Leuze

Original operating instructions

ROD-300-500

Communication Protocol

Ethernet

We reserve the right to make technical changes
EN 2025/09/16 – 50153078

Leuze electronic GmbH + Co. KG
In der Braike 1
D-73277 Owen / Germany
Phone: +49 7021 573-0
Fax: +49 7021 573-199
http://www.leuze.com
info@leuze.de

1 Introduction 4
1.1 Scope 4
1.1.1 Definitions, acronyms and abbreviations 4

2 Hardware interface 5
2.1 Communication characteristics 5
2.2 Wiring diagram 5

3 Ethernet communication protocol 6
3.1 Command formats 6
3.1.1 Binary protocol 6
3.1.2 ASCII protocol 6
3.2 Data types of variables 6
3.3 Example and explanation 7

4 Commands 8
4.1 Request for MDI 8
4.1.1 Command: Start to send MDI 8
4.1.2 Command: Stop to send MDI 8
4.2 Read out 9
4.2.1 Command: Get protocol 9
4.2.2 Command: Get MDI data packet type 10
4.2.3 Command: Get angular resolution 11
4.2.4 Command: Get MDI data output direction 12
4.2.5 Command: Get angle range 13
4.2.6 Command: Get skip spots 14
4.2.7 Command: Get contamination threshold 15
4.2.8 Command: Get version 15
4.2.9 Command: Get internal temperature 16
4.2.10 Command: Get error log 17
4.2.11 Command: Get LED control 20
4.2.12 Command: Get lamp state 21
4.2.13 Command: Get ethernet configuration 22
4.2.14 Command: Get runtime hours 23
4.2.15 Command: Get device name 24
4.2.16 Command: Get window calibration status 25
4.2.17 Command: Get filter type 26
4.2.18 Command: Get wms value 27
4.2.19 Command: Get error code 28
4.2.20 Command: Get MDI transmission status 28
4.3 Configurate 29
4.3.1 Command: Set protocol 29
4.3.2 Command: Set MDI data packet type 30
4.3.3 Command: Set angular resolution 31
4.3.4 Command: Set MDI data output direction 32
4.3.5 Command: Set angle range 33
4.3.6 Command: Set skip spots 34
4.3.7 Command: Set contamination threshold 35
4.3.8 Command: Set LED control 36
4.3.9 Command: Set ethernet configuration 37
4.3.10 Command: Set device name 38
4.3.11 Command: Set window calibration 39
4.3.12 Command: Set filter type 40
4.3.13 Command: Reset 41
4.3.14 Command: Reboot device 42
4.4 MDI packet 42

1 Introduction

1.1 Scope

This document specifies the communication protocol at the ethernet output of the ROD 300/500 LiDAR.

1.1.1 Definitions, acronyms and abbreviations

This subsection provides the definitions of all terms, acronyms, and abbreviations required to properly interpret this document.

- AGV: Automated Guided Vehicle.
- MDI: Measured Distance Information.
- MRI: Measured Remissions Information
- LiDAR: Light Detection And Ranging

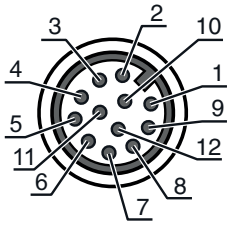
2 Hardware interface

2.1 Communication characteristics

The device has Ethernet interface, The commands are transferred only via TCP/IP, while the MDI data can be transferred via TCP/IP or UDP/IP.

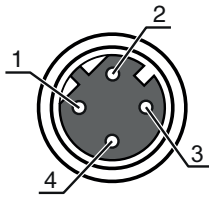
2.2 Wiring diagram

A-CODE: LEUZE ROD-xxx



PN	DESIGNATION
1	Output 1
2	VIN+
3	N.C.
4	N.C.
5	ERR
6	N.C.
7	GND
8	N.C.
9	N.C.
10	N.C.
11	N.C.
12	N.C.

D-CODE



PN	DESIGNATION
1	TX+
2	RX+
3	TX-
4	RX-

3 Ethernet communication protocol

3.1 Command formats

There are two types of command format: Binary and ASCII. Both types use Big endian.

3.1.1 Binary protocol

Binary communication frame			
Start	Length	DATA	CHK
6Bytes	2Bytes	0 – 22Bytes	1Bytes

The binary protocol of the LiDAR Detector has the format as the table shows and the command string can be converted from the ASCII commands described in this document. The binary protocol has a special framing so that the LiDAR can recognize the start of a binary telegram. The string must start with 1 STX symbols (02) and 4C 45 55 5A 45 that is followed by the length of the telegram in HEX (i.e.:00 0E). Then are the DATA zone. Last is the check sum.

Example:

Binary: 02 4C 45 55 5A 45 | 00 0E | 63 57 4E 20 53 65 74 49 50 20 C0 A8 01 01 | 49

Special Characters: Header: 02 4C 45 55 5A 45; Length: 00 0E; Space: 20; Checksum: 49

The length can be created by counting every letter of the DATA (without checksum and framing but with blanks) and convert the sum into HEX(all dark blue area). Zeros are added in front until a string of eight characters is built.The command itself starts after the length characters. Every single letter of the written command is converted to HEX turning into a pair of numbers, followed by a blank and then the parameters also converted in HEX. In between parameters there are no blanks. The “Checksum” is built with XOR beginning the calculation right after the length (i.e. all dark blue area).

3.1.2 ASCII protocol

The framing of a telegram in ASCII is a <STX> at the start and an <ETX> at the end of each telegram. Commands are written as letters, followed by the parameters as defined in this document. There must be a blank in between the command and the parameters and in between each parameter (as shown in the example below as_).

Example:

<STX>cWN_SetIP_192_168_1_1<ETX>

In HEX the command starts with 02 and ends with 03. The spaces are marked as 20.

HEX:

02 63 57 4E 20 53 65 74 49 50 20 31 39 32 20 31 36 38 20 31 20 31 03

(Both examples show the same command and parameters.)

3.2 Data types of variables

The following data types are specified for the variable values in the telegram:

Type	Size(bits)	Range of Value(decimal)	Sign
Uint8	8	0 – 255	unsigned
Int8	8	-128 – +128	signed
Uint16	16	0 – 65535	unsigned
Int16	16	-32768 – +32767	signed
Uint32	32	0 – 4294967295	unsigned

Type	Size(bits)	Range of Value(decimal)	Sign
Int32	32	-2147483648 – +2147483647	signed
Enum8	8	0 – 255	unsigned
String	Depends on content		

3.3 Example and explanation

The different between the two kind of protocol is shown in the following table.

Description	Value ASCII	Value Hex	Value Binary
Start of text	<STX>	02	02 4C 45 55 5A 45+ given length
End of text	<ETX>	03	Check sum
Command read by name	cRN	63 52 4E	63 52 4E
Command write by name	cWN	63 57 4E	63 57 4E
Response read command	cRA	63 52 41	63 52 41
Response write command	cWA	63 57 41	63 57 41
Space	{SPC}	20	20

The following table shows the example of different protocol

cWN SetIP 192 168 1 1	
ASCII	<STX>cWN{SPC}SetIP{SPC}192{SPC}168{SPC}1{SPC}1<ETX>
HEX	02 63 57 4E 20 53 65 74 49 50 20 31 39 32 20 31 36 38 20 31 20 31 03
Binary	02 4C 45 55 5A 45 00 0E 63 57 4E 20 53 65 74 49 50 20 C0 A8 01 01 49

4 Commands

4.1 Request for MDI

4.1.1 Command: Start to send MDI

Ask laser scanner to start to send MDI.

