

Telegrams for Configuring and
Operating the LMS1xx, LMS5xx,
TiM3xx, JEF300, JEF500



Contents

1	DESCRIPTION	4
2	COMMUNICATION FORMAT	4
2.1	Binary Telegram (only LMS1xx)	4
2.2	ASCII Telegram	5
2.3	Variable Types.....	6
2.4	Command Basics	6
2.5	Blanks	7
2.6	Login	7
3	WORKFLOWS.....	7
3.1	Parameterize the scan.....	7
3.2	Set Timestamp/Data Angle.....	7
4	LOG IN TO DEVICE	8
5	BASIC SETTINGS	10
5.1	Set frequency and angular resolution.....	10
5.2	Get frequency and angular resolution	13
5.3	Get the status of the LMS.....	14
6	MEASUREMENT OUTPUT TELEGRAM	16
6.1	Configure the data content for the scan	16
6.2	Configure measurement angle of the scan data for output	19
6.2.1	Ask for actual output range.....	20
6.3	Polling one Telegram.....	22
6.4	Send data permanent.....	23
6.4.1	Example and Interpretation of one Telegram	30
7	TIMESTAMP	34
7.1	Set timestamp.....	34
7.2	Ask timestamp and device status	36
8	SAVE PARAMETERS PERMANENT.....	38
9	SET TO RUN	39
10	FILTER	40
10.1	Particle Filter.....	40
10.2	Mean Filter	41
10.3	Set n-Pulse to 1-Pulse Filter	42
10.4	Echo Filter.....	43
10.5	Fog Filter.....	44

11 ENCODER.....	45
11.1 Increment source.....	45
11.2 Encoder Settings.....	46
11.3 Encoder resolution.....	47
11.4 Fixed speed.....	48
11.5 Ask speed threshold.....	49
11.6 Encoder speed.....	50
12 OUTPUTS.....	51
12.1 Ask state of the outputs.....	51
12.2 Set output state.....	53
12.3 Change output 6/3 function.....	54
12.4 Change Input 4 function.....	55
12.5 Reset output counter.....	56
13 OTHER COMMANDS.....	57
13.1 Device Ident.....	57
13.2 Device State.....	58
13.3 Device Name.....	59
13.3.1 Set Device Name.....	59
13.3.2 Ask Device Name.....	61
13.4 Operating hours.....	62
13.5 Power On Counter.....	63
13.6 IP-Address.....	64
13.7 Set factory defaults.....	65
13.8 Reboot Device.....	66
13.9 Contamination Measurement.....	67
13.9.1 Set Contamination values.....	67
13.9.2 Ask for contamination settings.....	68
13.10 Synchronization Phase.....	69
13.11 Function Front Panel.....	70
14 STANDBY MODE.....	72
15 START MEASUREMENT.....	73
16 STOP MEASUREMENT.....	74
17 SOPAS ERROR CODES.....	75
18 PROBLEMS.....	76

1 Description

That document shows how to send telegrams via terminal program in ASCII (also in Hex) or Binary to the LMS1xx. It includes the descriptions for the commands how they work and some examples commands in ASCII, HEX and Binary for sending a telegram. Also the answers to expect from the LMS after sending a command are shown. The sensor always answers in the language he was talked to.

Two workflows for getting a data scan out of the device and setting the timestamp can be found here.

This document is for LMS1xx as well as for LMS5xx and all parameters of the commands are listed but it is not a description of the differences of a LMS5xx LITE or PRO.

Also it is valid for the TiM and the JEF, please find the information which telegram is valid for which device directly in the header of each telegram.

That Guide doesn't show the differences of all the parameters and options between the devices completely.

ATTENTION: Some commands may change during SICK development process. Please use always the least version of the developer's guide.

2 Communication format

2.1 Binary Telegram (only LMS1xx)

The binary protocol is the basic protocol of the scanner. It has always a fix length and the content and byte length of the string fit to that document.

The binary protocol has a special framing so that the scanner is able to recognize it as the start of a binary telegram.

The string has to start with 4 STX symbols (for example: 02 02 02 02), that is followed by the length of the telegram in HEX (for example: 00 00 00 17).

Example:

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
	Header: 02 02 02 02; Length: 00 00 00 17; Checksum: B3

The length could be created by counting every single letter (Hex value) of the command (without checksum and framing but with blanks) and convert the value into HEX.

After the length the command itself starts. All letters of the command converted to HEX and that the parameters (mostly numbers) written directly behind the command in pairs of two. All parameters of the command have to be in hex (for example: scan frequency 25Hz is 00009C4h (It is a 4 byte value).

Checksum is calculated with XOR.

Between the command and the parameters, there has to be a blank, but not between the parameters itself.

Example string:

```
sMN SetAccessMode 04 81BE23AA
```

Binary string:

```
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 04 81 BE 23 AA 87
```

In the scan data telegram from the scanner, the range values could be used as they are, they don't have to be converted. Every value is 2 byte long.

The binary protocol could only be used at the host port of the scanner, and at the moment only with the LMS1xx.

2.2 ASCII Telegram

The ASCII telegram is an additional format and because of the ASCII signs it's a little better to understand.

The framing of the telegram is a STX at the start and an ETX at the end of each telegram.

The command is written in ASCII letters, followed by the parameters like defined in that document. Parameters could be transferred in hex or decimal format, but in decimal format they need a sign (for example: scan frequency 25Hz: 09C4h/+2500d)

Attention: leading zeros of each parameter and value were deleted, so the byte length of a parameter may not fit to what is standing in that document. That also causes different string length in the scan data telegram.

For using with PLC's the binary protocol is recommended.

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		No
Enum_16	2		No
Float_32	4	$-10^{-44.85}...+10^{38.53}$	Yes
String	Context-dependent	Strings are not terminated in zeroes	

Data length is given always 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 by name	sRN		73 52 4E
Write by name	sWN		73 57 4E
Method	sMN		73 4D 4E
Event	sEN		73 45 4E
Space	{SPC}	20	20

If there are values coming in two parts (for example the outputs in the measurement telegram documented as: 00 07, output will be 07 00; LSB first, than MSB)

2.5 Blanks

The position of the blanks in a string is different in ASCII and Binary format, so they are not listed in the tables, but they can be found in the example strings.

2.6 Login

You must be logged in before you are allowed to send any parameterization commands. Request for a data telegram can be done without login.

3 Workflows

3.1 Parameterize the scan



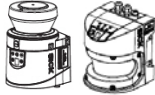
- | | |
|----------------------------------|-----------------------------------|
| 1. Log in: | sMN SetAccessMode |
| 2. Set Frequency and Resolution: | sMN mLMPsetscancfg |
| 3. Configure scan data content: | sWN LMDscandatacfg |
| 4. Configure scan data output: | sWN LMPoutputRange |
| 5. Store Parameters: | sMN mEEwriteall |
| 6. Log out: | sMN Run |
| 7. Request Scan: | sRN LMDscandata / sEN LMDscandata |

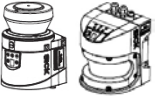


Get the exact description of that commands down in that document.

3.2 Set Timestamp/Data Angle

- | | |
|------------------|--------------------|
| 1. Log in: | sMN SetAccessMode |
| 2. Sopas command | sMN LSPsetdatetime |
| 3. Log out: | sMN Run |

4 Log in to device

   <p>PC LMS 1xx/5xx</p>					
Telegram structure: sMN SetAccessMode					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sMN	73 4D 4E
Command	User level	String	13	SetAccessMode	53 65 74 41 63 63 65 73 73 4D 6F 64 65
User level	select user level	Int_8	1	02 maintenance 03 authorized client 04 Service	02 maintenance 03 authorized client 04 Service
Password: main	"Hash" - value for the User level "Maintenance"	Uint_32	4	B21ACE26	B2 1A CE 26
Password: client	"Hash" - value for the User level "Authorized Client"	Uint_32	4	F4724744	F4 72 47 44
Password: service level	"Hash" - value for the User level "Service"	Uint_32	4	81BE23AA	81 BE 23 AA
Example: sMN SetAccessMode 03 F4724744					
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				
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				

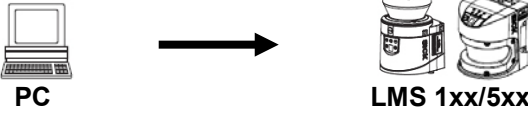
   <p>LMS 1xx/5xx PC</p>					
Telegram structure: sAN SetAccessMode					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sAN	73 41 4E
Command	User level	String	13	SetAccessMode	53 65 74 41 63 63 65 73 73 4D 6F 64 65
Change user level	changed level	Bool_1	1	0 Error 1 Success	00 Error 01 Success

Example: sAN SetAccessMode

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
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 39

5 Basic settings

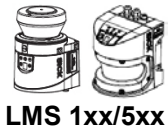
5.1 Set frequency and angular resolution

					
Telegram structure: sMN mLMPsetscancfg					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sMN	73 4D 4E
Command	Config of scan frequency and angular resolution	String	14	mLMPsetscancfg	6D 4C 4D 50 73 65 74 73 63 61 6E 63 66 67
Scan Frequency	Scan Frequency [1/100Hz]	Uint_32	4	LMS1xx: 25Hz: 9C4h (2500d) 50Hz: 1388h (5000d) LMS5xx: 25Hz: 9C4h (2500d) 35Hz: DACH (3500d) 50Hz: 1388h (5000d) 75Hz: 1A0Bh (7500d) 100Hz: 2710h (10000d)	25Hz: 00 00 09 C4 50Hz: 00 00 13 88 00 00 09 C4 00 00 0D AC 00 00 13 88 00 00 1A 0B 00 00 27 10
Value	Reserved	Int_16	2	1	00 01
Angular resolution	Angle Resolution [1/10000°]	Uint_32	4	LMS1xx: 0,25°: 9C4h (2500d) 0,5°: 1388h (5000d) LMS5xx: 0,1667°: 683h (1667d) 0,25°: 9C4h (2500d) 0,333°: D05h (3333d) 0,5°: : 1388h (5000d) 0,667°: 1A0Bh (6670d) 1°: 2710h (10000d)	0,25°: 00 00 09 C4 0,5°: 00 00 13 88 00 00 06 83 00 00 09 C4 00 00 0D 05 00 00 13 88 00 00 1A 0B 00 00 27 10
Start angle *	StartAngle [1/10000°]	Int_32	4	LMS1xx: FFF92230h..225510h -450000d..+2250000d LMS5xx: FFFF3CB0h..1C3A90h -50000d..+1850000d	FF F9 22 30 - 00 22 55 10
Stop angle *	Stop Angle [1/10000°]	Int_32	4	LMS1xx: FFF92230h..225510h -450000d..+2250000d LMS5xx: FFFF3CB0h..1C3A90h -50000d..+1850000d	FF F9 22 30 - 00 22 55 10

Example: sMN mLMPsetscancfg +5000 +1 +5000 -450000 +2250000

ASCII	<pre><STX>sMN{SPC}mLMPsetscancfg{SPC}+5000{SPC}+1{SPC}+5000{SPC}-450000{SPC}+2250000<ETX></pre> <p>alternatively:</p> <pre><STX>sMN{SPC}mLMPsetscancfg{SPC}1388{SPC}1{SPC}1388{SPC}FFF92230{SPC}225510<ETX></pre>
HEX	<pre>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</pre>
Binary	<pre>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</pre>

*** ATTENTION: Scan angle can not be changed here, only in the data output !**





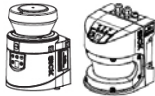
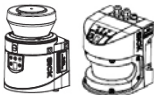


