100: Telegram structure: sAN mEEwriteall 56
Table 101: Example: sAN mEEwriteall 56
Table 102: Telegram structure: sMN Run 57
Table 103: Example: sMN Run 57
Table 104: Telegram structure: sAN Run 57
Table 105: Example: sAN Run 58
Table 106: Telegram structure: sWN LMDscandatacfg 60
Table 107: Example 1: sWN LMDscandatacfg 60

Table 108: Example 2: sWN LMDscandatacfg 61
Table 109: Example3: sWN LMDscandatacfg 61
Table 110: Telegram structure: sWA LMDscandatacfg 61
Table 111: Example: sWA LMDscandatacfg 61
Table 112: Telegram structure: sWN LMPoutputRange 63
Table 113: Example: sWN LMPoutputRange 0,50° resolution, 0°-90° 63
Table 114: Telegram structure: sWA LMPoutputRange 63
Table 115: Example: sWA LMPoutputRange 64
Table 116: Telegram structure: sRN LMPoutputRange 64
Table 117: Example: sRN LMPoutputRange 64
Table 118: Telegram structure: sRA LMPoutputRange 66
Table 119: Example: sRA LMPoutputRange 66
Table 120: Telegram structure: sRN LMDscandata 67
Table 121: Example: sRN LMDscandata 67
Table 122: Telegram structure: sRA LMDscandata 68
Table 123: Example: sRA LMDscandata 68
Table 124: Telegram structure: sEN LMDscandata 69
Table 125: Example: sEN LMDscandata 69
Table 126: Telegram structure: sEA LMDscandata 70
Table 127: Example: Confirmation of sEA LMDscandata 70
Table 128: Telegram structure: Datastream of sRA LMDscandata/sSN LMDscandata 82
Table 129: Example of one telegram stream 87
Table 130: Telegram structure: sMN LSPsetdatetime 88
Table 131: Example 1: sMN LSPsetdatetime 89
Table 132: Example 2: sMN LSPsetdatetime 89
Table 133: Telegram structure: sAN LSPsetdatetime 89
Table 134: Example 1, 2: sAN LSPsetdatetime 89
Table 135: Telegram structure: sRN STlms 90
Table 136: Example: sRN STlms 90
Table 137: Telegram structure: sRA STlms 91
Table 138: Example: sRA STlms 91
Table 139: Telegram structure: sRN DeviceTime 92
Table 140: Example: sRN DeviceTime 92
Table 141: Telegram structure: sRA DeviceTime 92
Table 142: Example: sRA DeviceTime 0 93
Table 143: Telegram structure: sWN TSCRole 93
Table 144: Example: sWN TSCRole 93
Table 145: Telegram structure: sWA TSCRole 94

Table 146: Example: sWA TSCRole 94
Table 147: Telegram structure: sWN TSCTCInterface 94
Table 148: Example: sWN TSCTCInterface 94
Table 149: Telegram structure: sWA TSCTCInterface 95
Table 150: Example: sWA TSCTCInterface 95
Table 151: Telegram structure: sWN TSCTCSrvAddr 95
Table 152: Example: sWN TSCTCSrvAddr 192.168.0.11 96
Table 153: Telegram structure: sWA TSCTCSrvAddr 96
Table 154: Example: sWA TSCTCSrvAddr 96
Table 155: Telegram structure: sWN TSCTCtimezone 97
Table 156: Example: sWN TSCTCtimezone GMT + 1 hour 97
Table 157: Telegram structure: sWA TSCTCtimezone 97
Table 158: Example: sWA TSCTCtimezone 98
Table 159: Telegram structure: sWN TSCTCupdatetime 98
Table 160: Example: sWN TSCTCupdatetime 600 s 98
Table 161: Telegram structure: sWA TSCTCupdatetime 99
Table 162: Example: sWA TSCTCupdatetime 99
Table 163: Telegram structure: sRN TSCTCmaxoffset 99
Table 164: Example: sRN TSCTCmaxoffset 99
Table 165: Telegram structure: sRA TSCTCmaxoffset 100
Table 166: Example: sRA TSCTCmaxoffset 18000 s 100
Table 167: Telegram structure: sRN TSCTCdelay 100
Table 168: Example: sRN TSCTCdelay 101
Table 169: Telegram structure: sRA TSCTCdelay 101
Table 170: Example: sRA TSCTCdelay 0.003 s 101
Table 171: Telegram structure: sMN mResetMaxOff 102
Table 172: Example: sMN mResetMaxOff 102
Table 173: Telegram structure: sAN mResetMaxOff 102
Table 174: Example: sAN mResetMaxOff 102
Table 175: Telegram structure: sWN LFPparticle 103
Table 176: Example: sWN LFPparticle 103
Table 177: Telegram structure: sWA LFPparticle 103
Table 178: Example: sWA LFPparticle 104
Table 179: Telegram structure: sWN LFPmeanfilter 104
Table 180: Example: sWN LFPmeanfilter 104
Table 181: Telegram structure: sWA LFPmeanfilter 105
Table 182: Example: sWA LFPmeanfilter 105
Table 183: Telegram structure: sWN LFPnto1filter 105
Table 184: Example: sWN LFPnto1filter 106