Supported Firmware

Prototype Number	P1 or higher

Request

Command Syntax: **cWN SendMDI**

Command part	Description	Type	Value
Command type	Request	String	cWN
Command	Start to send MDI	String	SendMDI

E.g :

cWN SendMDI	
ASCII	<STX>cWN{SPC}SendMDI<ETX>
HEX	02 63 57 4E 20 53 65 6E 64 4D 44 49 03
Binary	02 4C 45 55 5A 45 00 0B 63 57 4E 20 53 65 6E 64 4D 44 49 26

Response

Command Syntax: **cWA SendMDI**

Command part	Description	Type	Value
Command type	Response	String	cWA
Command	Start to send MDI	String	SendMDI

E.g :

cWA SendMDI	
ASCII	<STX>cWA{SPC}SendMDI<ETX>
HEX	02 63 57 41 20 53 65 6E 64 4D 44 49 03
Binary	02 4C 45 55 5A 45 00 0B 63 57 41 20 53 65 6E 64 4D 44 49 29

4.1.2 Command: Stop to send MDI

Ask laser scanner to stop to send MDI.

Supported Firmware

Prototype Number	P1 or higher

Request

Command Syntax: **cWN StopMDI**

Command part	Description	Type	Value
Command type	Request	String	cWN
Command	Stop to send MDI	String	StopMDI

E.g :

cWN StopMDI	
ASCII	<STX>cWN{SPC}StopMDI<ETX>
HEX	02 63 57 4E 20 53 74 6F 70 4D 44 49 03
Binary	02 4C 45 55 5A 45 00 0B 63 57 4E 20 53 74 6F 70 4D 44 49 22

Response

Command Syntax: **cWA StopMDI**

Command part	Description	Type	Value
Command type	Response	String	cWA
Command	Stop to send MDI	String	StopMDI

E.g :

cWA StopMDI	
ASCII	<STX>cWA{SPC}StopMDI<ETX>
HEX	02 63 57 41 20 53 74 6F 70 4D 44 49 03
Binary	02 4C 45 55 5A 45 00 0B 63 57 41 20 53 74 6F 70 4D 44 49 2D

4.2 Read out

4.2.1 Command: Get protocol

Query laser scanner for its MDI transmission protocol.

Supported Firmware

Prototype Number	P1 or higher

Request

Command Syntax: **cRN** GetProto

Command part	Description	Type	Value
Command type	Request	String	cRN
Command	Get protocol type	String	GetProto

E.g :

cRN GetProto	
ASCII	<STX>cRN{SPC}GetProto<ETX>
HEX	02 63 52 4E 20 47 65 74 50 72 6F 74 6F 03
Binary	02 4C 45 55 5A 45 00 0C 63 52 4E 20 47 65 74 50 72 6F 74 6F 5F

Response

Command Syntax: **cRA GetProto Data**

Command part	Description	Type	Value
Command type	Response	String	cRA
Command	Return protocol type	String	GetProto
Data	Protocol type	Enum8	0 – UDP 1 – TCP

E.g :

cRA GetProto 1	
ASCII	<STX>cRA{SPC}GetProto{SPC}1<ETX>
HEX	02 63 52 41 20 47 65 74 50 72 6F 74 6F 20 31 03
Binary	02 4C 45 55 5A 45 00 0E 63 52 41 20 47 65 74 50 72 6F 74 6F 20 01 71

4.2.2 Command: Get MDI data packet type

Query laser scanner for its MDI data packet type.

Supported Firmware

Prototype Number	P1 or higher

Request

Command Syntax: **cRN GetPType**

Command part	Description	Type	Value
Command type	Request	String	cRN
Command	Get MDI data packet type	String	GetPType

E.g :

cRN GetPType	
ASCII	<STX>cRN{SPC}GetPType<ETX>
HEX	02 63 52 4E 20 47 65 74 50 54 79 70 65 03
Binary	02 4C 45 55 5A 45 00 0C 63 52 4E 20 47 65 74 50 54 79 70 65 61

Response

Command Syntax: **cRA GetPType Data**

Command part	Description	Type	Value
Command type	Response	String	cRA
Command	Return MDI data packe type	String	GetPType
Data	MDI data packet type	Enum8	0 – Distance only 1 – Distance & intensity

E.g :

cRA GetPType 0	
ASCII	<STX>cRA{SPC}GetPType{SPC}0<ETX>
HEX	02 63 52 41 20 47 65 74 50 54 79 70 65 20 30 03
Binary	02 4C 45 55 5A 45 00 0E 63 52 41 20 47 65 74 50 54 79 70 65 20 00 4E

4.2.3 Command: Get angular resolution

Query laser scanner for its angular resolution.

Supported Firmware

Prototype Number	P1 or higher

Request

Command Syntax: **cRN GetResol**

Command part	Description	Type	Value
Command type	Request	String	cRN
Command	Get angular resolution	String	GetResol

E.g :

cRN GetResol	
ASCII	<STX>cRN{SPC}GetResol<ETX>
HEX	02 63 52 4E 20 47 65 74 52 65 73 6F 6C 03
Binary	02 4C 45 55 5A 45 00 0C 63 52 4E 20 47 65 74 52 65 73 6F 6C 4E

Response

Command Syntax: **cRA GetResol Data**

Command part	Description	Type	Value
Command type	Response	String	cRA
Command	Return angular resolution	String	GetResol
Data	Angular resolution	Enum8	ROD 300: 0 – 0.2°@80Hz 1 – 0.1°@40Hz 4 – 0.2°@50Hz ROD 500: 0 – 0.2°@80Hz 1 – 0.1°@40Hz 2 – 0.05°@20Hz 3 – 0.025°@10Hz 4 – 0.2° @ 50 Hz

E.g :

cRA GetResol 1	
ASCII	<STX>cRA{SPC}GetResol{SPC}1<ETX>
HEX	02 63 52 41 20 47 65 74 52 65 73 6F 6C 20 31 03
Binary	02 4C 45 55 5A 45 00 0E 63 52 41 20 47 65 74 52 65 73 6F 6C 20 01 60

4.2.4 Command: Get MDI data output direction

Query laser scanner for its MDI data output direction.

Supported Firmware

Prototype Number	P1 or higher

Request

Command Syntax: **cRN GetDir**

Command part	Description	Type	Value
Command type	Request	String	cRN
Command	Get MDI data output direction	String	GetDir

E.g :

cRN GetDir	
ASCII	<STX>cRN{SPC}GetDir<ETX>
HEX	02 63 52 4E 20 47 65 74 44 69 72 03
Binary	02 4C 45 55 5A 45 00 0A 63 52 4E 20 47 65 74 44 69 72 56

Response

Command Syntax: **cRA GetDir Data**

Command part	Description	Type	Value
Command type	Response	String	cRA
Command	Get MDI data output direction	String	GetDir
Data	MDI data output direction	Enum8	0 – Clockwise 1 – Counterclockwise

E.g :

cRA GetDir 0	
ASCII	<STX>cRA{SPC}GetDir{SPC}0<ETX>
HEX	02 63 52 41 20 47 65 74 44 69 72 20 30 03
Binary	02 4C 45 55 5A 45 00 0C 63 52 41 20 47 65 74 44 69 72 20 00 79

4.2.5 Command: Get angle range

Query laser scanner for its absolute angle range (unit : 0.01 degree) of one complete scan.

Supported Firmware

Prototype Number	P1 or higher

Request

Command Syntax: **cRN GetRange**

Command part	Description	Type	Value
Command type	Request	String	cRN
Command	Get angle range	String	GetRange

E.g :

cRN GetRange	
ASCII	<STX>cRN{SPC}GetRange<ETX>
HEX	02 63 52 4E 20 47 65 74 52 61 6E 67 65 03
Binary	02 4C 45 55 5A 45 00 0C 63 52 4E 20 47 65 74 52 61 6E 67 65 56

Response

Command Syntax: **cRA GetRange Start Stop**

Command part	Description	Type	Value
Command type	Response	String	cRA
Command	Return angle range	String	GetRange
Start	Start angle	Int16	-13760 – 13760
Stop	Stop angle	Int16	-13760 – 13760

E.g :

cRA GetRange -13760 13760	
ASCII	<STX>cRA{SPC}GetRange{SPC}-13760{SPC}13760<ETX>
HEX	02 63 52 41 20 47 65 74 52 61 6E 67 65 20 2D 34 37 35 30 20 32 32 37 35 30 03
Binary	02 4C 45 55 5A 45 00 11 63 52 41 20 47 65 74 52 61 6E 67 65 20 ED 72 58 DE 60

4.2.6 Command: Get skip spots

Query laser scanner for its skip spots between two successive output measurements.