Telegram structure: sMN mLMPsetscancfg

Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sAN	
Command	Info of scan frequency and angular resolution	String	14	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	0 no Error 1 Frequency Error 2 Resolution Error 3 Res. and Scn. Error 4 Scan area Error 5 other Errors	00 01 02 03 04 05
Scan Frequency	Scan Frequency [1/100Hz]	Uint_32	4	LMS1xx: 25Hz: 9C4h (2500d) 50Hz: 1388h (5000d) LMS5xx: 25Hz: 9C4h (2500d) 35Hz: DACH (3500d) 50Hz: 1388h (5000d) 75Hz: 1A0Bh (7500d) 100Hz: 2710h (10000d)	25Hz: 00 00 09 C4 50Hz: 00 00 13 88 00 00 09 C4 00 00 0D AC 00 00 13 88 00 00 1A 0B 00 00 27 10
Value	Reserved	Int_16	2	1	00 01

Angular resolution	Angle Resolution[1/10000°]	Uint_32	4	LMS1xx: 0,25°: 9C4h (2500d) 0,5°: 1388h (5000d) LMS5xx: 0,1667°: 683h (1667d) 0,25°: 9C4h (2500d) 0,333°: D05h (3333d) 0,5°: : 1388h (5000d) 0,667°: 1A0Bh (6670d) 1°: 2710h (10000d)	0,25°: 00 00 09 C4 0,5°: 00 00 13 88 00 00 06 83 00 00 09 C4 00 00 0D 05 00 00 13 88 00 00 1A 0B 00 00 27 10
Start angle	StartAngle [1/10000°]	Int_32	4	LMS1xx: FFF92230h..225510h (-450000d..+2250000d) LMS5xx: FFFF3CB0h..1C3A90h (-50000d..+1850000d)	FF F9 22 30 - 00 22 55 10
Stop angle	Stop Angle [1/10000°]	Int_32	4	LMS1xx: FFF92230h..225510h (-450000d..+2250000d) LMS5xx: FFFF3CB0h..1C3A90h (-50000d..+1850000d)	FF F9 22 30 - 00 22 55 10
Example: sAN SetAccessMode 03 F4724744					
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				
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				

Attention: Logout from the device (sMN Run) to get the new values active !

5.2 Get frequency and angular resolution



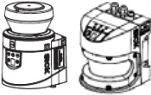
 PC		 LMS 1xx/5xx			
Telegram structure: sRN LMPscancfg					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sRN	73 52 4E
Command	Info of scan frequency and angular resolution	String	10	LMPscancfg	4C 4D 50 73 63 61 6E 63 66 67
Example: sRN LMPscancfg					
ASCII	<STX>sRN{SPC}LMPscancfg<ETX>				
HEX	02 73 52 4E 20 4C 4D 50 73 63 61 6E 63 66 67 03				
Binary	02 02 02 02 00 00 00 0E 73 52 4E 20 4C 4D 50 73 63 61 6E 63 66 67 63				
 LMS 1xx/5xx		 PC			
Telegram structure: sAN LMPscancfg					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sRA	73 52 41
Command	Info of scan frequency and angular resolution	String	10	LMPscancfg	4C 4D 50 73 63 61 6E 63 66 67
Scan Frequency	Scan Frequency [1/100Hz]	Uint_32	4	LMS1xx: 25Hz: 9C4h (2500d) 50Hz: 1388h (5000d) LMS5xx: 25Hz: 9C4h (2500d) 35Hz: DACH (3500d) 50Hz: 1388h (5000d) 75Hz: 1A0Bh (7500d) 100Hz: 2710h (10000d)	25Hz: 00 00 09 C4 50Hz: 00 00 13 88
Value	reserved	Int_16	2	1	00 01
Angular resolution	Angle Resolution [1/10000°]	Uint_32	4	LMS1xx: 0,25°: 9C4h (2500d) 0,5° 1388h (5000d)	0,25°: 00 00 09 C4 0,5°: 00 00 13 88

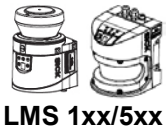
				LMS5xx: 0,1667°: 683h (1667d) 0,25°: 9C4h (2500d) 0,333°: D05h (3333d) 0,5°: : 1388h (5000d) 0,667°: 1A0Bh (6670d) 1°: 2710h (10000d)	
Start angle	StartAngle [1/10000°]	Int_32	4	LMS1xx: FFF92230h..225510h (-450000d..+2250000d) LMS5xx: FFFF3CB0h..1C3A90h (-50000d..+1850000d)	FF F9 22 30 - 00 22 55 10
Stop angle	Stop Angle [1/10000°]	Int_32	4	LMS1xx: FFF92230h..225510h (-450000d..+2250000d) LMS5xx: FFFF3CB0h..1C3A90h (-50000d..+1850000d)	FF F9 22 30 - 00 22 55 10

Example: sRA LMPscancfg

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
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

5.3 Get the status of the LMS

 PC		 LMS 1xx/5xx			
Telegram structure: sRN LCMstate					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sRN	73 52 4E
Command	Status of LMS	String	11	LCMstate	4C 43 4D 73 74 61 74 65
Example: sRN LCMstate					
ASCII	<STX>sRN{SPC}LCMstate<ETX>				
HEX	02 73 52 4E 20 4C 43 4D 73 74 61 74 65 03				
Binary	02 02 02 02 00 00 00 0C 73 52 4E 20 4C 43 4D 73 74 61 74 65 7A				



Telegram structure: sAN LCMstate

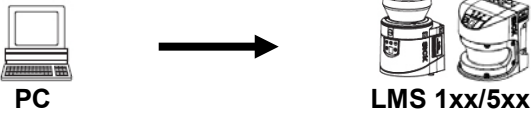
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sRA	73 52 41
Command	Status of LMS	String	8	LCMstate	4C 43 4D 73 74 61 74 65
Status Code		Enum_8	1	0 no Error 1 pollution warning 2 pollution error 3 fatal error	00 no Error 01 pollution warning 02 pollution error 03 fatal error

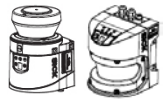
Example: sRA LCMstate

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
Binary	02 02 02 02 00 00 00 0E 73 52 41 20 4C 43 4D 73 74 61 74 65 20 00 55

6 Measurement output telegram

6.1 Configure the data content for the scan

Telegram	Description	Variable	Length	Values ASCII	Values Binary
 <p>PC → LMS 1xx/5xx</p> <p>Telegram structure: sWN LMDscandatacfg</p>					
Command Type	Sopas by name	String	3	sWN	73 57 4E
Command	Configure Scan data	String	14	LMDscandatacfg	4C 4D 44 73 63 61 6E 64 61 74 61 63 66 67
Data channel	Defines the Telegram content	Uint_8	2 x 1	LMS1xx: Output channel 1: 01 00 Output channel 2: 02 00 Output channel 1+2: 03 00 10 reserved FF reserved LMS5xx: Set via Echo Filter Set this value to 0	Output channel 1: 01 00 Output channel 2: 02 00 Output channel 1+2: 03 00 10 reserved FF reserved LMS5xx: Set via Echo Filter Set this value to 0
Remission	Remission data output	Bool_1	1	0 no 1 yes	00 no 01 yes
Resolution	Resolution of Remission Data (LMS5xxV1.10 only 8bit)	Enum_8	1	0: 8 Bit 1: 16 Bit	00: 8 Bit 01: 16 Bit
Unit	Unit of Remission Data	Enum_8	1	0 Digits	00 Digits
Encoder	Encoder Data	Uint_8	2 x 1	00 00 no Encoder 01 00 Channel 1 02 00 reserved FF 00 reserved	00 00 no Encoder 01 00 Channel 1 02 00 reserved FF 00 reserved
Position	Position Values	Bool_1	1	0 no 1 yes	00 no 01 yes
Device Name	Sends the device name	Bool_1	1	0 no 1 yes	00 no 01 yes
Comment	Saved comment	Bool_1	1	0 no 1 yes	00 no 01 yes
Time	Sends time information	Bool_1	1	0 no 1 yes	00 no 01 yes
Output rate	Sends the output rate	Uint_16	2	+1 all Scans +2 each 2.nd Scan 50000 each 50000 nd. Scan	00 01 all Scans 00 02 each 2.nd Scan 50000 each 50000 nd. Scan



LMS 1xx/5xx



PC

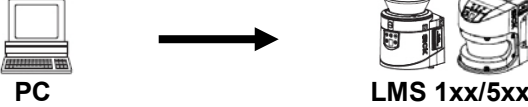
Telegram structure: sWA LMDscandatacfg

Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sWA	73 57 4E
Command	Scan data Configuration	String	14	LMDscandatacfg	4C 4D 44 73 63 61 6E 64 61 74 61 63 66 67

Example 1, 2, 3: sWA LMDscandatacfg

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
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

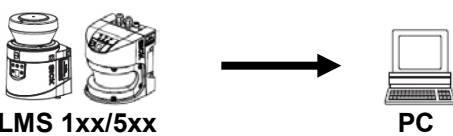
6.2 Configure measurement angle of the scan data for output

Telegram	Description	Variable	Length	Values ASCII	Values Binary
					
Telegram structure: sWN LMPoutputRange					
Command Type	Sopas by name	String	3	sWN	73 57 4E
Command	Change output angle range	String	14	LMPoutputRange	4C 4D 50 6F 75 74 70 75 74 52 61 6E 67 65
Status Code	Length	Int_16	2	1	00 01
Angle Resolution *	[1/10000°]	Uint_32	4	LMS1xx: 0,25°: 9C4h (2500d) 0,5°: 1388h (5000d) LMS5xx: 0,1667°: 683h (1667d) 0,25°: 9C4h (2500d) 0,333°: D05h (3333d) 0,5°: 1388h (5000d) 0,667°: 1A0Bh (6670d) 1°: 2710h (10000d)	0,25°: 00 00 09 C4 0,5°: 00 00 13 88
StartAngle	[1/10000°]	Int_32	4	LMS1xx: FFF92230h..225510h (-450000d..+2250000d) LMS5xx: FFFF3CB0h..1C3A90h (-50000d..+1850000d)	FF F9 22 30 ... 00 22 55 10
Stop Angle	[1/10000°]	Int_32	4	LMS1xx: FFF92230h..225510h (-450000d..+2250000d) LMS5xx: FFFF3CB0h..1C3A90h (-50000d..+1850000d)	FF F9 22 30 ... 00 22 55 10
Example: sWN LMPoutputRange 50Hz 0° - 90°					
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				
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 0D BB A0 F7				

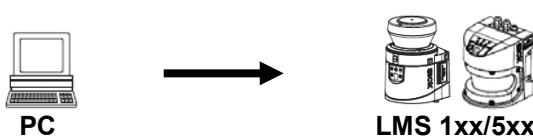
- ATTENTION: Angle resolution can not be changed here, it is taken automatically from the basic scan settings !**

The angular resolution is not exactly 0.1667 degree, and this value should not be used for calculations. What it means is that the ang. resolution is 0.166666666... or 1°/6 (six shots per degree). When used for calculations a customer should recover the real value, e.g. by `double AngRes = 2.0 / round(2.0 / GivenAngRes);`

This is how we handle it internally as well.