Table 185: Telegram structure: sWA LFPnto1filter 106
Table 186: Example: sWA LFPnto1filter 106
Table 187: Telegram structure: sWN FREchoFilter 107
Table 188: Example: sWN FREchoFilter 107
Table 189: Telegram structure: sWA FREchoFilter 107
Table 190: Example: sWa FREchoFilter 107
Table 191: Telegram structure: sWN MSsuppmode 108
Table 192: Example: sWN MSsuppmode 108
Table 193: Telegram structure: sWA MSsuppmode 108
Table 194: Example: sWA MSsuppmode 109
Table 195: Telegram structure: sWN CLFogFilterEn 109
Table 196: Example: sWN CLFogFilterEn 109
Table 197: Telegram structure: sWA CLFogFilterEn 110
Table 198: Example: sWA CLFogFilterEn 110
Table 199: Telegram structure: sRN CLFogFilterEn 110
Table 200: Example: sRN CLFogFilterEn 110
Table 201: Telegram structure: sRA CLFogFilterEn 111
Table 202: Example: sRA CLFogFilterEn 111
Table 203: Telegram structure: sWN MCSenseLevel 111
Table 204: Example: sWN MCSenseLevel 112
Table 205: Telegram structure: sWA MCSenseLevel 112
Table 206: Example: sWA MCSenseLevel 112
Table 207: Telegram structure: sWN CLNFDigFilterEn 113
Table 208: Example: sWN CLNFDigFilterEn 113
Table 209: Telegram structure: sWA CLNFDigFilterEn 113
Table 210: Example: sWA CLNFDigFilterEn 114
Table 211: Telegram structure: sWN CLHWFilterSectEn 114
Table 212: Example: sWN CLHWFilterSectEn 1 0 0 0 114
Table 213: Telegram structure: sWA CLHWFilterSectEn 115
Table 214: Example: sWA CLHWFilterSectEn 115
Table 215: Telegram structure: sWN LFPmedianfilter 115
Table 216: Example: sWN LFPmedianfilter 1 3 116
Table 217: Telegram structure: sWA LFPmedianfilter 116
Table 218: Example: sWA LFPmedianfilter 116
Table 219: Telegram structure: sWN LFPedgefilter 117
Table 220: Example: sWN LFPedgefilter 1 117
Table 221: Telegram structure: sWA LFPmedianfilter 117
Table 222: Example: sWA LFPedgefilter 117
Table 223: Telegram structure: sWN LICsrc 118

Table 224: Example: sWN LICsrc 118
Table 225: Telegram structure: sWA LICsrc 118
Table 226: Example: sWA LICsrc 119
Table 227: Telegram structure: sWN LICencset 119
Table 228: Example: sWN LICencset 119
Table 229: Telegram structure: sWA LICencset 120
Table 230: Example: sWA LICencset 120
Table 231: Telegram structure: sWN LICencres 120
Table 232: Example: sWN LICencres 121
Table 233: Telegram structure: sWA LICencres 121
Table 234: Example: sWA LICencres 121
Table 235: Telegram structure: sWN LICFixVel 122
Table 236: Example: sWN LICFixVel 122
Table 237: Telegram structure: sWA LICFixVel 122
Table 238: Example: sWA LICFixVel 123
Table 239: Telegram structure: sRN LICSpTh 123
Table 240: Example: sRN LICSpTh 123
Table 241: Telegram structure: sRA LICSpTh 123
Table 242: Example: sRA LICSpTh 124
Table 243: Telegram structure: sRN LICencsp 124
Table 244: Example: sRN LICencsp 124
Table 245: Telegram structure: sRA LICencsp 125
Table 246: Example: sRA LICencsp 125
Table 247: Telegram structure: sRN PortConfiguration 126
Table 248: Example: sRN PortConfiguration 126
Table 249: Telegram structure: sRN LIDoutputstate 128
Table 250: Example: sRN LIDoutputstate 128
Table 251: Telegram structure: sRA LIDoutputstate 128
Table 252: Telegram structure: sEN LIDoutputstate 129
Table 253: Example: sEN LIDoutputstate 129
Table 254: Telegram structure: sRA/sSN LIDoutputstate 132
Table 255: Example: sRA LIDoutputstate 132
Table 256: Telegram structure: sMN mDOSetOutput 133
Table 257: Example: sMN mDOSetOutput 133
Table 258: Telegram structure: sAN mDOSetOutput 133
Table 259: Example: sAN mDOSetOutput 134
Table 260: Telegram structure PRO: sWN DO6Fnc/Lite: sWN DO3Fnc 134
Table 261: Example: sWN DO6Fnc ➜ Out6 to master sync 134
Table 262: Telegram structure: PRO: sWN DO6Fnc/Lite: sWN DO3Fnc 135