Supported Firmware

Prototype Number	P1 or higher

Request

Command Syntax: **cRN GetSkip**

Command part	Description	Type	Value
Command type	Request	String	cRN
Command	Get skip spots number	String	GetSkip

E.g :

cRN GetSkip	
ASCII	<STX>cRN{SPC}GetSkip<ETX>
HEX	02 63 52 4E 20 47 65 74 53 6B 69 70 03
Binary	02 4C 45 55 5A 45 00 0B 63 52 4E 20 47 65 74 53 6B 69 70 28

Response

Command Syntax: **cRA GetSkip Data**

Command part	Description	Type	Value
Command type	Response	String	cRA
Command	Return skip spots number	String	GetSkip
Data	Skip spots number	Uint16	0 – max. Spot - 1

E.g :

cRA GetSkip 10	
ASCII	<STX>cRA{SPC}GetSkip{SPC}10<ETX>
HEX	02 63 52 41 20 47 65 74 53 6B 69 70 20 31 30 03
Binary	02 4C 45 55 5A 45 00 0E 63 52 41 20 47 65 74 53 6B 69 70 20 00 0A 0D

4.2.7 Command: Get contamination threshold

Query laser scanner for threshold percentage (%) values of contamination warning.

Supported Firmware

Prototype Number	P1 or higher

Request

Command Syntax: **cRN GetCont**

Command part	Description	Type	Value
Command type	Request	String	cRN
Command	Get contamination threshold	String	GetCont

E.g :

cRN GetCont	
ASCII	<STX>cRN{SPC}GetCont<ETX>
HEX	02 63 52 4E 20 47 65 74 43 6F 6E 74 03
Binary	02 4C 45 55 5A 45 00 0B 63 52 4E 20 47 65 74 43 6F 6E 74 3F

Response

Command Syntax: **cRA GetCont Warning1 Warning2**

Command part	Description	Type	Value
Command type	Response	String	cRA
Command	Return contamination threshold	String	GetCont
Warning1	Warning1 threshold	Uint8	0 – 100
Warning2	Warning2 threshold	Uint8	0 – 100

E.g :

cRA GetCont 20 40	
ASCII	<STX>cRA{SPC}GetCont{SPC}20{SPC}40<ETX>
HEX	02 63 52 41 20 47 65 74 43 6F 6E 74 20 32 30 20 34 30 03
Binary	02 4C 45 55 5A 45 00 0E 63 52 41 20 47 65 74 43 6F 6E 74 20 14 28 2C

4.2.8 Command: Get version

Query laser scanner for its version information.

Supported Firmware

Prototype Number	P1 or higher

Request

Command Syntax: cRN GetVer

Command part	Description	Type	Value
Command type	Request	String	cRN
Command	Get version information	String	GetVer

E.g :

cRN GetVer	
ASCII	<STX>cRN{SPC}GetVer<ETX>
HEX	02 63 52 4E 20 47 65 74 56 65 72 03
Binary	02 4C 45 55 5A 45 00 0A 63 52 4E 20 47 65 74 56 65 72 48

Response

Command Syntax: cRA GetVer PartNumber HWversion SWversion SWrevision Prototype CAN ProductId

Command part	Description	Type	Value
Command type	Response	String	cRA
Command	Return version information	String	GetVer
PartNumber	Product part number	Uint32	39000 – 39999
HWversion	Hardware version	Uint8	0 – 255
SWversion	Software version	Uint8	0 – 255
SWrevision	Software revision	Uint8	0 – 255
Prototype	Software prototype	Uint8	0 – 255
CAN	CAN number	Uint32	0 – 4294967295
ProductId	Product identification	Enum8	0 : Undefined 30: ROD 300 50: ROD 500

E.g :

cRA GetVer 39000 0 1 0 2 1234567 30	
ASCII	<STX>cRA{SPC}GetVer{SPC}39000{SPC}0{SPC}1{SPC}0{SPC}2{SPC}1234567{SPC}30<ETX>
HEX	02 63 52 41 20 47 65 74 56 65 72 20 33 39 30 30 30 20 30 20 31 20 30 20 31 32 33 34 35 36 37 20 33 30 03
Binary	02 4C 45 55 5A 45 00 18 63 52 41 20 47 65 74 56 65 72 20 00 00 98 58 00 01 00 02 00 12 D6 87 1E F9

4.2.9 Command: Get internal temperature

Query laser scanner for its internal temperature (unit : 0.01 ℃).

Supported Firmware

Prototype Number	P1 or higher

Request
Command Syntax: cRN GetTem

Command part	Description	Type	Value
Command type	Request	String	cRN
Command	Get internal temperature	String	GetTem

E.g :

cRN GetTem	
ASCII	<STX>cRN{SPC}GetTem<ETX>
HEX	02 63 52 4E 20 47 65 74 54 65 6D 03
Binary	02 4C 45 55 5A 45 00 0A 63 52 4E 20 47 65 74 54 65 6D 55

Response
Command Syntax: cRA GetTem Data

Command part	Description	Type	Value
Command type	Response	String	cRA
Command	Return internal temperature	String	GetTem
Data	Internal temperature	Int16	-5000 – 15000

E.g :

cRA GetTem -100	
ASCII	<STX>cRA{SPC}GetTem{SPC}-100<ETX>
HEX	02 63 52 41 20 47 65 74 54 65 6D 20 2D 31 30 30 03
Binary	02 4C 45 55 5A 45 00 0D 63 52 41 20 47 65 74 54 65 6D 20 FF 9C 19

4.2.10 Command: Get error log

Query laser scanner for its internal stored error log.

Supported Firmware

Prototype Number	P1 or higher

Request
Command Syntax: cRN GetELog

Command part	Description	Type	Value
Command type	Request	String	cRN
Command	Get error log	String	GetELog

E.g :

cRN GetELog	
ASCII	<STX>cRN{SPC}GetELog<ETX>
HEX	02 63 52 4E 20 47 65 74 45 4C 6F 67 03
Binary	02 4C 45 55 5A 45 00 0B 63 52 4E 20 47 65 74 45 4C 6F 67 08

Response

Command Syntax: cRA GetELog ErrorCount Error1 Date1 ... Error10 Date10

Command part	Description	Type	Value
Command type	Response	String	cRA
Command	Return error log	String	GetELog
ErrorCount	Number of error	Uint8	10 (Fixed)
Error1	The 1st error code	Uint16	0-65535
Date1	Date of the 1st error	Uint16	0-65535
...	...	Uint16	0-65535
...	...	Uint16	0-65535
Error10	The 10th error code	Uint16	0-65535
Date10	Date of the 10th error	Uint16	0-65535

E.g :

cRA GetELog 10 112 0 510 0 322 0 109 0 307 0 106 0 0 0 0 0 0 0 0 0	
ASCII	<STX>cRA{SPC}GetELog{SPC}10{SPC}112{SPC}0{SPC}510{SPC}0{SPC}322{SPC}0{SPC}109{SPC}0{SPC}307{SPC}0{SPC}106{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0{SPC}0<ETX>
HEX	02 63 52 41 20 47 65 74 45 4C 6F 67 20 31 30 20 31 31 32 20 30 20 35 31 30 20 30 20 33 32 32 20 30 20 31 30 39 20 30 20 33 30 37 20 30 20 31 30 36 20 30 20 30 20 30 20 30 20 30 20 30 20 30 20 30 20 30 20 30 20 30 20 30 20 03
Binary	02 4C 45 55 5A 45 00 35 63 52 41 20 47 65 74 45 4C 6F 67 20 0A 00 70 00 00 01 FE 00 00 01 42 00 00 00 6D 00 00 01 33 00 00 00 6A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 D4