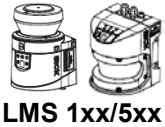
					
Telegram structure: sWA LMPoutputRange					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sWA	73 57 41
Command	Store parameters	String	14	LMPoutputRange	4C 4D 50 6F 75 74 70 75 74 52 61 6E 67 65
Example: sWA LMPoutputRange					
ASCII	<STX>sWA{SPC}LMPoutputRange<ETX>				
HEX	02 73 57 41 20 4C 4D 50 6F 75 74 70 75 74 52 61 6E 67 65 30				
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 20 74				

6.2.1 Ask for actual output range

					
Telegram structure: sRN LMPoutputRange					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sRN	73 52 4E
Command	Output range	String	14	LMPoutputRange	4C 4D 44 73 63 61 6E 64 61 74 61

Example: sRN LMPoutputRange

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
Binary	02 02 02 02 00 00 00 0F 73 52 4E 20 4C 4D 50 6F 75 74 70 75 74 52 61 6E 67 65 5E

**Telegram structure: sRA LMPoutputRange**

Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sRN	73 52 4E
Command	Output range	String	11	LMDscandata	4C 4D 44 73 63 61 6E 64 61 74 61


Example: sRA LMPoutputRange


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
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

6.3 Polling one Telegram

Output of measured values of one scan.

Sends the last valid scan data back from the memory of the LMS. Also if the measurement is not running, the last measurement is available.

					
Telegram structure: sRN LMDscandata					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sRN	73 52 4E
Command	Only one Telegram	String	11	LMDscandata	4C 4D 44 73 63 61 6E 64 61 74 61
Example: sRN LMDscandata					
ASCII	<STX>sRN{SPC}LMDscandata<ETX>				
HEX	02 73 52 4E 20 4C 4D 44 73 63 61 6E 64 61 74 61 03				
Binary	02 02 02 02 00 00 00 0F 73 52 4E 20 4C 4D 44 73 63 61 6E 64 61 74 61 05				






					
Telegram structure: sRS LMDscandata					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Find complete telegram structure of the Answer under topic "Send data permanent"					
Example: sRN LMDscandata					
ASCII	→ No ASCII Answer possible				
HEX	Find complete telegram structure of the Answer under topic "Send data permanent"				
Binary	Find complete telegram structure of the Answer under topic "Send data permanent"				






```

Received/Sent data
Connecting to 192.168.0.1 ...
Connected to 192.168.0.1
sRN LMDscandatasRA LMDscandata 1 1 89C997 0 0 1AAE
1AB1 581CBC15 581D153D 0 0 7 0 0 1388 168 0 1
DIST1 3F800000 00000000 186A0 1388 15 F6 F9 F5 EF
F6 F2 EF ED F5 E9 F2 FA FC FF F1 F2 107 FC FC 102
FF 0 0 0 0 0 0

```

6.4 Send data permanent

 PC	→	   			
Telegram structure: sEN LMDscandata					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sEN	73 45 4E
Command	Data Telegram	String	11	LMDscandata	4C 4D 44 73 63 61 6E 64 61 74 61
Measurement	Start/Stop	Enum_8	1	0 Stop 1 Start	01
Example: sEN LMDscandata					
ASCII	<STX>sEN{SPC}LMDscandata{SPC}1<ETX>				
HEX	02 73 45 4E 20 4C 4D 44 73 63 61 6E 64 61 74 61 20 31 03				
Binary	02 02 02 02 00 00 00 11 73 45 4E 20 4C 4D 44 73 63 61 6E 64 61 74 61 20 01 33				

   	→	 PC			
Telegram structure: sEA LMDscandata					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sEA	73 45 41
Command	Data Telegram	String	11	LMDscandata	4C 4D 44 73 63 61 6E 64 61 74 61
Measurement	Start/Stop	Enum_8	1	0 Stop 1 Start	01

Example: sEA LMDscandata

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
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

**The answer to the telegram will be followed by the scan data:
 (Attention: leading zeros of a value will not be displayed in ASCII)**

Telegram Stream

Telegram structure: sRA LMDscandata / sSN LMDscandata

Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sRA sSN	73 52 41 73 53 4E
Command	Data Telegram	String	11	LMDscandata	4C 4D 44 73 63 61 6E 64 61 74 61
Version Number	For detecting format changes by the version. Version is always 1 up to now	Uint_16	2	0000h - FFFFh	
Device Information	Device Number	defined with Sopas	Uint_16	2	0000h - FFFFh
	Serial Number	defined in Factory	Uint_32	4	00000000h - FFFFFFFFh
	Device Status		Uint_8	2 x 1	00 00 OK 00 01 Error 00 02 Pollution Warning 00 04 Pollution Error
Status Info	Telegram Counter	Counter starting with first measured value after reaching the highest number	Uint_16	2	0000h - FFFFh
	Scan Counter	Counter starting with first measured value after reaching the highest number	Uint_16	2	0000h - FFFFh
	Time since start up	Counting the time since power up the device; starting with 0. In the output telegram this is the time at the zero index (-14°) before the measurement itself starts.	Uint_32	4	00000000h - FFFFFFFFh
	Time of transmission	Time in µs when the complete scan is transmitted to the buffer for data output; starting with 0 at scanner boot up.	Uint_32	4	00000000h - FFFFFFFFh
	Status of digital Inputs	Low byte represents Input 1	Uint_8	2 x 1	00 00 all Inputs low 00 03 all input high
	Status of digital outputs	Low byte represents Output 1	Uint_8	2 x 1	00 00 all Outputs low 00 07 all Output high

Telegram	Description	Variable	Length	Values ASCII	Values Binary
	Reserved	Uint_16	2		
Frequencies	Scan frequency	Uint_32	4	LMS1xx: 25Hz: 9C4h (2500d) 50Hz: 1388h (5000d) LMS5xx: 25Hz: 9C4h (2500d) 35Hz: DACH (3500d) 50Hz: 1388h (5000d) 75Hz: 1A0Bh (7500d) 100Hz: 2710h (10000d)	
	Measurement frequency	Uint_32	4	00000000h - FFFFFFFFh	
Amount of Encoder		Enum_16	2	0..3 if 0, than next two values are missing.	
	Encoder Position	Uint_16	2	LMS1xx: 0000h - 3FFFh LMS5xx: 0000h - FFFFh	
	Encoder Speed	Uint_16	2	0000h - FFFFh	
Amount of 16 Bit Channels		Enum_16	2	LMS1xx: 1..2 Output channels LMS5xx: 0 or 5 Output channels	
Outputchannel 1..4 (16 bit)	Content	String	5	LMS1xx: DIST1: radial Values of first pulse in mm RSSI1:Energy Values of first pulse DIST2: radial Values of 2nd pulse in mm RSSI2:Energy Values of 2nd pulse LMS5xx: DIST1 DIST2 DIST3 DIST4 DIST5 No RSSI Values	44 49 53 54 31 52 53 53 49 31 44 49 53 54 32 52 53 53 49 32
	Scale factor	Float_32	4	3F800000h = factor x1 40000000h = factor x2 (values have to be scaled by factor two)	
	Scale factor offset	LMS = 0	Float_32	4	00000000h - FFFFFFFFh

Telegram	Description	Variable	Length	Values ASCII	Values Binary	
	Start angle	Output format : 1/10.000°	Uint_32	4	LMS1xx: -450.000 +2250.000 LMS5xx: -50.000 +1850.000	
	Steps	Output format : 1/10.000°	Uint_16	2	LMS1xx: 1000 10.000 LMS5xx: 1667..10.000	
	Amount of Data	Defines the number of items on measured output	Uint_16	2	0000h – FFFFh	
	Data_1 Data_n	Data stream starting Data_1 to Data_n	Uint_16	2	0000h - 4E20 (LMS100) C350 (LMS150) FDE8 (LMS 1xx without limit)	
Amount of 8 Bit Channels	Amount of 8 Bit channels, giving out the Measured Data	Enum_16	2	LMS1xx: 1..2 Output channels LMS5xx: 1 or 5 Output channels		
Outputchannel 1..4 (8 bit)	Content	Defines the Content of the Output channel	String	5	LMS1xx: DIST1 RSSI1 DIST2 RSSI2 LMS5xx: DIST1 DIST2 DIST3 DIST4 DIST5 No RSSI Values	44 49 53 54 31 52 53 53 49 31 44 49 53 54 32 52 53 53 49 32
	Scale factor	Scale factor or of the measurement values (in LMS5xx depends on the angular resolution)	Real	4	3F800000h = factor x1 40000000h = factor x2 (values have to be scaled by factor two)	
	Scale factor offset	LMS = 0	Real	4	00000000h - FFFFFFFFh	
	Start angle	Output format : 1/10.000°	Int_32	4	LMS1xx: -450.000 +2250.000 LMS5xx: -50.000 +1850.000	
	Steps	Output format : 1/10.000°	Uint_16	2	LMS1xx: 1000 10.000 LMS5xx: 1667..10.000	
	Amount of Data		Uint_16	2	0000h – FFFFh	
	Data_1 Data_n	Data stream starting Data_1 to Data_n	Uint_8	1	00h – FFh	

Attention: The grey written parts are not given out by the sensor.

Position	Output of Position data	Enum_16	2	0 no position Data 1 Position Data	00 00 no position Data 00 01 Position Data	
Position Information	X Position	X- Coordinate	Real	4	00000000h - FFFFFFFFh	
	Y Position	Y- Coordinate	Real	4	00000000h - FFFFFFFFh	
	Z Position	Z- Coordinate	Real	4	00000000h - FFFFFFFFh	
	X Rotation	X- Rotation in the Coordinate system	Real	4	00000000h - FFFFFFFFh	
	Y Rotation	Y - Rotation in the Coordinate system	Real	4	00000000h - FFFFFFFFh	
	Z Rotation	Z - Rotation in the Coordinate system	Real	4	00000000h - FFFFFFFFh	
	Rotations Type	kind of Rotation	Enum_8	1	0 no rotation 1 pitch 2 rollin 3 free	00 no rotation 01 pitch 02 rollin 03 free
	Transmits the Name of device	enum 8	Uint_8	1	0 no Name 1 Name	00 no Name 01 Name
Name	Device Name	Enum_16	2	0 no name 1 name	00 00 no name 00 01 name	
Length	Length of Name	Uint_8	1	0h - Fh	00 – 0F	
	Name	Device Name	String	2	0 ... 16 Chars (20h..FFh)	
Comment	Comment	Enum_16	2	0 no Comment 1 comment	00 00 no Comment 00 01 comment	
Length	Length of comment	Uint_8	1	0h - Fh	00 – 0F	
	Comment	Transmits a comment	String	2	0 ... 16 Chars (20h..FFh)	
Time	transmits a time stamp	Enum_16	2	0 no time 1 time	00 00 no time 00 01 time	
Time Info	Year		Uint_16	2	0000h ... 270Fh	
	Month	1 to 12	Uint_8	1	00h ... 0Ch	
	Day	Day of Month 1 to 31	Uint_8	1	00h ... 1Fh	
	Hour	0 to 23	Uint_8	1	00h ... 17h	
	Minute	0 to 59	Uint_8	1	00h ... 3Bh	
	Second	0 to 59	Uint_8	1	00h ... 3Bh	
	µ seconds	0 to 999.999	Unit_32	4	00000000h - 000F423Fh	

Event Info		Give out event info	Enum_16	2	0 no Info 1 transmit info	00 00 no Info 00 01 transmit info
Event Information	Type	Fast digital input	String	4	FDIN	FDIN
	Encoder Position	Position of encoder when event happened	Uint_32	4	00000000h - FFFFFFFFh	
	Time of Event	Time (µs) of encoder when event happened	Uint_32	4	00000000h - FFFFFFFFh	
	Angle of Event	Angle of encoder when event happened	Int_32	4	0...3.600.000	

Attention: The grey written parts are not given out by the sensor.

Example for data amount LMS5xx:

With ASCII protocol (Cola A) a distance value needs 5 Byte and a remission value 3 Byte.
shot rate is max. 54 kHz, for example configuration 75Hz 0.5°

- 5 + 3 Byte / Echo
- 5 Echo / Spot
- 190° / 0.5° + 1 Spot / Scan
- 75 Scan / s
- = 1.1 MB/s (without Overhead)
- = 1.2 MB/s (Brutto, incl. Header)

Means in that configuration a 10 MBit connection is not enough. With a 100MBit Hub, 3 - 4 scanners can be used, with a 1GBit Hub accordingly more.