Table 263: Example: sWA DO6Fnc 135
Table 264: Telegram structure: sWN DO1Fnc 135
Table 265: Example: sWN DO1Fnc ➜ Out1 to device ready 136
Table 266: Telegram structure: sWA DO1Fnc 136
Table 267: Example: sWA DO1Fnc 136
Table 268: Telegram structure: sWN DO1Logic 137
Table 269: Example: sWN DO1Logic ➜ Active_Low 137
Table 270: Telegram structure: sWA DO1Logic 137
Table 271: Example: sWA DO1Logic 137
Table 272: Telegram structure: sWN DO2Fnc 138
Table 273: Example: sWN DO2Fnc ➜ Out2 to device ready 138
Table 274: Telegram structure: sWA DO2Fnc 138
Table 275: Example: sWA DO2Fnc 138
Table 276: Telegram structure: sWN DO2Logic 139
Table 277: Example: sWN DO2Logic ➜ Active_High 139
Table 278: Telegram structure: sWA DO2Logic 139
Table 279: Example: sWA DO2Logic 139
Table 280: Telegram structure: sWN SYMode 140
Table 281: Example: sWN SYMode 140
Table 282: Telegram structure: sWA SYMode 140
Table 283: Example: sWA SYMode 140
Table 284: Telegram structure: sWN SYPhase 141
Table 285: Example: sWN SYPhase +90 141
Table 286: Telegram structure: sWA SYPhase 141
Table 287: Example: sWA SYPhase 141
Table 288: Telegram structure: sWN DO3And4Fnc 142
Table 289: Example: sWN In4 ➜ In3+4 to slave sync 142
Table 290: Telegram structure: sWA DO3And4Fnc 142
Table 291: Example: sWA DO3And4Fnc 143
Table 292: Telegram structure: sWN DI3DebTim 143
Table 293: Example: sWN DI3DebTim 143
Table 294: Telegram structure: sWA DI3DebTim 144
Table 295: Example: sWA DI3DebTim 144
Table 296: Telegram structure: sRN SYextmon 144
Table 297: Example: sRN SYextmon 144
Table 298: Telegram structure: sRA SYextmon 145
Table 299: Example: sRA SYextmon 145
Table 300: Telegram structure: sRN LCMstate 146
Table 301: Example: sRN LCMstate 146

Table 302: Telegram structure: sRA LCMstate ... 146
Table 303: Example for LMS100: sRA LCMstate ... 147
Table 304: Telegram structure: sRN DeviceIdent ... 147
Table 305: Example: sRN DeviceIdent ... 147
Table 306: Telegram structure: sRA DeviceIdent ... 148
Table 307: Example: sRA DeviceIdent ... 148
Table 308: Telegram structure: sRN SCdevicestate ... 149
Table 309: Example: sRN SCdevicestate ... 149
Table 310: Telegram structure: sRA SCdevicestate ... 150
Table 311: Example: sRA SCdevicestate ... 150
Table 312: Telegram structure: sRN LMCmeasstate ... 152
Table 313: Example: sRN LMCmeasstate ... 152
Table 314: Telegram structure: sRA LMCmeasstate ... 152
Table 315: Example: sRA LMCmeasstate is Measure 2D ... 153
Table 316: Telegram structure: sRN EMCustomerInfo ... 153
Table 317: Example: sRN EMCustomerInfo ... 153
Table 318: Telegram structure: sRA EMCustomerInfo ... 154
Table 319: Example: sRA EMCustomerInfo = Device OK ... 154
Table 320: Telegram structure: sRN LDMSenStat ... 155
Table 321: Example: sRN LDMSenStat ... 155
Table 322: Telegram structure: sRA LDMSenStat ... 156
Table 323: Example: sRA LDMSenStat Device is in Idle mode ... 156
Table 324: Telegram structure: sRN DIornr ... 157
Table 325: Example: sRN DIornr ... 157
Table 326: Telegram structure: sRA DIornr ... 157
Table 327: Example for LMS511-20100: sRA DIornr ... 158
Table 328: Example for TiM561-2050101: sRA DIornr ... 158
Table 329: Telegram structure: sRN DItype ... 158
Table 330: Example: sRN DItype ... 158
Table 331: Telegram structure: sRA DItype ... 159
Table 332: Example for LMS511-20100: sRA DItype ... 159
Table 333: Example for TiM561-2050101: sRA DItype ... 159
Table 334: Telegram structure: sRN ODoprh ... 160
Table 335: Example: sRN ODoprh ... 160
Table 336: Telegram structure: sRA ODoprh ... 160
Table 337: Example: sRA ODoprh ... 161
Table 338: Telegram structure: sRN ODpwrc ... 161
Table 339: Example: sRN ODpwrc ... 161
Table 340: Telegram structure: sRA ODpwrc ... 162