All the supported error code is listed as below:

Error code	Description
101	Internal error
102	Internal error
103	Internal error
104	Internal error
105	Internal error
106	Internal error

Error code	Description
107	Internal error
108	Internal error
109	Internal error
110	Internal error (Config error)
112	Internal 5V error
114	Laser voltage error
115	APD voltage error
116	TDC error(parameter)
117	TDC error(refer distance)
118	TDC error(refer pw)
119	TDC error(calc distance)
120	TDC error(calc pw)
121	FB distance error
122	FB pw error
123	BUFOVR error
124	Motor error
125	Fork error
129	Vref error
130	Internal error (Task overflow)
131	Internal 3.3V error
132	APD1 error
133	APD2 error
134	Internal temperature error
135	Internal temperature error
136	VCOMP1 error
137	VCOMP2 error
138	Optical1 error
139	Optical2 error
140	Optical1 FB error
141	Optical2 FB error
142	WMS ref led error
143	WMS ref led hardware error
144	TDC error(Saturation)
145	TDC error(Dyntable)

Error code	Description
146	Shift register error
148	16V error
201	Power supply error
202	Internal temperature error
205	Internal temperature error
206	Internal temperature error
207	WMS ambient light error
208	Internal temperature error
307	Internal COM2 timing error
322	Internal COM1 error
323	Internal COM2 error
324	Internal SYNC timeout error
402	Masking error
405	Beam blocking error
510	Ethernet error
999	Unknown error

4.2.11 Command: Get LED control

This command is used to get the control status of the LEDs whether it should be on in normal operation state.

Supported Firmware

Prototype Number	P8 or higher

Request

Command Syntax: **cRN GetLED**

Command part	Description	Type	Value
Command type	Request	String	cRN
Command	Get control status of LED	String	GetLED

E.g :

cRN GetLED	
ASCII	<STX>cRN{SPC}GetLED<ETX>
HEX	02 63 52 4E 20 47 65 74 4C 45 44 03
Binary	02 4C 45 55 5A 45 00 0A 63 52 4E 20 47 65 74 4C 45 44 44

Response

Command Syntax: **cRA GetLED Data1 Data2**

Command part	Description	Type	Value
Command type	Response	String	cRA
Command	Return control status of LED	String	GetLED
Data1	Control status of status LEDs	Enum8	0 – Disabled 1 – Enabled
Data2	Reserved	Enum8	0 (Fixed)

E.g :

cRA GetLED 1 0	
ASCII	<STX>cRA{SPC}GetLED{SPC}1{SPC}0<ETX>
HEX	02 63 52 41 20 47 65 74 4C 45 44 20 30 03
Binary	02 4C 45 55 5A 45 00 0D 63 52 41 20 47 65 74 4C 45 44 20 01 00 6A

4.2.12 Command: Get lamp state

Query laser scanner for its lamp color state.

Supported Firmware

Prototype Number	P1 or higher

Request

Command Syntax: **cRN GetLamp**

Command part	Description	Type	Value
Command type	Request	String	cRN
Command	Get lamp color state	String	GetLamp

E.g :

cRN GetLamp	
ASCII	<STX>cRN{SPC}GetLamp<ETX>
HEX	02 63 52 4E 20 47 65 74 4C 61 6D 70 03
Binary	02 4C 45 55 5A 45 00 0B 63 52 4E 20 47 65 74 4C 61 6D 70 39

Response

Command Syntax: **cRA GetLamp Data1 Data2 Data3 Data4**

Command part	Description	Type	Value
Command type	Response	String	cRA
Command	Return lamp color state	String	GetLamp
Data1	LED1 state	Enum8	0 – Black 1 – Red 2 – Green 3 – Orange

Command part	Description	Type	Value
Data2	LED2 state	Enum8	0 – Black 1 – Red 2 – Green 3 – Orange
Data3	LED3 state	Enum8	0 – Black 1 – Red 2 – Green 3 – Orange
Data4	Reserved	Enum8	0 (Fixed)

E.g :

cRA GetLamp 2 1 1 0	
ASCII	<STX>cRA{SPC}GetLamp{SPC}2{SPC}1{SPC}1{SPC}0<ETX>
HEX	02 63 52 41 20 47 65 74 4C 61 6D 70 20 32 20 31 20 31 20 30 03
Binary	02 4C 45 55 5A 45 00 10 63 52 41 20 47 65 74 4C 61 6D 70 20 02 01 01 00 14

4.2.13 Command: Get ethernet configuration

Query laser scanner for its Ethernet configuration.

Supported Firmware

Prototype Number	P1 or higher

Request

Command Syntax: **cRN GetEthCfg**

Command part	Description	Type	Value
Command type	Request	String	cRN
Command	Get Ethernet configuration	String	GetEthCfg

E.g :

cRN GetEthCfg	
ASCII	<STX>cRN{SPC}GetEthCfg<ETX>
HEX	02 63 52 4E 20 47 65 74 45 74 68 43 66 67 03
Binary	02 4C 45 55 5A 45 00 0D 63 52 4E 20 47 65 74 45 74 68 43 66 67 12

Response

Command Syntax: **cRA GetEthCfg MAC IP SubnetMask Gateway Port**

Command part	Description	Type	Value
Command type	Response	String	cRA
Command	Return Ethernet configuration	String	GetEthCfg
MAC	MAC address	Uint8(6*)	0 – 255(6*)

Command part	Description	Type	Value
IP	IP	Uint8(4*)	0 – 255(4*)
SubnetMask	Subnet mask	Uint8(4*)	0 – 255(4*)
Gateway	Gateway	Uint8(4*)	0 – 255(4*)
Port	Port number	Uint16	1024 – 65535

E.g :

cRA GetEthCfg BE A0 BE A0 12 34 192 168 61 100 255 255 255 0 192 168 1 1 3050	
ASCII	<STX>cRA{SPC}GetEthCfg{SPC}BE{SPC}A0{SPC}BE{SPC}A0{SPC}12 {SPC}34{SPC}192{SPC}168{SPC}61{SPC}100{SPC}255{SPC}255{SPC} 255{ SPC}0{SPC}192{SPC}168{SPC}1{SPC}1{SPC}3050<ETX>
HEX	02 63 52 41 20 47 65 74 45 74 68 43 66 67 20 42 45 20 41 30 20 42 45 20 41 30 20 31 32 20 33 34 20 31 39 32 20 31 36 38 20 36 31 20 31 30 30 20 32 35 35 20 32 35 35 20 32 35 35 20 30 20 31 39 32 20 31 36 38 20 31 20 31 20 33 30 35 30 03
Binary	02 4C 45 55 5A 45 00 22 63 52 41 20 47 65 74 45 74 68 43 66 67 20 BE A0 BE A0 12 34 C0 A8 3D 64 FF FF FF 00 C0 A8 01 01 0B EA 5C

4.2.14 Command: Get runtime hours

Query laser scanner for its runtime hours.