6.4.1 Example and Interpretation of one Telegram

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





ASCII

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

All Values are separated with a 20hex {SPC}

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 C8 C8 C8 CC 15 58 86 D8 15 58 8C 5A 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 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
```

		 LMS 1xx/5xx	 TiM 3xx	 JEF 300/500	→	 PC
Telegram	Values ASCII	Values Binary	Variable	Length	Possible Values	
Frame/Header	02: STX	02 02 02 02				
Length		00 00 00 83				
Command Type	sRA{SPC}	73 52 41 20	String	3	sRA/ sSN	
Command	LMDscandata{SPC}	4C 4D 44 73 63 61 6E 64 61 74 61 20	String	11	LMDscandata	
Version Number	1{SPC}	00 01	Uint_16	2	0000h FFFFh	
Device Info	Device Number	1{SPC}	00 01	Uint_16	2	0000h FFFFh
	Serial Number	89A27F{SPC} Dec: 9020031	00 89 A2 7F	Uint_32	4	00000000h FFFFFFFFh
	Device Status	0{SPC} 0{SPC}	00 00	Uint_8	2 x 1	00 00 OK 00 01 Error 00 02 Pollution Warning 00 04 Pollution Error
Status	Telegram counter	343{SPC} Dec:835	C8 C8	Uint_16	2	0000h FFFFh
	Scan Counter	347{SPC} Dec:839	C8 CC	Uint_16	2	0000h FFFFh

	Time since start up μ sek	27477BA9{SPC} Dec: 658996137	15 58 86 D8	Uint_32	4	00000000h FFFFFFFFh
	Time of transmission μ sek	2747813B{SPC} Dec: 568997563	15 58 8C 5A	Uint_32	4	00000000h FFFFFFFFh
	Status of digital Inputs	0{SPC}0{SPC} input 1 & 2 low	00 00	Uint_8	2 x 1	00 00 all Inputs low 00 03 all input high
	Status of digital outputs	7{SPC}0{SPC} 0111 – all internal outputs high external outputs here not set!	07 00	Uint_8	2 x 1	00 00 all Outputs low 00 07 all Output high
	Reserved	0{SPC}	00 00	Uint_16	2	
Frequencies	Scan Frequency	1388{SPC} Dec: 5000 -> 50Hz	00 00 13 88	Uint_32	4	2500 25hz 50 Hz: 1388h (5000d)
	Measurement frequency	168{SPC}	00 00 01 68	Uint_32	4	00000000h FFFFFFFFh
	Amount of Encoder	0{SPC} No encoder data	00 00	Enum_1 6	2	1 ..3
Position	Encoder Position	not generated, not existing because amount is 0	not generated, not existing because amount is 0	Uint_16	2	0000h 3FFFh
	Encoder Speed	not generated, not existing because amount is 0	not generated, not existing because amount is 0	Uint_16	2	0000h FFFFh
	Amount of 16 Bit Channels	1{SPC}	00 01	Enum_1 6	2	1 ..4 Output channels
Outputchannel 1..4 (16 bit)	Content	DIST1{SPC}	44 49 53 54 31	String	5	DIST1: radial Values of first pulse RSSI1:Energy Values of first pulse DIST2: radial Values of 2nd pulse RSSI2:Energy Values of 2nd pulse
	Scale Factor	3F800000{SPC} Floating Point: <u>Value = 1</u>	3F 80 00 00	Real	4	3F800000h = factor x1 40000000h = factor x2 (values have to be scaled by factor two)
	Scale Factor offset	00000000{SPC} Floating Point: <u>Value = 0</u>	00 00 00 00	Real	4	00000000h FFFFFFFFh
	Start angle	186A0 {SPC} Dec: 100000	00 01 86 A0	Int_32	4	-450.000 +2250.000
	Steps	1388{SPC} Dec:5000	13 88	Uint_16	2	1000 10.000
	Amount of Data	15{SPC} Dec: 21 Measurement points	00 15	Uint_16	2	0000h FFFFh

	Data_1 Data_2 ...Data-n	Min 16h = 22mm Max. 4E20h = 20000mm	Min. 00 16h = 22mm Max. 4E 20h = 20000mm	Uint_16	2	0000h FFFFh
	Amount of 8 Bit Channels	0{SPC} No 8 Bit Data	00 00 No 8 Bit Data	Enum_1 6	2	1...4Outputchannels
Outputchannel 1..4 (8 bit)	Content			String	5	DIST1 RSSI1 DIST2 RSSI2
	Scale factor			Real	4	3F800000h = factor x1 40000000h = factor x2 (values have to be scaled by factor two)
	Scale factor offset			Real	4	00000000h FFFFFFFFh
	Start angle			Int_32	4	-450.000 + 2250.000
	Steps			Uint_16	2	1000 10.000
	Amount of Data			Uint_16	2	0000h FFFFh
	Data_1 Data_n			Uint_8	1	00h FFh
	Position	0{SPC} No position data	00 00 No position data	Enum_1 6	2	0 no position Data 1 Position Data
Position Information	X Position			Real	4	00000000h FFFFFFFFh
	Y Position			Real	4	00000000h FFFFFFFFh
	Z Position			Real	4	00000000h FFFFFFFFh
	X Rotation			Real	4	00000000h FFFFFFFFh
	Y Rotation			Real	4	00000000h FFFFFFFFh
	Z Rotation			Real	4	00000000h FFFFFFFFh
	Rotations Type			Enum_8	1	0 no rotation 1 pitch 2 rollin 3 free
	Transmits the Name of device			Uint_8		0 no Name 1 Name
	Name	0{SPC} No device Name	00 00 No device Name	Enum_1 6	2	0 no name 1 name
Name info	Length of name			Enum_8	1	0h- Fh
	Name			String	2	0 ... 16 Chars (20h..FFh)

Comment	0{SPC} No comment	00 00 No comment	Enum_1 6	2	0 no Comment 1 comment
Comment	Length of comment		Enum_8	1	0h- Fh
	comment		String	2	0 ... 16 Chars (20h..FFh)
Time	0{SPC} No time transmitted	00 00 No time transmitted	Enum_1 6	2	0 no time 1 time
Time Info	Year		Uint_16	2	0000h 270Fh
	Month		Uint_8	1	00h 0Ch
	Day		Uint_8	1	00h 1Fh
	Hour		Uint_8	1	00h 17h
	Minute		Uint_8	1	00h 3Bh
	Second		Uint_8	1	00h 3Bh
	µ seconds		Unit_32	4	00000000h 000F423Fh
Event Info	0{SPC} No event info available	00 00 No event info available	Enum 16	2	0 no Info 1 transmit info
Event Information	Type		String	4	FDIN
	Encoder Position		Uint_32	4	00000000h FFFFFFFFh
	Time of Event		Uint_32	4	00000000h FFFFFFFFh
	Angle of Event		Int_32	4	-450000 +2250000
Frame	ETX: 03	Checksum: 2B			

7 Timestamp

7.1 Set timestamp


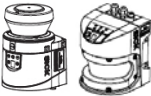
The data format in the telegram is: +2009 +7 +22 +12 +0 +0 +0

Represents: (year month day hour minute second microsecond) always with blank in between.

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

Attention: It is no real time clock inside the LMS, so if the device is turned off, the time is not running on, so after rebooting the device, time is not actual anymore and has to be set again.

Time can be saved permanent anyway, for example for analyzing the „OFF“- time of the sensor.

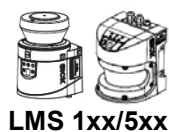
 PC	→	 LMS 1xx/5xx			
Telegram structure: sMN LSPsetdatetime					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sMN	73 4D 4E
Command	Set timestamp	String	14	LSPsetdatetime	4C 53 50 73 65 74 64 61 74 65 74 69 6D 65
Year		Uint_16	2	+0000d...+9999d 0000h... FFFFh	00 00h ...FF FFh
Month		Uint_8	1	+00d ... +99d 00h ... FFh	00h ... FFh
Day		Uint_8	1	+00d ... +99d 00h ... FFh	00h ... FFh
Hour		Uint_8	1	+00d ... +99d 00h ... FFh	00h ... FFh
Minute		Uint_8	1	+00d ... +99d 00h ... FFh	00h ... FFh
Second		Uint_8	1	+00d ... +99d 00h ... FFh	00h ... FFh
Microsecond		Uint_32	4	+00000000d ... +99999999d 00000000h ... FFFFFFFFh	00 00 00 00h... FF FF FF FFh

Example 1: sMN LSPsetdatetime

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
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

Example 2: sMN LSPsetdatetime

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
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



Telegram structure: sAN LSPsetdatetime


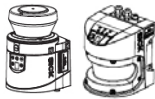
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sAN	73 41 4E
Command	Set timestamp	String	14	LSPsetdatetime	4C 53 50 73 65 74 64 61 74 65 74 69 6D 65
Status Code		Enum_8	1	1 = Success	01 = Success

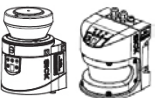

Example 1, 2: sAN LSPsetdatetime

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
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

Activate time stamp in the [output string format](#) or on Sopas page "data processing"

7.2 Ask timestamp and device status

 PC	→	 LMS 1xx/5xx			
Telegram structure: sRN STlms					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Request	String	3	sRN	73 52 4E
Command	Ask for time and status	String	5	STlms	53 54 6C 6D 73
Example: sRN STlms					
ASCII	<STX>sRN{SPC}STlms<ETX>				
HEX	02 73 52 4E 20 53 54 6C 6D 73 03				
Binary	02 02 02 02 00 00 00 09 73 52 4E 20 53 54 6C 6D 73 3A				


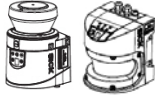
 LMS 1xx/5xx	→	 PC			
Telegram structure: sRA STlms					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sRA	73 52 41
Command	Status and time	String	5	STlms	53 54 6C 6D 73
Status Code		Enum_16	2	Status 0 = undefined 1 = initialization 2 = configuration 3 = lower case 4 = rotating 5 = in preparation 6 = ready 7 = measurement active 8 .. 11 = reserved	Status 00 00 = undefined 00 01 = initialization 00 02 = configuration 00 03 = lower case 00 04 = rotating 00 05 = in preparation 00 06 = ready 00 07 = measurement active 00 08 .. 00 11 = reserved
Op. Temp. Range		Uint_8	1		00h..FFh
...		Uint_16	2		00 00h..FF FFh
Time	HH HH	Uint_16	2	00d..99d	30 30h..39 39h
	:	Uint_8	1	:	3A

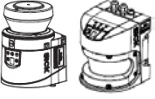

	MM MM	Uint_16	2	00d..99d	30 30h..39 39h
	:	Uint_8	1	:	3A
	SS SS	Uint_16	2	00d..99d	30 30h..39 39h
...		Uint_16	2		
Date	DD DD	Uint_16	2	00d..99d	30 30h..39 39h
	.	Uint_8	1	.	2E
	MM MM	Uint_16	2	00d..99d	30 30h..39 39h
	.	Uint_8	1	.	2E
	JJ JJ JJ JJ	Uint_32	4	0000d..9999d	30 30 30 30h..39 39 39 39h
LED1		Uint_16	2	0 = inactive 1 = active	00 00 = inactive 00 01 = active
LED2		Uint_16	2	0 = inactive 1 = active	00 00 = inactive 00 01 = active
LED3		Uint_16	2	0 = inactive 1 = active	00 00 = inactive 00 01 = active

Example: sRA STImS


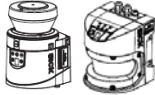
ASCII	<STX>sRA{SPC}STImS{SPC}7{SPC}0{SPC}8{SPC}16:36:54{SPC}8{SPC}17.03.2030{SPC}0{SPC}0{SPC}0<ETX>
HEX	Not available
Binary	02 02 02 02 00 00 00 2F 73 52 41 20 53 54 6C 6D 73 20 00 07 00 00 08 31 36 3A 33 36 3A 35 34 00 0A 31 37 2E 30 33 2E 32 30 33 30 00 00 00 00 00 00 00 00 00 00 00 00 17

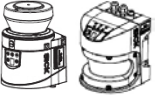

8 Save parameters permanent

 →  PC LMS 1xx/5xx					
Telegram structure: sMN mEEwriteall					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sMN	73 4D 4E
Command	store Parameters permanent	String	11	mEEwriteall	6D 45 45 77 72 69 74 65 61 6C 6C
Example: sMN mEEwriteall					
ASCII	<STX>sMN{SPC}mEEwriteall<ETX>				
HEX	02 73 4D 4E 20 6D 45 45 77 72 69 74 65 61 6C 6C 03				
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				

 →  LMS 1xx/5xx PC					
Telegram structure: sAN mEEwriteall					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sAN	73 41 4E
Command	Store parameters	String	11	mEEwriteall	6D 45 45 77 72 69 74 65 61 6C 6C
Status Code	accepted when value is 1	Enum_8	1	0 Error 1 Success	00 Error 01 Success
Example: sAN mEEwriteall					
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				
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				



9 Set to run

 PC	→	 LMS 1xx/5xx			
Telegram structure: sMN Run					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Request (SOPAS method by name)	String	3	sMN	73 4D 4E
Command	Start the device	String	3	Run	52 75 6E
Example: sMN Run					
ASCII	<STX>sMN{SPC}Run<ETX>				
HEX	02 73 4D 4E 20 52 75 6E 03				
Binary	02 02 02 02 00 00 00 07 73 4D 4E 20 52 75 6E 19				