Table 341: Example: sRA ODpwrc ... 162
Table 342: Telegram structure: sRN OPcurtmpdev ... 162
Table 343: Example: sRN OPcurtmpdev ... 163
Table 344: Telegram structure: sRA OPcurtmpdev ... 163
Table 345: Example: sRA OPcurtmpdev ... 163
Table 346: Telegram structure: sWN LocationName ... 164
Table 347: Example: sWN LocationName +13 OutdoorDevice ... 164
Table 348: Telegram structure: sWA LocationName ... 165
Table 349: Example: sWA LocationName ... 165
Table 350: Telegram structure: sRN LocationName ... 165
Table 351: Example: sRN LocationName ... 166
Table 352: Telegram structure: sRA LocationName ... 166
Table 353: Example: sRA LocationName ... 166
Table 354: Telegram structure: sRN MCAngleCompSin ... 167
Table 355: Example: sRN MCAngleCompSin ... 167
Table 356: Telegram structure: sRA MCAngleCompSin ... 167
Table 357: Example: sRA MCAngleCompSin ... 168
Table 358: Telegram structure: sMN LIDrstoutpcnt ... 168
Table 359: Example: sMN LIDrstoutpcnt ... 169
Table 360: Telegram structure: sAN LIDrstoutpcnt ... 169
Table 361: Example: sAN LIDrstoutpcnt ... 169
Table 362: Telegram structure: sRN OPheatstateext ... 170
Table 363: Example: sRN OPheatstateext ... 170
Table 364: Telegram structure: sRA OPheatstateext ... 171
Table 365: Example: sRA OPheatstateext 2 ... 171
Table 366: Telegram structure: sWN EIIpAddr ... 172
Table 367: Example: sWN EIIpAddr 192.168.0.2 ... 172
Table 368: Telegram structure: sWA EIIpAddr ... 173
Table 369: Example: sWA EIIpAddr ... 173
Table 370: Telegram structure: sRN EIIpAddr ... 174
Table 371: Example: srN EIIpAddr ... 174
Table 372: Telegram structure: sRA EIIpAddr ... 174
Table 373: Example: sRA EIIpAddr 192.168.0.2 ... 175
Table 374: Telegram structure: sWN EIgate ... 175
Table 375: Example: sWN EIgate 192.168.0.1 ... 175
Table 376: Telegram structure: sWA EIgate ... 176
Table 377: Example: sWA EIgate ... 176
Table 378: Telegram structure: sRN EIgate ... 176
Table 379: Example: sRN EIgate ... 177