Supported Firmware

Prototype Number	P1 or higher

Request

Command Syntax: **cRN GetHours**

Command part	Description	Type	Value
Command type	Request	String	cRN
Command	Get runtime hours	String	GetHours

E.g :

cRN GetHours	
ASCII	<STX>cRN{SPC}GetHours<ETX>
HEX	02 63 52 4E 20 47 65 74 48 6F 75 72 73 03
Binary	02 4C 45 55 5A 45 00 0C 63 52 4E 20 47 65 74 48 6F 75 72 73 5A

Response

Command Syntax: **cRA GetHours Data**

Command part	Description	Type	Value
Command type	Response	String	cRA
Command	Return runtime hours	String	GetHours
Data	Runtime hours	Uint32	0 – 4294967295

E.g :

cRA GetHours 100	
ASCII	<STX>cRA{SPC}GetHours{SPC}100<ETX>
HEX	02 63 52 41 20 47 65 74 48 6F 75 72 73 20 31 30 30 03
Binary	02 4C 45 55 5A 45 00 11 63 52 41 20 47 65 74 48 6F 75 72 73 20 00 00 00 64 11

4.2.15 Command: Get device name

Query laser scanner for the device name.
The maximum length of name is 20. Last space charactors and '\0' would be ignored.

Supported Firmware

Prototype Number	P1 or higher

Request
Command Syntax: **cRN GetName**

Command part	Description	Type	Value
Command type	Request	String	cRN
Command	Get device name	String	GetName

E.g :

cRN GetName	
ASCII	<STX>cRN{SPC}GetName<ETX>
HEX	02 63 52 4E 20 47 65 74 4E 61 6D 65 03
Binary	02 4C 45 55 5A 45 00 0B 63 52 4E 20 47 65 74 4E 61 6D 65 2E

Response
Command Syntax: **cRA GetName Data**

Command part	Description	Type	Value
Command type	Response	String	cRA
Command	Return device name	String	GetName
Data	Device name	String	stringName

E.g :

cRA GetName DeviceName	
ASCII	<STX>cRA{SPC}GetName{SPC}DeviceName<ETX>
HEX	02 63 52 41 20 47 65 74 4E 61 6D 65 20 44 65 76 69 63 65 4E 61 6D 65 03
Binary	02 4C 45 55 5A 45 00 16 63 52 41 20 47 65 74 4E 61 6D 65 20 44 65 76 69 63 65 4E 61 6D 65 1E

4.2.16 Command: Get window calibration status

Query laser scanner for window calibration status. Command [cWN SetWCalib] should be called prior to get window calibration status.

Supported Firmware

Prototype Number	P1 or higher

Request

Command Syntax: **cRN GetWCalib**

Command part	Description	Type	Value
Command type	Request	String	cRN
Command	Get window calibration status	String	GetWCalib

E.g :

cRN GetWCalib	
ASCII	<STX>cRN{SPC}GetWCalib<ETX>
HEX	02 63 52 4E 20 47 65 74 57 43 61 6C 69 62 03
Binary	02 4C 45 55 5A 45 00 0D 63 52 4E 20 47 65 74 57 43 61 6C 69 62 1B

Response

Command Syntax: **cRA GetWCalib Data**

Command part	Description	Type	Value
Command type	Response	String	cRA
Command	Return window calibration status	String	GetWCalib
Data	Window calibration status	Enum8	0 – Processing 1 – Done 3 – Failed

E.g :

cRA GetWCalib 0	
ASCII	<STX>cRA{SPC}GetWCalib{SPC}0<ETX>
HEX	02 63 52 41 20 47 65 74 57 43 61 6C 69 62 20 30 03
Binary	02 4C 45 55 5A 45 00 0F 63 52 41 20 47 65 74 57 43 61 6C 69 62 20 00 34

4.2.17 Command: Get filter type

Query laser scanner for the filter type.

The filter can be applied to spatial neighboring spots, historical spots or both. Depending on the type of filter, the max value of historical spots number and/or spatial neighboring spots number is different, shown as below:

- Median filter : historical spots number + spatial spots number should be less than or equal to 4
- Combo filter : historical spots number should be less than or equal to 4, AND
  historical spots number + spatial spots number should be less than or equal to 7
- Others : historical spots number + spatial spots number should be less than or equal to 7

Supported Firmware

Prototype Number	P1 or higher

Request

Command Syntax: **cRN GetFilter**

Command part	Description	Type	Value
Command type	Request	String	cRN
Command	Get filter type	String	GetFilter

E.g :

cRN GetFilter	
ASCII	<STX>cRN{SPC}GetFilter<ETX>
HEX	02 63 52 4e 20 47 65 74 46 69 6c 74 65 72 03
Binary	02 4C 45 55 5A 45 00 0D 63 52 4e 20 47 65 74 46 69 6c 74 65 72 29

Response

Command Syntax: **cRA GetFilter Type NumH NumS**

Command part	Description	Type	Value
Command type	Response	String	cRA
Command	Return filter type	String	GetFilter
Type	Fitler type	Enum8	0 – Median filter 1 – Average filter 2 – Max filter 3 – Combo filter
NumH[a)]	Historical spots number at current position	Uint8	0 – 7 or 0 – 4
NumS[a)]	Spatial neighboring spots number on both sides	Uint8	0 – 7 or 0 – 4

a) Note : Depending on the type of filter, the max value of NumH/NumS is different.

E.g :

cRA GetFilter 0 4 0	
ASCII	<STX>cRA{SPC}GetFilter{SPC}0{SPC}4{SPC}0<ETX>
HEX	02 63 52 41 20 47 65 74 46 69 6C 74 65 72 20 30 20 34 20 30 03
Binary	02 4C 45 55 5A 45 00 11 63 52 41 20 47 65 74 46 69 6C 74 65 72 20 00 04 00 02

4.2.18 Command: Get wms value

Query laser scanner for its detailed value in percentage (%) of all 132*2 sectors (Tx and Rx window) of window monitoring system (WMS). This command only valid in **BIN** mode.

Supported Firmware

Prototype Number	P1 or higher

Request

Command Syntax: **cRN GetWms**

Command part	Description	Type	Value
Command type	Request	String	cRN
Command	Get wms value	String	GetWms

E.g :

cRN GetWms	
ASCII	–
HEX	–
Binary	02 4C 45 55 5A 45 00 0A 63 52 4E 20 47 65 74 57 6D 73 40

Response

Command Syntax: **cRA GetWms D1...D264**

Command part	Description	Type	Value
Command type	Response	String	cRA
Command	Get wms value	String	GetWms
D1...D264	Wms value of all sectors	Uint8(264*)	0 – 100 (264*)

E.g :

cRA GetWms 0 0 0 0 ... 0	
ASCII	–
HEX	–
Binary	02 4C 45 55 5A 45 01 13 63 52 41 20 47 65 74 57 6D 73 20 00 00 00 00 ... 00 6F

4.2.19 Command: Get window contamination state

Query laser scanner for its contamination state (percentage value).

Sector	Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9
Angular	-137.5 – -105	-105 – -75	-75 – -45	-45 – -15	-15 – 15	15 – 45	45 – 75	75 – 105	105 – 137.5

*Sectors are not strictly divided by angle, and-there may be overlap between adjacent sectors

Supported Firmware

Prototype Number	P8 or higher

Request

Command Syntax: **cRN GetWinStat**

Command part	Description	Type	Value
Command type	Request	String	cRN
Command	Get contamination state	String	GetWinStat

E.g :

cRN GetECode	
ASCII	<STX>cRN{SPC}GetWinStat<ETX>
HEX	02 63 52 4E 20 47 65 74 57 69 6E 53 74 61 74 03
Binary	02 4C 45 55 5A 45 00 0E 63 52 4E 20 47 65 74 57 69 6E 53 74 61 74 6B