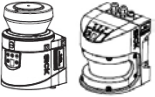

 LMS 1xx/5xx	→	 PC			
Telegram structure: sAN Run					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sAN	73 41 4E
Command	Start the device	String	3	Run	52 75 6E
Status Code	The command has been accepted if the status code 1 is returned	Bool_1	1	1 Success 0 Error	01 Success 00 Error
Example: sAN Run					
ASCII	<STX>sAN{SPC}Run{SPC}1<ETX>				
HEX	02 73 41 4E 20 52 75 6E 20 31 03				
Binary	02 02 02 02 00 00 00 09 73 41 4E 20 52 75 6E 20 01 34				

10 Filter



10.1 Particle Filter



 PC	→	 LMS 1xx/5xx			
Telegram structure: sWN LFPparticle					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sWN	73 57 4E
Command	Set particle filter	String	11	LFPparticle	4C 46 50 70 61 72 74 69 63 6C 65
Status code		Bool_1	1	0 inactive 1 active	00 inactive 01 active
Threshold *	Particle threshold in mm	Uint_16	2	+500 (must be taken)	01 F4 (must be taken)
Example: sWN LFPparticle					
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				
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				

* Never change the threshold here, it is take by the device to handle the particles



 LMS 1xx/5xx	→	 PC			
Telegram structure: sWA LFPparticle					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sWA	73 57 41
Command	Cont. values	String	6	LCMcfg	4C 43 4D 63 66 67
Example: sWA LFPparticle					
ASCII	<STX>sWA{SPC}LFPparticle<ETX>				
HEX	02 73 57 41 20 4C 46 50 70 61 72 74 69 63 6C 65 03				
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				



10.2 Mean Filter

 PC	→	 LMS 1xx	(only LMS1xx)		
Telegram structure: sWN LFPmeanfilter					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sWN	73 57 4E
Command	Set mean filter	String	13	LFPmeanfilter	4C 46 50 6D 65 61 6E 66 69 6C 74 65 72
Status code		Bool_5	1	0 inactive 1 active	00 inactive 01 active
Number of scans		Uint_16	2	+2...+100	00 02...00 64
		Enum_8	1	0	00
Example: sWN LFPmeanfilter					
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				
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				



 LMS 1xx	→	 PC			
Telegram structure: sWA LFPmeanfilter					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sWA	73 57 41
Command	Cont. values	String	6	LCMcfg	4C 43 4D 63 66 67
Example: sWA LFPmeanfilter					
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				
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 20 38				



10.3 Set n-Pulse to 1-Pulse Filter

 PC	→	 LMS 1xx	Only LMS1xx, for LMS5xx take the echo filter		
Telegram structure: sWN LFPnto1filter					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sWN	73 57 4E
Command	Set n-to-1 Filter	String	13	LFPnto1filter	4C 46 50 6E 74 6F 31 66 69 6C 74 65 72
Status code		Bool_1	1	0 inactive 1 active	00 inactive 01 active
Example: sWN LFPnto1filter					
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				
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				


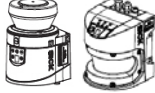
 LMS 1xx	→	 PC			
Telegram structure: sWA LFPnto1filter					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sWA	73 57 41
Command	Set n-to-1 Filter	String	13	LFPnto1filter	4C 46 50 6E 74 6F 31 66 69 6C 74 65 72
Example: sWA LFPnto1filter					
ASCII	<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				
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 20 7B				

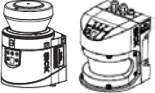

10.4 Echo Filter

 PC	→	 LMS 5xx	Only LMS5xx, LMS1xx take the n-Pulse to 1-Pulse filter		
Telegram structure: sWN FREchoFilter					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sWN	73 57 4E
Command	Set echo Filter	String	12	FREchoFilter	46 52 45 63 68 6F 46 69 6C 74 65 72
Status code		Enum_8	1	0 = First Echo 1 = All Echos 2 = Last Echo	00 = First Echo 01 = All Echos 02 = Last Echo
Example: sWN FREchoFilter					
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				
Binary	Not available in V1.10 firmware				

 LMS 5xx	→	 PC			
Telegram structure: sWA FREchoFilter					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sWA	73 57 41
Command	Set echo Filter	String	12	FREchoFilter	46 52 45 63 68 6F 46 69 6C 74 65 72
Example: sWA FREchoFilter					
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				
Binary	Not available in V1.10 firmware				



10.5 Fog Filter

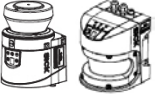

 PC	→	 LMS 1xx/5xx			
Telegram structure: sWN MSsuppmode					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sWN	73 57 41
Command	Set fog filter	String	10	MSsuppmode	4D 53 73 75 70 70 6D 6F 64 65
Status code		Bool_1	1	0 Glitch 1 Fog	00 Glitch 01 Fog
Example: sWN MSsuppmode					
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				
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				

 LMS 1xx/5xx	→	 PC			
Telegram structure: sWA MSsuppmode					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sWA	73 57 41
Command	Cont. values	String	6	LCMcfg	4C 43 4D 63 66 67
Example: sWA MSsuppmode					
ASCII	<STX>sWA{SPC}MSsuppmode<ETX>				
HEX	02 73 57 41 20 4D 53 73 75 70 70 6D 6F 64 65 03				
Binary	02 02 02 02 00 00 00 0F 73 57 41 20 4D 53 73 75 70 70 6D 6F 64 65 20 7E				


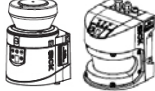
11 Encoder

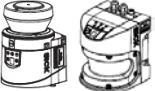

11.1 Increment source

 PC	→	 LMS 1xx/5xx			
Telegram structure: sWN LICsrc					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sWN	73 57 4E
Command	Set increment source	String	6	LICsrc	4C 49 43 73 72 63
Increment source		Enum_8	1	0 = Fixed speed 1 = Encoder	00 = Fixed speed 01 = Encoder
Example: sWN LICsrc					
ASCII	<STX>sWN{SPC}LICsrc{SPC}0<ETX>				
HEX	02 73 57 4E 20 4C 49 43 73 72 63 20 30 03				
Binary	02 02 02 02 00 00 00 0C 73 57 4E 20 4C 49 43 73 72 63 20 01 4F				


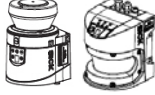
 LMS 1xx/5xx	→	 PC			
Telegram structure: sWA LICsrc					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sWA	73 57 41
Command		String	6	LICsrc	4C 49 43 73 72 63
Example: sWA LICsrc					
ASCII	<STX>sWA{SPC}LICsrc<ETX>				
HEX	02 73 57 41 20 4C 49 43 73 72 63 03				
Binary	02 02 02 02 00 00 00 0B 73 57 41 20 4C 49 43 73 72 63 41				

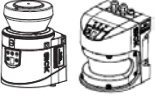

11.2 Encoder Settings

 PC	→	 LMS 1xx/5xx			
Telegram structure: sWN LICencset					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	string	3	sWN	73 57 4E
Command	Set encoder settings	string	9	LICencset	4C 49 43 65 6E 63 73 65 74
Encoder setting		Enum_8	1	0 = Off 1 = single Increment/INC1 2 = Direction recognition (phase) 3 = Direction recognition (level)	00 01 02 03
Example: sWN LICencset					
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				
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				


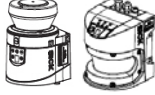
 LMS 1xx/5xx	→	 PC			
Telegram structure: sWA LICencset					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sWA	73 57 41
Command		String	9	LICencset	4C 49 43 65 6E 63 73 65 74
Example: sWA LICencset					
ASCII	<STX>sWA{SPC}LICencset<ETX>				
HEX	02 73 57 41 20 4C 49 43 65 6E 63 73 65 74 03				
Binary	02 02 02 02 00 00 00 0E 73 57 41 20 4C 49 43 65 6E 63 73 65 74 29				

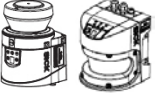

11.3 Encoder resolution

 PC	→	 LMS 1xx/5xx			
Telegram structure: sWN LICencres					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sWN	73 57 4E
Command	Set encoder resolution	String	9	LICencres	4C 49 43 65 6E 63 72 65 73
Encoder resolution				+0.001..+2000	
Example: sWN LICencres					
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				
Binary					

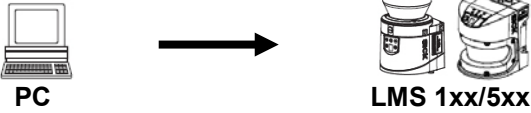
 LMS 1xx/5xx	→	 PC			
Telegram structure: sWN LICencres					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sWA	73 57 41
Command	encoder resolution	String	9	LICencres	4C 49 43 65 6E 63 72 65 73
Example: sWA LICencres					
ASCII	<STX>sWA{SPC}LICencres<ETX>				
HEX	02 73 57 41 20 4C 49 43 65 6E 63 72 65 73 03				
Binary					


11.4 Fixed speed

 PC	→	 LMS 1xx/5xx			
Telegram structure: sWN LICFixVel					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sWN	73 57 4E
Command	Set fixed speed	String	8	LICFixVel	4C 49 43 46 69 78 56 65 6C
Fixed speed				+0.001..+10	
Example: sWN LICFixVel					
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				
Binary					



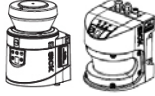
 LMS 1xx/5xx	→	 PC			
Telegram structure: sWN LICFixVel					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sWA	73 57 41
Command	fixed speed	String	8	LICFixVel	4C 49 43 46 69 78 56 65 6C
Example: sWA LICFixVel					
ASCII	<STX>sWA{SPC}LICFixVel<ETX>				
HEX	02 73 57 41 20 4C 49 43 46 69 78 56 65 6C 03				
Binary					

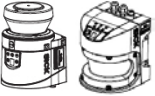


11.5 Ask speed threshold

					
Telegram structure: sRN LICSpTh					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sRN	73 52 4E
Command	Ask speed threshold	String	7	LICSpTh	4C 49 43 53 70 54 68
Example: sRN LICSpTh					
ASCII	<STX>sRN{SPC}LICSpTh<ETX>				
HEX	02 73 52 4E 20 4C 49 43 53 70 54 68 03				
Binary	02 02 02 02 00 00 00 0D 73 52 4E 20 4C 49 43 53 70 54 68 16				

					
Telegram structure: sRA LICSpTh					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sRA	73 52 41
Command	Ask speed threshold	String	7	LICSpTh	4C 49 43 53 70 54 68
Example: sRA LICSpTh					
ASCII	<STX>sRA{SPC}LICSpTh<ETX>				
HEX	02 73 52 41 20 4C 49 43 53 70 54 68 03				
Binary	02 02 02 02 00 00 00 0D 73 52 41 20 4C 49 43 53 70 54 68 20 05 3C				


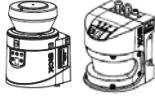
11.6 Encoder speed

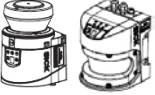

					
PC		LMS 1xx/5xx			
Telegram structure: sRN LICencsp					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sRN	73 52 4E
Command	Ask encoder speed	String	8	LICencsp	4C 49 43 65 6E 63 73 70
Encoder speed				00000000.. FFFFFFFF	00000000.. FFFFFFFF
Example: sRN LICencsp					
ASCII	<STX>sRN{SPC}LICencsp<ETX>				
HEX	02 73 52 4E 20 4C 49 43 65 6E 63 73 70 03				
Binary	02 02 02 02 00 00 00 0C 73 52 4E 20 4C 49 43 65 6E 63 73 70 62				

					
LMS 1xx/5xx		PC			
Telegram structure: sRA LICencsp					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sRA	73 52 41
Command	Ask speed threshold	String	7	LICSpTh	4C 49 43 53 70 54 68
Example: sRA LICencsp					
ASCII	<STX>sRA{SPC}LICencsp{SPC}00000000<ETX>				
HEX	02 73 52 41 20 4C 49 43 65 6C 63 73 70 20 30 30 30 30 30 30 30 03				
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				

12 Outputs

12.1 Ask state of the outputs

 PC	→	 LMS 1xx/5xx			
Telegram structure: sRN LIDoutputstate					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sRN	73 52 4E
Command	Ask output state	String	14	LIDoutputstate	4C 49 44 6F 75 74 70 75 74 73 74 61 74 65