Table 380: Telegram structure: sRA Elgate ... 177
Table 381: Example: sRA Elgate 192.168.0.1 ... 177
Table 382: Telegram structure: sWN Elmask ... 178
Table 383: Example: sWN Elmask 255.255.254.0 ... 178
Table 384: Telegram structure: sWA Elmask ... 178
Table 385: Example: sWA Elmask ... 179
Table 386: Telegram structure: sRN Elmask ... 179
Table 387: Example: sRN Elmask ... 179
Table 388: Telegram structure: sRA Elmask ... 180
Table 389: Example: sRA Elmask 255.255.254.0 ... 180
Table 390: Telegram structure: sWN SIHstBaud ... 181
Table 391: Example: sWN SIHstBaud ... 181
Table 392: Telegram structure: sWA SIHstBaud ... 181
Table 393: Example: sWA SIHstBaud ... 182
Table 394: Telegram structure: sRN SIHstBaud ... 182
Table 395: Example: sRN SIHstBaud ... 182
Table 396: Telegram structure: sRA SIHstBaud ... 183
Table 397: Example: sRA SIHstBaud ... 183
Table 398: Telegram structure: sWN SIHstHw ... 184
Table 399: Example: sWN SIHstHw ... 184
Table 400: Telegram structure: sWA SIHstHw ... 184
Table 401: Example: sWA SIHstHw ... 185
Table 402: Telegram structure: sRN SIHstHw ... 185
Table 403: Example: sRN SIHstHw ... 185
Table 404: Telegram structure: sRA SIHstHw ... 186
Table 405: Example: sRA SIHstHw ... 186
Table 406: Telegram structure: sWN LMLfpFcn ... 187
Table 407: Example: sWN LMLfpFcn ... 187
Table 408: Telegram structure: sWA LMLfpFcn ... 188
Table 409: Example: sWA LMLfpFcn ... 188
Table 410: Telegram structure: sMN mLMLSetLed ... 188
Table 411: Example: sMN mLMLSetLed 1 1 (Stop LED) ... 189
Table 412: Telegram structure: sAN mLMLSetLed ... 189
Table 413: Example: sAN mLMLSetLed ... 189
Table 414: Telegram structure: sWN HMIfpFcn_Y1 ... 190
Table 415: Example: sWN HMIfpFcn_Y1 = Command ... 190
Table 416: Telegram structure: sWA HMIfpFcn_Y1 ... 190
Table 417: Example: sWA HMIfpFcn_Y1 ... 191
Table 418: Telegram structure: sWN HMIfpFcn_Y2 ... 191

Table 419: Example: sWN HMIfpFcn_Y2 = Command 191
Table 420: Telegram structure: sWA HMIfpFcn_Y2 192
Table 421: Example: sWA HMIfpFcn_Y2 192
Table 422: Telegram structure: sMN mHMISetLed 192
Table 423: Example: sMN mHMISetLed 1 = On 193
Table 424: Telegram structure: sAN mHMISetLed 193
Table 425: Example: sAN mHMISetLed 01 193
Table 426: Telegram structure: sMN mLMLSetDisp 195
Table 427: Example: sMN mLMLSetDisp 07 (Showing the number “7” on the display) 195
Table 428: Telegram structure: sAN mLMLSetDisp 196
Table 429: Example: sAN mLMLSetDisp 1 196
Table 430: Telegram structure: sMN EIHstCola 197
Table 431: Example: sWN EIHstCola 0 197
Table 432: Telegram structure: sWA EIHstCola 198
Table 433: Example: sWA EIHstCola 198
Table 434: Telegram structure: sEN ECRChangeArr 199
Table 435: Example: sEN ECRChangeArr 1 199
Table 436: Telegram structure: sEA ECRChangeArr 200
Table 437: Example: sEA ECRChangeArr 1 200
Table 438: Telegram structure: sSN ECRChangeArr 201
Table 439: Example: sSN ECRChangeArr 1 202
Table 440: Telegram structure: sMN mLFEgetField 203
Table 441: Example: sMN mLFEgetField 1 204
Table 442: Example: sAN mLFEgetField 207
Table 443: Example: sAN mLFEgetField 207
Table 444: Telegram structure: sRN LFEperpdistresult 209
Table 445: Example: sRN LFEperpdistresult 209
Table 446: Telegram structure: sRA LFEperpdistresult 213
Table 447: Example: sRA LFEperpdistresult 213
Table 448: Telegram structure: sEN LFEperpdistresult 215
Table 449: Example: sEN LFEperpdistresult 1 215
Table 450: Telegram structure: sEA LFEperpdistresult 216
Table 451: Example: sEA LFEperpdistresult 1 216
Table 452: Telegram structure: sSN LFEperpdistresult 219
Table 453: Example: sSN LFEperpdistresult 219
Table 454: Telegram structure: sRN LFEinfringementinfo 220
Table 455: Example: sEN LFEinfringementinfo 220
Table 456: Example: sRA LFEinfringementinfo 222

Table 457: Example: sRA LFEinfringementinfo 222
Table 458: Telegram structure: sEN LFEinfringementinfo 223
Table 459: Example: sEN LFEinfringementinfo 1 223
Table 460: Telegram structure: sEA LFEinfringementinfo 224
Table 461: Example: sEA LFEinfringementinfo 1 224
Table 462: Example: sSN LFEinfringementinfo 226
Table 463: Example: sSN LFEinfringementinfo 227
Table 464: SOPAS error codes 229
Table 465: Example: sFA ErrorCode Wrong userlevel 229

8014631/2019-06-12 · A4 4c int49

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) 30 229 25 44
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

Russia
Phone +7 495 283 09 90
E-Mail info@sick.ru

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
Sensor Intelligence.