Response

Command Syntax: **cRA GetWinStat Z1 Z2 Z3 Z4 Z5 Z6 Z7 Z8 Z9**

Command part	Description	Type	Value
Command type	Response	String	cRA
Command	Return contamination state	String	GetWinStat
Z1	Contamination state of Z1	Uint8	0 – 100
Z2	Contamination state of Z2	Uint8	0 – 100
Z3	Contamination state of Z3	Uint8	0 – 100
Z4	Contamination state of Z4	Uint8	0 – 100
Z5	Contamination state of Z5	Uint8	0 – 100
Z6	Contamination state of Z6	Uint8	0 – 100
Z7	Contamination state of Z7	Uint8	0 – 100
Z8	Contamination state of Z8	Uint8	0 – 100
Z9	Contamination state of Z9	Uint8	0 – 100

E.g :

cRA GetWinStat 10 20 30 40 50 60 70 80 90	
ASCII	<STX>cRA{SPC}GetWinStat{{SPC}10{SPC}20{SPC}30{SPC}40{SPC}50 {SPC}60{SPC}70{SPC}80SPC}90<ETX>
HEX	02 63 52 41 20 47 65 74 57 69 6E 53 74 61 74 20 31 30 20 32 30 20 33 30 20 34 30 20 35 30 20 36 30 20 37 30 20 38 30 20 39 30 03
Binary	02 4C 45 55 5A 45 00 18 63 52 41 20 47 65 74 57 69 6E 53 74 61 74 20 0A 14 1E 28 32 3C 46 50 5A 2E

4.2.20 Command: Get error code

Query laser scanner for error code that are currently occurring (0 indicates there is no error occurring).

Supported Firmware

Prototype Number	P1 or higher

Request

Command Syntax: **cRN GetECode**

Command part	Description	Type	Value
Command type	Request	String	cRN
Command	Get currently occurring error code	String	GetECode

E.g :

cRN GetECode	
ASCII	<STX>cRN{SPC}GetECode<ETX>
HEX	02 63 52 4E 20 47 65 74 45 43 6F 64 65 03
Binary	02 4C 45 55 5A 45 00 0C 63 52 4E 20 47 65 74 45 43 6F 64 65 61

Response

Command Syntax: **cRA GetECode Data**

Command part	Description	Type	Value
Command type	Response	String	cRA
Command	Get current occurring error code	String	GetECode
Data	Error code	Uint16	0 – 999 (refer to 0)

E.g :

cRA GetECode 0	
ASCII	<STX>cRA{SPC}GetECode{SPC}0<ETX>
HEX	02 63 52 41 20 47 65 74 45 43 6F 64 65 20 30 03
Binary	02 4C 45 55 5A 45 00 0F 63 52 41 20 47 65 74 45 43 6F 64 65 20 00 00 4E

4.2.21 Command: Get MDI transmission status

Query laser scanner for current MDI transmission status.

Supported Firmware

Prototype Number	P1 or higher

Request

Command Syntax: **cRN GetTxMDI**

Command part	Description	Type	Value
Command type	Request	String	cRN
Command	Get MDI transmission status	String	GetTxMDI

E.g :

cRN GetTxMDI	
ASCII	<STX>cRN{SPC}GetTxMDI<ETX>
HEX	02 63 52 4E 20 47 65 74 54 78 4D 44 49 03
Binary	02 4C 45 55 5A 45 00 0C 63 52 4E 20 47 65 74 54 78 4D 44 49 65

Response

Command Syntax: **cRA GetTxMDI Data**

Command part	Description	Type	Value
Command type	Response	String	cRA
Command	Get current MDI transmission stattus	String	GetTxMDI
Data	current MDI transmission stattus	Uint8	0 : OFF 1 : ON

E.g :

cRA GetTxMDI 0	
ASCII	<STX>cRA{SPC}GetTxMDI{SPC}0<ETX>
HEX	02 63 52 41 20 47 65 74 54 78 4D 44 49 20 30 03
Binary	02 4C 45 55 5A 45 00 0E 63 52 41 20 47 65 74 54 78 4D 44 49 20 00 4A

4.2.22 Command: Get platform version

Query laser scanner for platform version.

Supported Firmware

Prototype Number	P8 or higher

Request

Command Syntax: **cRN GetPLVer**

Command part	Description	Type	Value
Command type	Request	String	cRN
Command	Get platform version	String	GetPLVer

E.g :

cRN GetPLVer	
ASCII	<STX>cRN{SPC}GetPLVer<ETX>
HEX	02 63 52 4E 20 47 65 74 50 4C 56 65 72 03
Binary	02 4C 45 55 5A 45 00 0C 63 52 4E 20 47 65 74 50 4C 56 65 72 54

Response

Command Syntax: **cRA GetPLVer Data**

Command part	Description	Type	Value
Command type	Response	String	cRA
Command	Get platform version	String	GetPLVer
Data	Platform version (=Data/100)	Uint16	100 – V1 200 – V2

E.g :

cRA GetPLVer 100	
ASCII	<STX>cRA{SPC}GetPLVer{SPC}100<ETX>
HEX	02 63 52 41 20 47 65 74 50 4C 56 65 72 20 31 30 30 03
Binary	02 4C 45 55 5A 45 00 0F 63 52 41 20 47 65 74 50 4C 56 65 72 20 00 64 1F

4.3 Configurate

Set parameter into laser scanner

4.3.1 Command: Set protocol

Set MDI transmission protocol of laser scanner.

Supported Firmware

Prototype Number	P1 or higher

Request Write

Command Syntax: **cWN SetProto Data**

Command part	Description	Type	Value
Command type	Request	String	cWN
Command	Set protocol	String	SetProto
Data	Protocol	Enum8	0 – UDP 1 – TCP

E.g :

cWN SetProto 1	
ASCII	<STX>cWN{SPC}SetProto{SPC}1<ETX>
HEX	02 63 57 4E 20 53 65 74 50 72 6F 74 6F 20 31 03
Binary	02 4C 45 55 5A 45 00 0E 63 57 4E 20 53 65 74 50 72 6F 74 6F 20 01 6F

Response

Command Syntax: **cWA SetProto Data**

Command part	Description	Type	Value
Command type	Response	String	cWA
Command	Set protocol	String	SetProto
Data	Protocol	Enum8	0 – UDP 1 – TCP

E.g :

cWA SetProto 1	
ASCII	<STX>cWA{SPC}SetProto{SPC}1<ETX>
HEX	02 63 57 41 20 53 65 74 50 72 6F 74 6F 20 31 03
Binary	02 4C 45 55 5A 45 00 0E 63 57 41 20 53 65 74 50 72 6F 74 6F 20 01 60

4.3.2 Command: Set MDI data packet type

Set MDI data packet type of laser scanner.

Supported Firmware

Prototype Number	P1 or higher

Request Write

Command Syntax: **cWN SetPType Data**

Command part	Description	Type	Value
Command type	Request	String	cWN
Command	Set MDI data packet type	String	SetPType
Data	MDI data packet type	Enum8	0 – Distance only 1 – Distance & Intensity

E.g :

cWN SetPType 1	
ASCII	<STX>cWN{SPC}SetPType{SPC}1<ETX>
HEX	02 63 57 4E 20 53 65 74 50 54 79 70 65 20 31 03
Binary	02 4C 45 55 5A 45 00 0E 63 57 4E 20 53 65 74 50 54 79 70 65 20 01 51

Response

Command Syntax: **cWA SetPType Data**

Command part	Description	Type	Value
Command type	Response	String	cWA
Command	Set MDI data packet type	String	SetPType
Data	MDI data packet type	Enum8	0 – Distance only 1 – Distance & Intensity

E.g :

cWA SetPType 1	
ASCII	<STX>cWA{SPC}SetPType{SPC}1<ETX>
HEX	02 63 57 41 20 53 65 74 50 54 79 70 65 20 31 03
Binary	02 4C 45 55 5A 45 00 0E 63 57 41 20 53 65 74 50 54 79 70 65 20 01 5E

4.3.3 Command: Set angular resolution

Set angular resolution of laser scanner.