Example: sRN LIDoutputstate					
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				
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				

 LMS 1xx/5xx	→	 PC			
Telegram structure: sRA LIDoutputstate					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sRA	73 52 41
Command	Output state	String	14	LIDoutputstate	4C 49 44 6F 75 74 70 75 74 73 74 61 74 65
Status Code		Uint_16	2	0	00 00
State of the outputs and count value in hex. (Values of an example)	Out1 State	Uint_32	4	0	00 00 00 00
	Out1 Count	Enum_8	1	0..2	00 - 02
	Out2 State	Uint_32	4	0	00 00 00 00
	Out2 Count	Enum_8	1	0..2	00 - 02
	Out3 State	Uint_32	4	0	00 00 00 00
	Out3 Count	Enum_8	1	0..2	00 - 02
	Out4 State*	Uint_32	4	0	00 00 00 00
	Out4 Count*	Enum_8	1	0..2	00 - 02
	Out5 State*	Uint_32	4	0	00 00 00 00
	Out5 Count*	Enum_8	1	0..2	00 - 02
	Out6 State*	Uint_32	4	0	00 00 00 00
	Out6 Count*	Enum_8	1	0..2	00 - 02
	ext.Out1 State	Uint_32	4	0	00 00 00 00
	ext.Out1 Count	Enum_8	1	0..2	00 - 02


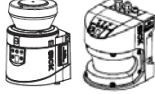
ext.Out2 State	Enum_8	1	0..2	00 - 02
ext.Out2 Count	Uint_32	4	0	00 00 00 00
ext.Out3 State	Enum_8	1	0..2	00 - 02
ext.Out3 Count	Uint_32	4	0	00 00 00 00
ext.Out4 State	Enum_8	1	0..2	00 - 02
ext.Out4 Count	Uint_32	4	0	00 00 00 00
ext.Out5 State	Enum_8	1	0..2	00 - 02
ext.Out5 Count	Uint_32	4	0	00 00 00 00
ext.Out6 State	Enum_8	1	0..2	00 - 02
ext.Out6 Count	Uint_32	4	0	00 00 00 00
ext.Out7 State	Enum_8	1	0..2	00 - 02
ext.Out7 Count	Uint_32	4	0	00 00 00 00
ext.Out8 State	Enum_8	1	0..2	00 - 02
ext.Out8 Count	Uint_32	4	0	00 00 00 00
reserved	Uint_16	2	0	00
			States: 0 = low 1 = High 2 = Tristate (undefined)	States: 00 = low 01 = High 02 = Tristate (undefined)

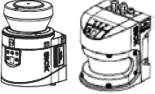

Example: sRA LIDoutputstate

ASCII	<STX>sRA{SPC}LIDoutputstate{SPC}0{SPC}0{SPC}1{SPC}7{SPC}1{SPC}D{SPC}1{SPC}0{SPC}1{SPC}7{SPC}1{SPC}0{SPC}1{SPC}0{SPC}1{SPC}14{SPC}1{SPC}0{SPC}1{SPC}0{SPC}1{SPC}0{SPC}1{SPC}35{SPC}0<ETX>
HEX	02 73 52 41 20 4C 49 44 6F 75 74 70 75 74 73 74 61 74 65 20 30 20 30 20 31 20 30 20 31 20 30 20 31 20 30 20 32 20 30 20 32 20 30 20 32 20 30 20 32 20 30 20 32 20 30 20 32 20 30 20 32 20 30 20 32 20 30 20 32 20 30 20 32 20 30 20 30 03
Binary	02 02 02 02 00 00 00 52 73 52 41 20 4C 49 44 6F 75 74 70 75 74 73 74 61 74 65 20 00 00 00 00 00 00 01 00 00 00 00 01 00 00 00 00 02 00 00 00 00 02 00 00 00 00 02 00 00 00 00 02 00 00 00 00 02 00 00 00 00 02 00 00 00 00 00 00 48



*only in LMS5xx, not available in LMS1xx and not in the telegram there



12.2 Set output state

 PC	→	 LMS 1xx/5xx			
Telegram structure: sMN mDOSetOutput					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sMN	73 4D 4E
Command	Set output state	String	12	mDOSetOutput	6D 44 4F 53 65 74 4F 75 74 70 75 74
Output Number		Uint_8	1	1 - 3	1 - 3
Output State		Enum_8	1	0 = inactive 1 = active	00 = inactive 01 = active
Example: sMN mDOSetOutput					
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				
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				



 LMS 1xx/5xx	→	 PC			
Telegram structure: sAN mDOSetOutput					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sAN	73 41 4E
Command	Output state	String	12	mDOSetOutput	6D 44 4F 53 65 74 4F 75 74 70 75 74
Status Code		Bool_1		1 = Success 0 = Error	01 = Success 00 = Error
Example: sAN mDOSetOutput					
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				
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 00 67				



12.3 Change output 6/3 function

 PC	→	 LMS 5xx			
Telegram structure PRO: sWN DO6Fnc Telegram structure LITE: sWN DO3Fnc					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sWN	73 57 4E
Command	Output function	String	6	DO6Fnc/DO3Fnc	44 4F 36 46 6E 63 / 44 4F 33 46 6E 63
Output State		Enum_8	1	0 = No Function 1 = Command 2 = Device Ready 3 = Application 4 = Applic/Dev.Ready 5 = Dev.Ready/Poll. 6 = Pollution 7 = Zero index (Master sync)	
Example: sWN DO6Fnc → Out6 to master sync:					
ASCII	<STX>sWN{SPC}DO6Fnc{SPC}7<ETX>				
HEX	02 73 57 4E 20 44 4F 36 46 6E 63 20 37 03				
Binary	Not available with firmware V1.10				


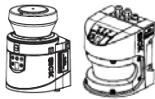
 LMS 5xx	→	 PC			
Telegram structure: sAN DO6Fnc					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sWN	73 57 41
Command	Output function	String	12	DO6Fnc	44 4F 36 46 6E 63
Example: sAN DO6Fnc					
ASCII	<STX>sWA{SPC}DO6Fnc<ETX>				
HEX	02 73 57 41 20 44 4F 36 46 6E 63 03				
Binary	Not available with firmware V1.10				

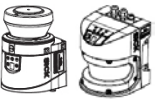

12.4 Change Input 4 function

 PC	→	 LMS 5xx			
Telegram structure: sWN DO6Fnc					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sWN	73 57 4E
Command	Input function	String	10	DO3And4Fnc	44 4F 33 41 6E 64 34 46 6E 63
Output State		Enum_8	1	0 = No Function 1 = Encoder 2 = Slave Sync 3 = Digital Input	
Example: sWN In4 → In3+4 to Slave Sync					
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				
Binary	Not available with firmware V1.10				

 LMS 5xx	→	 PC			
Telegram structure: sWA DO3And4Fnc					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sWN	73 57 41
Command	Output function	String	10	DO3And4Fnc	44 4F 33 41 6E 64 34 46 6E 63
Example: sWA DO3And4Fnc					
ASCII	<STX>sWA{SPC}DO3And4Fnc<ETX>				
HEX	02 73 57 41 20 44 4F 33 41 6E 64 34 46 6E 63 03				
Binary	Not available with firmware V1.10				



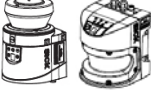
12.5 Reset output counter

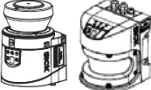


 PC	→	 LMS 1xx/5xx			
Telegram structure: sMN LIDrstoutpnt					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sMN	73 4D 4E
Command	Reset output counter	String	13	LIDrstoutpnt	4C 49 44 72 73 74 6F 75 74 70 63 6E 74
Example: sMN LIDrstoutpnt					
ASCII	<STX>sMN{SPC}LIDrstoutpnt<ETX>				
HEX	02 73 4D 4E 20 4C 49 44 72 73 74 6F 75 74 70 63 6E 74 03				
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				

 LMS 1xx/5xx	→	 PC			
Telegram structure: sAN LIDrstoutpnt					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sAN	73 41 4E
Command	Reset state	String	13	LIDrstoutpnt	4C 49 44 72 73 74 6F 75 74 70 63 6E 74
Status Code		Bool_1	1	0 = Success	00 = Success
Example: sAN LIDrstoutpnt					
ASCII	<STX>sAN{SPC}LIDrstoutpnt{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				
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				

13 Other Commands

13.1 Device Ident

 PC		 LMS 1xx/5xx			
Telegram structure: sRN DevicelIdent/sRI 0					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sRN	73 52 4E
			3	sRI	
Command	Ask Ident	String	11	DevicelIdent	44 65 76 69 63 65 49 64 65 6E 74
			1	0	
Example: sRN DevicelIdent					
ASCII	<code><STX>sRN{SPC}DevicelIdent<ETX></code> or <code><STX>sRI{SPC}0<ETX></code>				
HEX	<code>02 73 52 4E 20 44 65 76 69 63 65 49 64 65 6E 74 03</code> or <code>02 73 52 49 20 30 03</code>				
Binary	<code>02 02 02 02 00 00 00 0F 73 52 4E 20 44 65 76 69 63 65 49 64 65 6E 74 25</code>				

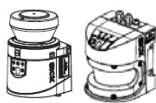
 LMS 1xx/5xx		 PC			
Telegram structure: sRA DevicelIdent					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sRA	73 52 41
Command	Start the device	String	3	DevicelIdent	44 65 76 69 63 65 49 64 65 6E 74
Ident String	Ident information	String		See examples	

Example: sRA Devicident

ASCII	<STX> sRA{SPC}Devicident 10 LMS10x_FieldEval 10 V1.36-21.10.2010 <ETX>
HEX	Always ASCII answer
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

13.2 Device State

PC



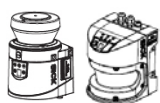
LMS 1xx/5xx

Telegram structure: sRN SCdevicestate

Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sRN	73 52 4E
Command	Ask state	String	11	SCdevicestate	53 43 64 65 76 69 63 65 73 74 61 74 65

Example: sRN device state

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
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



LMS 1xx/5xx



PC

Telegram structure: sRN device state

Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sRA	73 52 41
Command	Ask state	String	11	SCdevicestate	53 43 64 65 76 69 63 65 73 74 61 74 65
Status Code		Enum_8	1	0 Busy 1 Ready 2 Error	00 Busy 01 Ready 02 Error

Example: sRN device state

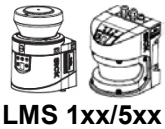
ASCII	<STX>sRA{SPC}SCdevicestate{SPC}0<ETX>
HEX	02 73 52 41 20 53 43 64 65 76 69 63 65 73 74 61 74 65 20 00 03
Binary	02 02 02 02 00 00 00 13 73 52 41 20 53 43 64 65 76 69 63 65 73 74 61 74 65 20 00 1F

13.3 Device Name**13.3.1 Set Device Name****Telegram structure: sWN LocationName**

Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sWN	73 57 4E
Command	Set Device name	String	12	LocationName	4C 6F 63 61 74 69 6F 6E 4E 61 6D 65
Value	array of visible characters with preceding current length	Uint_16	2	0000h - 0010h	00 00h – 00 10h
Value	Device Name	String	16		

Example: sWN LocationName D OutdoorDevice

ASCII	<STX>sWN{SPC}LocationName{SPC}D{SPC}OutdoorDevice<ETX>
HEX	02 73 57 4E 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
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 44 20 4F 75 74 64 6F 6F 72 44 65 76 69 63 65 71



Telegram structure: sWA LocationName

Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sWA	73 57 41
Command	Ask state	String	11	LocationName	4C 6F 63 61 74 69 6F 6E 4E 61 6D 65
Value	array of visible characters with preceding current length	Uint_16	2	0000h - 0010h	00 00h – 00 10h
Value	Device Name	String	16		

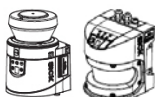
Example: sWA LocationName D OutdoorDevice

ASCII	<STX>sWA{SPC}LocationName{SPC}D{SPC}OutdoorDevice<ETX>
HEX	02 73 57 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
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 20 00 0D 4F 75 74 64 6F 6F 72 44 65 76 69 63 65 17

13.3.2 Ask Device Name



PC



LMS 1xx/5xx

Telegram structure: sRN LocationName

Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sRN	73 52 4E
Command	Set Device name	String	12	LocationName	4C 6F 63 61 74 69 6F 6E 4E 61 6D 65

Example: sRN LocationName

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
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



LMS 1xx/5xx



PC


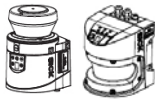
Telegram structure: sRA LocationName

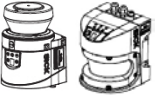

Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sRA	73 52 41
Command	Set Device name	String	12	LocationName	4C 6F 63 61 74 69 6F 6E 4E 61 6D 65
Value	array of visible characters with preceding current length	Uint_16	2	0000h - 0010h	00 00h – 00 10h
Value	Device Name	String	16		

Example: sRA LocationName

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