Supported Firmware

Prototype Number	P1 or higher

Request Write

Command Syntax: **cWN SetResol Data**

Command part	Description	Type	Value
Command type	Request	String	cWN
Command	Set angular resolution	String	SetResol
Data	Angular resolution	Enum8	ROD 300: 0 – 0.2°@80Hz 1 – 0.1°@40Hz 4 – 0.2°@50Hz ROD 500: 0 – 0.2°@80Hz 1 – 0.1°@40Hz 2 – 0.05°@20Hz 3 – 0.025°@10Hz 4 – 0.2°@50Hz

E.g :

cWN SetResol 1	
ASCII	<STX>cWN{SPC}SetResol{SPC}1<ETX>
HEX	02 63 57 4E 20 53 65 74 52 65 73 6F 6C 20 31 03
Binary	02 4C 45 55 5A 45 00 0E 63 57 4E 20 53 65 74 52 65 73 6F 6C 20 01 7E

Response

Command Syntax: **cWA SetResol Data**

Command part	Description	Type	Value
Command type	Response	String	cWA
Command	Set angular resolution	String	SetResol
Data	Angular resolution	Enum8	ROD 300: 0 – 0.2°@80Hz 1 – 0.1°@40Hz 4 – 0.2°@50Hz ROD 500: 0 – 0.2°@80Hz 1 – 0.1°@40Hz 2 – 0.05°@20Hz 3 – 0.025°@10Hz 4 – 0.2°@50Hz

E.g :

cWA SetResol 1	
ASCII	<STX>cWA{SPC}SetResol{SPC}1<ETX>
HEX	02 63 57 41 20 53 65 74 52 65 73 6F 6C 20 31 03
Binary	02 4C 45 55 5A 45 00 0E 63 57 41 20 53 65 74 52 65 73 6F 6C 20 01 71

4.3.4 Command: Set MDI data output direction

Set MDI data output direction of laser scanner.

Supported Firmware

Prototype Number	P1 or higher

Request Write

Command Syntax: **cWN SetDir Data**

Command part	Description	Type	Value
Command type	Request	String	cWN
Command	Set MDI data output direction	String	SetDir
Data	MDI data output direction	Enum8	0 – Clockwise 1 – Counterclockwise

E.g :

cWN SetDir 0	
ASCII	<STX>cWN{SPC}SetDir{SPC}0<ETX>
HEX	02 63 57 4E 20 53 65 74 44 69 72 20 30 03
Binary	02 4C 45 55 5A 45 00 0C 63 57 4E 20 53 65 74 44 69 72 20 00 67

Response

Command Syntax: **cWA SetDir Data**

Command part	Description	Type	Value
Command type	Response	String	cWA
Command	Set MDI data output direction	String	SetDir
Data	MDI data output direction	Enum8	0 – Clockwise 1 – Counterclockwise

E.g :

cWA SetDir 0	
ASCII	<STX>cWA{SPC}SetDir{SPC}0<ETX>
HEX	02 63 57 41 20 53 65 74 44 69 72 20 30 03
Binary	02 4C 45 55 5A 45 00 0C 63 57 41 20 53 65 74 44 69 72 20 00 68

4.3.5 Command: Set angle range

Set absolute angle range (unit : 0.01 degree) of one complete scan of laser scanner.

Supported Firmware

Prototype Number	P1 or higher

Request Write

Command Syntax: **cWN SetRange Start Stop**

Command part	Description	Type	Value
Command type	Request	String	cWN
Command	Set angle range	String	SetRange
Start	Start angle	Int16	-13760 – 13760
Stop	Stop angle	Int16	-13760 – 13760

E.g :

cWN SetRange -13760 13760	
ASCII	<STX>cWN{SPC}SetRange{SPC}-4750{SPC}22750<ETX>
HEX	02 63 57 4E 20 53 65 74 52 61 6E 67 65 20 2D 34 37 35 30 20 32 32 37 35 30 03
Binary	02 4C 45 55 5A 45 00 11 63 57 4E 20 53 65 74 52 61 6E 67 65 20 ED 72 58 DE 7E

Response

Command Syntax: **cWA SetRange Start Stop**

Command part	Description	Type	Value
Command type	Response	String	cWA
Command	Set angle range	String	SetRange
Start	Start angle	Int16	-13760 – 13760
Stop	Stop angle	Int16	-13760 – 13760

E.g :

cWA SetRange -13760 13760	
ASCII	<STX>cWA{SPC}SetRange{SPC}-13760{SPC}13760<ETX>
HEX	02 63 57 41 20 53 65 74 52 61 6E 67 65 20 2D 31 33 37 36 30 20 31 33 37 36 30 03
Binary	02 4C 45 55 5A 45 00 11 63 57 41 20 53 65 74 52 61 6E 67 65 20 CA 40 35 C0 17

4.3.6 Command: Set skip spots

Set skip spots between two successive output measurements of laser scanner.

Supported Firmware

Prototype Number	P1 or higher

Request Write

Command Syntax: **cWN SetSkip Data**

Command part	Description	Type	Value
Command type	Request	String	cWN
Command	Set skip spots number	String	SetSkip
Data	Skip spots number	Uint16	0 – max. Spot - 1

E.g :

cWN SetSkip 0	
ASCII	<STX>cWN{SPC}SetSkip{SPC}0<ETX>
HEX	02 63 57 4E 20 53 65 74 53 6B 69 70 20 30 03
Binary	02 4C 45 55 5A 45 00 0E 63 57 4E 20 53 65 74 53 6B 69 70 20 00 00 19

Response

Command Syntax: **cWA SetSkip Data**

Command part	Description	Type	Value
Command type	Response	String	cWA
Command	Set skip spots number	String	SetSkip
Data	Skip spots number	Uint16	0 – max. Spot - 1

E.g :

cWA SetSkip 0	
ASCII	<STX>cWA{SPC}SetSkip{SPC}0<ETX>
HEX	02 63 57 41 20 53 65 74 53 6B 69 70 20 30 03
Binary	02 4C 45 55 5A 45 00 0E 63 57 41 20 53 65 74 53 6B 69 70 20 00 00 16

4.3.7 Command: Set contamination threshold

Set threshold percentage values of contamination warning of laser scanner.

Supported Firmware

Prototype Number	P1 or higher

Request Write

Command Syntax: **cWN SetCont Warning1 Warning2**

Command part	Description	Type	Value
Command type	Request	String	cWN
Command	Set contamination threshold	String	SetCont
Warning1	Warning1 threshold	Uint8	0 – 100
Warning2	Warning2 threshold (should bigger than warning)	Uint8	0 – 100

E.g :

cWN SetCont 20 40	
ASCII	<STX>cWN{SPC}SetCont{SPC}20{SPC}40<ETX>
HEX	02 63 57 4E 20 53 65 74 43 6F 6E 74 20 32 30 20 34 30 03
Binary	02 4C 45 55 5A 45 00 0E 63 57 4E 20 53 65 74 43 6F 6E 74 20 14 28 32

Response

Command Syntax: **cWA SetCont Warning1 Warning2**

Command part	Description	Type	Value
Command type	Response	String	cWA
Command	Set contamination threshold	String	SetCont
Warning1	Warning1 threshold	Uint8	0 – 100
Warning2	Warning2 threshold	Uint8	0 – 100

E.g :

cWA SetCont 20 40	
ASCII	<STX>cWA{SPC}SetCont{SPC}20{SPC}40<ETX>
HEX	02 63 57 41 20 53 65 74 43 6F 6E 74 20 32 30 20 34 30 03
Binary	02 4C 45 55 5A 45 00 0E 63 57 41 20 53 65 74 43 6F 6E 74 20 14 28 3D

4.3.8 Command: Set LED control

Set to control the status of LEDs whether it could be on in normal operation state. This would not block system error indication.