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


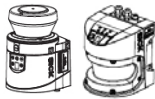
13.4 Operating hours

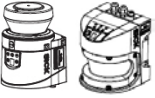

 PC	→	 LMS 1xx/5xx			
Telegram structure: sRN ODoprh					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sRN	73 52 4E
Command	Ask operating hours	String	6	ODoprh	4F 44 6F 70 72 68
Example: sRN ODoprh					
ASCII	<STX>sRN{SPC}ODoprh<ETX>				
HEX	02 73 52 4E 20 4F 44 6F 70 72 68 03				
Binary	02 02 02 02 00 00 00 0A 73 52 4E 20 4F 44 6F 70 72 68 41				

 LMS 1xx/5xx	→	 PC			
Telegram structure: sRA ODoprh					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sRA	73 52 41
Command	Ask operating hours	String	6	ODoprh	4F 44 6F 70 72 68
Value	Operating hours in 1/10h	Uint_32	4	00000000h – FFFFFFFFh	00 00 00 00h – FF FF FF FFh
Example: sRA ODoprh					
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				
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				


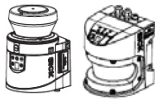
Calculation of the value: 0x2DC8B (hex) → 187531 (dez) x 1h/10 = 18753.1 h

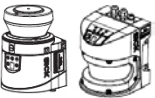

13.5 Power On Counter

 PC	→	 LMS 1xx/5xx			
Telegram structure: sRN ODpwrC					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sRN	73 52 4E
Command	Ask operating hours	String	6	ODpwrC	4F 44 70 77 72 63
Example: sRN ODpwrC					
ASCII	<STX>sRN{SPC}ODpwrC<ETX>				
HEX	02 73 52 4E 20 4F 44 70 77 72 63 03				
Binary	02 02 02 02 00 00 00 0A 73 52 4E 20 4F 44 70 77 72 63 52				



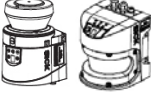
 LMS 1xx/5xx	→	 PC			
Telegram structure: sRA ODpwrC					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sRA	73 52 41
Command	Ask operating hours	String	6	ODpwrC	4F 44 70 77 72 63
Value	Power on Counter	Uint_32	4	00000000h – FFFFFFFFh	00 00 00 00h – FF FF FF FFh
Example: sRA ODpwrC					
ASCII	<STX>sRA{SPC}ODoprh{SPC}752D<ETX>				
HEX	02 73 52 41 20 4F 44 70 77 72 63 20 752D 03				
Binary	02 02 02 02 00 00 00 0F 73 52 41 20 4F 44 70 77 72 63 20 00 00 00 58 36				

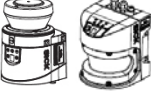


13.6 IP-Address

 PC	→	 LMS 1xx/5xx			
Telegram structure: sWN EIlpAddr					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sWN	73 57 4E
Command	Set IP-address	String	8	EIlpAddr	45 49 49 50 41 64 64 72
IP-address	Set values in hex	UInt32	4	00 00 00 00h	
Example (192.168.0.1): sWN EIlpAddr					
ASCII	<STX>sWN{SPC}EIlpAddr{SPC}C0{SPC}A8{SPC}0{SPC}1<ETX>				
HEX	02 73 57 4E 20 45 49 49 70 41 64 64 72 20 43 30 20 41 38 20 30 20 31 03				
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 01 05				


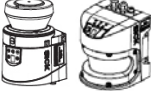
 LMS 1xx/5xx	→	 PC			
Telegram structure: sWA EIlpAddr					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sWA	73 57 41
Command	Set IP-address	String	8	EIlpAddr	45 49 49 50 41 64 64 72
Example: sWA EIlpAddr					
ASCII	<STX>sWA{SPC}EIlpAddr<ETX>				
HEX	02 73 57 41 20 45 49 49 70 41 64 64 72 03				
Binary	02 02 02 02 00 00 00 0D 73 57 41 20 45 49 49 70 41 64 64 72 20 63				

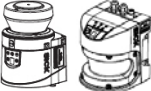

13.7 Set factory defaults

					
PC		LMS 1xx/5xx			
Telegram structure: sMN mSCloadfacdef					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sMN	
Command	Load factory defaults	String	13	mSCloadfacdef	Binary not possible
Example: sMN mSCloadfacdef					
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				
Binary	Not possible				

					
LMS 1xx/5xx		PC			
Telegram structure: sNA mSCloadfacdef					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sAN	
Command	Load factory defaults	String	13	mSCloadfacdef	Binary not possible
Example: sAN mSCloadfacdef					
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				
Binary	Not possible				


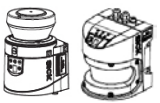
13.8 Reboot Device

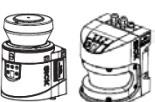

 PC	→	 LMS 1xx/5xx			
Telegram structure: sMN mSCreboot					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sMN	73 4D 4E
Command	Reboot device	String	9	mSCreboot	6D 53 43 72 65 62 6F 6F 74
(includes saving all parameters)					
Example: sMN mSCreboot					
ASCII	<STX> sMN{SPC}mSCreboot <ETX>				
HEX	02 73 4D 4E 20 6D 53 43 72 65 62 6F 6F 74 03				
Binary	02 02 02 02 00 00 00 0D 73 4D 4E 20 6D 53 43 72 65 62 6F 6F 74 2C				

 LMS 1xx/5xx	→	 PC			
Telegram structure: sAN mSCreboot					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sAN	73 41 4E
Command	Reboot device	String	9	mSCreboot	6D 53 43 72 65 62 6F 6F 74
Example: sAN mSCreboot					
ASCII	<STX> sAN{SPC}mSCreboot <ETX>				
HEX	02 73 41 4E 20 6D 53 43 72 65 62 6F 6F 74 03				
Binary	02 02 02 02 00 00 00 0E 73 41 4E 20 6D 53 43 72 65 62 6F 6F 74 00				

13.9 Contamination Measurement

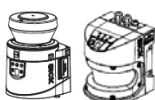
13.9.1 Set Contamination values

 PC	→	 LMS 1xx/5xx			
Telegram structure: sWN LCMcfg					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sWN	73 57 4E
Command	Contamination config	String	6	LCMcfg	4C 43 4D 63 66 67
Strategy		Enum_8	1	0 inactive 1 high available 2 available 3 sensitive 4 semi-sensitive	00 inactive 01 high available 02 available 03 sensitive 04 semi-sensitive
Response time		Uint_32	4	1...60	
Threshold warning		Uint_32	4	+0...+100	
Threshold error		Uint_32	4	+0...+100	
Example: sWN LCMcfg					
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 33 30 20 2B 36 35 20 2B 34 35 03				
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				

 LMS 1xx/5xx	→	 PC			
Telegram structure: sWA LCMcfg					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sWA	73 57 41
Command	Cont. values	String	6	LCMcfg	4C 43 4D 63 66 67

Example: sWA LCMcfg

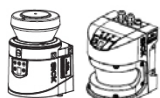
ASCII	<STX>sWA{SPC}LCMcfg<ETX>
HEX	02 73 57 41 20 4C 43 4D 63 66 67 03
Binary	02 02 02 02 00 00 00 0B 73 57 41 20 4C 43 4D 63 66 67 20 45

13.9.2 Ask for contamination settings**PC****LMS 1xx/5xx****Telegram structure: sRN LCMcfg**

Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sRN	73 52 4E
Command	Ask for settings	String	6	LCMcfg	4C 43 4D 63 66 67

Example: sRN LCMcfg

ASCII	<STX>sRN{SPC}LCMcfg<ETX>
HEX	02 73 57 4E 20 4C 43 4D 63 66 67 03
Binary	02 02 02 02 00 00 00 0A 73 52 4E 20 4C 43 4D 63 66 67 6F

**LMS 1xx/5xx****PC****Telegram structure: sWA LCMcfg**

Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sRA	73 52 4E
Command	Ask for settings	String	6	LCMcfg	4C 43 4D 63 66 67
Strategy		Enum_8	1		00 inactive 01 high available 02 available 03 sensitive 04 semi-sensitive
Response time		Uint_16	2	1...60	00 00h..00 3Ch
Threshold warning		Uint_16	2	+0...+100	00 00h..00 64h
Threshold error		Uint_16	2	+0...+100	00 00h..00 64h

Example: sWA LCMcfg

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
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

13.10 Synchronization Phase

PC



LMS 5xx

Telegram structure: sWN SYPhase

Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	SWN	73 57 4E
Command	Set sync phase	String	7	SYPhase	53 59 50 68 61 73 65

Example: sWN SYPhase +90

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
Binary	Not available with firmware V1.10



LMS 5xx



PC


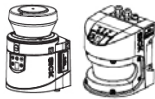
Telegram structure: sWA SYPhase

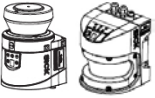

Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sWA	73 57 41
Command	Sync phase	String	7	SYPhase	53 59 50 68 61 73 65

Example: sWA SYPhase

ASCII	<STX>sWA{SPC}SYPhase<ETX>
HEX	02 73 57 41 20 53 59 50 68 61 73 65 03
Binary	Not available with firmware V1.10

13.11 Function Front Panel

 PC	→	 LMS 1xx/5xx			
Telegram structure: sWN LMLfpFcn					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sWN	73 57 4E
Command	Set function of the front panel	String	8	LMLfpFcn	4C 4D 4C 66 70 46 63 6E
Reserved		Bool	1	1	01
LED Function Q1/Q2		Enum_8	1	0 = No Function 1 = Application 2 = Command	00 = No Function 01 = Application 02 = Command
LED Function OK/Stop		Enum_8	1	0 = No Function 1 = Application 2 = Command	00 = No Function 01 = Application 02 = Command
Display Function		Enum_8	1	0 = Application 1 = Command	00 = Application 01 = Command
Example: sWN LMLfpFcn					
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				
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				

 LMS 1xx/5xx	→	 PC			
Telegram structure: sWN LMLfpFcn					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sWA	73 57 41
Command	Front LED function	String	10	mLMLSetLed	4C 4D 4C 66 70 46 63 6E

Example: sWA LMLfpFcn

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

Frontpanel



Frontpanel enable



Function LED Q1 Q2

Function LED OK STOP


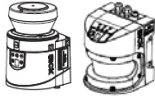
7-Segment-Display

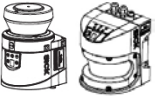

14 Standby Mode

 PC	→	 LMS 1xx/5xx	-----Only LMS1xx-----		
Telegram structure: sMN LMCstandby					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sMN	73 4D 4E
Command	Set device to Standby	String	10	LMCstandby	4C 4D 43 73 74 61 6E 64 62 79
Example: sMN LMCstandby					
ASCII	<STX>sMN{SPC}LMCstandby<ETX>				
HEX	02 73 4D 4E 20 4C 4D 43 73 74 61 6E 64 62 79 03				
Binary	02 02 02 02 00 00 00 0E 73 4D 4E 20 4C 4D 43 73 74 61 6E 64 62 79 65				

 LMS 1xx/5xx	→	 PC			
Telegram structure: sAN LMCstandby					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	String	3	sAN	73 41 4E
Command	Set device to Standby	String	10	LMCstandby	4C 4D 43 73 74 61 6E 64 62 79
Status Code	accepted when value is 0	Enum_8	1	0 no Error	00 no Error
Example: sAN LMCstandby					
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				
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				

15 Start Measurement

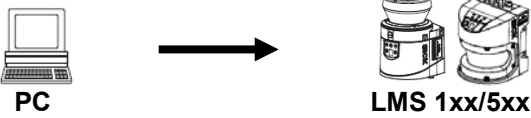
 PC	→	 LMS 1xx/5xx			
Telegram structure: sMN LMCstartmeas					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	string	3	sMN	73 4D 4E
Command	Start Measurement	string	12	LMCstartmeas	4C 4D 43 73 74 61 72 74 6D 65 61 73
Example: sMN LMCstartmeas					
ASCII	<STX>sMN{SPC}LMCstartmeas<ETX>				
HEX	02 73 4D 4E 20 4C 4D 43 73 74 61 72 74 6D 65 61 73 03				
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				

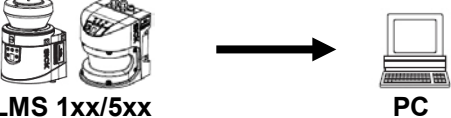
 LMS 1xx/5xx	→	 PC			
Telegram structure: sAN LMCstartmeas					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	string	3	sAN	73 41 4E
Command	Start Measurement	string	12	LMCstartmeas	20 4C 4D 43 73 74 61 72 74 6D 65 61 73
Status Code	accepted when value is 0	Enum8	1	0 no Error 1 not allowed	00 no Error 01 not allowed
Example: sAN LMCstartmeas					
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				
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				

```

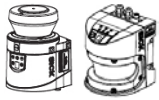
Connecting to 192.168.1.112 ...
TCP connection error :10061
Connecting to 192.168.1.112 ...
Connected to 192.168.1.112
sMN SetAccessMode 03 F4724744sAN SetAccessMode
1sMN LMCstartmeas sAN LMCstartmeas 0
  
```

16 Stop Measurement

					
Telegram structure: sMN LMCstopmeas					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	string	3	sMN	73 4D 4E
Command	Stop Measurement	string	11	LMCstopmeas	4C 4D 43 73 74 6F 70 6D 65 61 73
Example: sMN LMCstopmeas					
ASCII	<STX>sMN{SPC}LMCstopmeas<ETX>				
HEX	02 73 4D 4E 20 4C 4D 43 73 74 6F 70 6D 65 61 73 03				
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				

					
Telegram structure: sAN LMCstopmeas					
Telegram	Description	Variable	Length	Values ASCII	Values Binary
Command Type	Sopas by name	string	3	sAN	73 41 4E
Command	Stop Measurement	string	11	LMCstopmeas	4C 4D 43 73 74 6F 70 6D 65 61 73
Status Code	accepted when value is 0	Enum8	1	0 no Error 1 not allowed	00 no Error 01 not allowed
Example: sAN LMCstopmeas					
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				
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				

17 Sopas Error Codes



LMS 1xx/5xx

sFA x



PC

Name	Number	Explanation
Sopas_Ok	0	No error
Sopas_Error_METHODIN_ACCESSDENIED	1	Wrong userlevel, access to method not allowed.
Sopas_Error_METHODIN_UNKNOWNINDEX	2	Trying to access a method with an unknown Sopas index.
Sopas_Error_VARIABLE_UNKNOWNINDEX	3	Trying to access a variable with an unknown Sopas index
Sopas_Error_LOCALCONDITIONFAILED	4	Local condition violated, e.g. giving a value that exceeds the minimum or maximum allowed value for this variable
Sopas_Error_INVALID_DATA	5	Invalid data given for variable, this errorcode is deprecated (is not used anymore)
Sopas_Error_UNKNOWN_ERROR	6	An error with unknown reason occurred, this errorcode is deprecated
Sopas_Error_BUFFER_OVERFLOW	7	The communication buffer was too small for the amount of data that should be serialised
Sopas_Error_BUFFER_UNDERFLOW	8	More data was expected, the allocated buffer could not be filled.
Sopas_Error_ERROR_UNKNOWN_TYPE	9	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.
Sopas_Error_VARIABLE_WRITE_ACCESSDENIED	10	It is not allowed to write values to this variable. Probably the variable is defined as read-only
Sopas_Error_UNKNOWN_CMD_FOR_NAMESERVER	11	When using names instead of indices, a command was issued that the nameserver does not understand
Sopas_Error_UNKNOWN_COLA_COMMAND	12	The CoLa protocol specification does not define the given command, command is unknown
Sopas_Error_METHODIN_SERVER_BUSY	13	It is not possible to issue more than one command at a time to an SRT device.
Sopas_Error_FLEX_OUT_OF_BOUNDS	14	An array was accessed over its maximum length
Sopas_Error_EVENTREG_UNKNOWNINDEX	15	The event you wanted to register for does not exist, the index is unknown
Sopas_Error_COLA_A_VALUE_OVERFLOW	16	The value does not fit into the value field, it is

Name	Number	Explanation
		too large
Sopas_Error_COLA_A_INVALID_CHARACTER	17	Character is unknown, probably not alphanumeric
Sopas_Error_OSAI_NO_MESSAGE	18	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.
Sopas_Error_OSAI_NO_ANSWER_MESSAGE	19	This is the same as Sopas_Error_OSAI_NO_MESSAGE with the difference that it is thrown when trying to PUT a variable.
Sopas_Error_INTERNAL	20	Internal error in the firmware, probably a pointer to a parameter was null
Sopas_Error_HubAddressCorrupted	21	The Sopas Hubaddress is either too short or too long.
Sopas_Error_HubAddressDecoding	22	The Sopas Hubaddress is invalid, it can not be decoded (Syntax)
Sopas_Error_HubAddressAddressExceeded	23	Too many hubs in the address
Sopas_Error_HubAddressBlankExpected	24	When parsing a HubAddress an expected blank was not found. The HubAddress is not valid
Sopas_Error_AsyncMethodsAreSuppressed	25	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.
Sopas_Error_ComplexArraysNotSupported	26	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.

18 Problems

Every answer of the LMS starts with a separate framed string:

```
<STX>sSI 2 1<ETX><STX>"Answer"<ETX>
```

It is an event from Sopas, send command: <STX>sEN SCParmChngd 0<ETX> to deactivate that event

Australia

Phone +61 3 9497 4100
1800 334 802 - tollfree
E-Mail sales@sick.com.au

Belgium/Luxembourg

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

Brasil

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

Canada

Phone +1(952) 941-6780
1 800-325-7425 - tollfree
E-Mail info@sickusa.com

Ceská Republika

Phone +420 2 57 91 18 50
E-Mail sick@sick.cz

China

Phone +852-2763 6966
E-Mail ghk@sick.com.hk

Danmark

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

Deutschland

Phone +49 211 5301-301
E-Mail kundenservice@sick.de

España

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

France

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

Great Britain

Phone +44 (0)1727 831121
E-Mail info@sick.co.uk

India

Phone +91-22-4033 8333
E-Mail info@sick-india.com

Israel

Phone +972-4-999-0590
E-Mail info@sick-sensors.com

Italia

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

Japan

Phone +81 (0)3 3358 1341
E-Mail support@sick.jp

Magyarország

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

Nederlands

Phone +31 (0)30 229 25 44
E-Mail info@sick.nl

Norge

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

Österreich

Phone +43 (0)22 36 62 28 8-0
E-Mail office@sick.at

Polska

Phone +48 22 837 40 50
E-Mail info@sick.pl

România

Phone +40 356 171 120
E-Mail office@sick.ro

Russia

Phone +7 495 775 05 30
E-Mail info@sick.ru

Schweiz

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

Singapore

Phone +65 6744 3732
E-Mail admin@sicksgp.com.sg

South Africa

Phone +27 11 472 3733
E-Mail info@sickautomation.co.za

South Korea

Phone +82-2 786 6321/4
E-Mail info@sickkorea.net

Slovenija

Phone +386 (0)1-47 69 990
E-Mail office@sick.si

Suomi

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

Sverige

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

Taiwan

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

Türkiye

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

United Arab Emirates

Phone +971 4 8865 878
E-Mail info@sick.ae

USA/Canada/México

Phone +1(952) 941-6780
1 800-325-7425 - tollfree
E-Mail info@sickusa.com

More representatives and agencies
at www.sick.com