Supported Firmware

Prototype Number	P8 or higher

Request Write

Command Syntax: **cWN SetLED Data1 Data2**

Command part	Description	Type	Value
Command type	Request	String	cWN
Command	Set to control status of LED	String	SetLED
Data1	Control status of status LEDs	Enum8	0 – Disable 1 – Enable
Data2	Reserved	Enum8	0 (Fixed)

E.g :

cWN SetLED 0	
ASCII	<STX>cWN{SPC}SetLED{SPC}1{SPC}0<ETX>
HEX	02 63 57 4E 20 53 65 74 4C 45 44 20 31 20 30 03
Binary	02 4C 45 55 5A 45 00 0D 63 57 4E 20 53 65 74 4C 45 44 20 01 00 74

Response

Command Syntax: **cWA SetLED Data1 Data2**

Command part	Description	Type	Value
Command type	Response	String	cWA
Command	Set to control LED status	String	SetLED
Data1	Control status of status LEDs	Enum8	0 – Disable 1 – Enable
Data2	Reserved	Enum8	0 (Fixed)

E.g :

cWA SetLED 1	
ASCII	<STX>cWA{SPC}SetLED{SPC}1{SPC}0<ETX>
HEX	02 63 57 41 20 53 65 74 4C 45 44 20 31 20 30 03
Binary	02 4C 45 55 5A 45 00 0D 63 57 41 20 53 65 74 4C 45 44 20 01 00 7B

4.3.9 Command: Set ethernet configuration

Set Ethernet configuration of laser scanner.

Supported Firmware

Prototype Number	P1 or higher

Request Write

Command Syntax: **cWN SetEthCfg IP SubnetMask Gateway Port**

Command part	Description	Type	Value
Command type	Request	String	cWN
Command	Set Ethernet configuration	String	SetEthCfg
IP	IP	Uint8(4*)	0 – 255(4*)
SubnetMask	Subnet mask	Uint8(4*)	0 – 255(4*)
Gateway	Gateway	Uint8(4*)	0 – 255(4*)
Port	Port number	Uint16	1024 – 65535

E.g :

cWN SetEthCfg 192 168 1 2 255 255 255 0 192 168 1 1 3050	
ASCII	<STX>cWN{SPC}SetEthCfg{SPC}192{SPC}168{SPC}1{SPC}2{SPC}255{SPC}255{SPC}255{SPC}0{SPC}192{SPC}168{SPC}1{SPC}1{SPC}3050<ETX>
HEX	02 63 57 4E 20 53 65 74 45 74 68 43 66 67 20 31 39 32 20 31 36 38 20 31 20 32 20 32 35 35 20 32 35 35 20 32 35 35 20 30 20 31 39 32 20 31 36 38 20 31 20 31 20 33 30 35 30 03
Binary	02 4C 45 55 5A 45 00 1C 63 57 4E 20 53 65 74 45 74 68 43 66 67 20 C0 A8 01 02 FF FF FF 00 C0 A8 01 01 0B EA 3E

Response

Command Syntax: **cWA SetEthCfg IP SubnetMask Gateway Port**

Command part	Description	Type	Value
Command type	Response	String	cWA
Command	Set Ethernet configuration	String	SetEthCfg
IP	IP	Uint8(4*)	0 – 255(4*)
SubnetMask	Subnet mask	Uint8(4*)	0 – 255(4*)
Gateway	Gateway	Uint8(4*)	0 – 255(4*)
Port	Port number	Uint16	1024 – 65535

E.g :

cWA SetEthCfg 192 168 1 2 255 255 255 0 192 168 1 1 3050	
ASCII	<STX>cWA{SPC}SetEthCfg{SPC}192{SPC}168{SPC}1{SPC}2{SPC}255{SPC}255{SPC}255{SPC}0{SPC}192{SPC}168{SPC}1{SPC}1{SPC}3050<ETX>
HEX	02 63 57 41 20 53 65 74 45 74 68 43 66 67 20 31 39 32 20 31 36 38 20 31 20 32 20 32 35 35 20 32 35 35 20 32 35 35 20 30 20 31 39 32 20 31 36 38 20 31 20 31 20 33 30 35 30 03
Binary	02 4C 45 55 5A 45 00 1C 63 57 41 20 53 65 74 45 74 68 43 66 67 20 C0 A8 01 02 FF FF FF 00 C0 A8 01 01 0B EA 31

4.3.10 Command: Set device name

Set device name of laser scanner.
The maximum length of name is 20. If length of setting name is greater than 20, the over part would be cut off.

Supported Firmware

Prototype Number	P1 or higher

Request Write
Command Syntax: **cWN SetName Data**

Command part	Description	Type	Value
Command type	Request	String	cWN
Command	Set device name	String	SetName
Data	Device name	String	stringName

E.g :

cWN SetName myDevice	
ASCII	<STX>cWN{SPC}SetName{SPC}myDevice<ETX>
HEX	02 63 57 4E 20 53 65 74 4E 61 6D 65 20 6D 79 44 65 76 69 63 65 03
Binary	02 4C 45 55 5A 45 00 14 63 57 4E 20 53 65 74 4E 61 6D 65 20 6D 79 44 65 76 69 63 65 33

Response
Command Syntax: **cWA SetName Data**

Command part	Description	Type	Value
Command type	Response	String	cWA
Command	Set device name	String	SetName
Data	Device name	String	stringName

E.g :

cWA SetName myDevice	
ASCII	<STX>cWA{SPC}SetName{SPC}myDevice<ETX>
HEX	02 63 57 41 20 53 65 74 4E 61 6D 65 20 6D 79 44 65 76 69 63 65 03
Binary	02 4C 45 55 5A 45 00 14 63 57 41 20 53 65 74 4E 61 6D 65 20 6D 79 44 65 76 69 63 65 3C

4.3.11 Command: Set window calibration

Set to start window calibration.

Supported Firmware

Prototype Number	P1 or higher

Request Write

Command Syntax: **cWN SetWCalib Data**

Command part	Description	Type	Value
Command type	Request	String	cWN
Command	Set window calibration	String	SetWCalib
Data	Window calibration	Uint8	1 (Fixed)

E.g :

cWN SetWCalib 1	
ASCII	<STX>cWN{SPC}SetWCalib{SPC}1<ETX>
HEX	02 63 57 4E 20 53 65 74 57 43 61 6C 69 62 20 31 03
Binary	02 4C 45 55 5A 45 00 0F 63 57 4E 20 53 65 74 57 43 61 6C 69 62 20 01 2B

Response

Command Syntax: **cWA SetWCalib Data**

Command part	Description	Type	Value
Command type	Response	String	cWA
Command	Set window calibration	String	SetWCalib
Data	Window calibration	Uint8	1 (Fixed)

E.g :

cWA SetWCalib 1	
ASCII	<STX>cWA{SPC}SetWCalib{SPC}1<ETX>
HEX	02 63 57 41 20 53 65 74 57 43 61 6C 69 62 20 31 03
Binary	02 4C 45 55 5A 45 00 0F 63 57 41 20 53 65 74 57 43 61 6C 69 62 20 01 24

4.3.12 Command: Set filter type

Set the filter type of laser scanner.

The filter can be applied to spatial neighboring spots, historical spots or both. Depending on the type of filter, the max value of historical spots number and/or spatial neighboring spots number is different, shown as below:

- Median filter : historical spots number + spatial spots number should be less than or equal to 4
- Combo filter : historical spots number should be less than or equal to 4, AND
  historical spots number + spatial spots number should be less than or equal to 7
- Ohters : historical spots number + spatial spots number should be less than or equal to 7

Supported Firmware

Prototype Number	P1 or higher

Request Write

Command Syntax: **cWN SetFilter Type NumH NumS**

Command part	Description	Type	Value
Command type	Request	String	cWN
Command	Set filter type	String	SetFilter
Type	Filter type	Enum8	0 – Median filter 1 – Average filter 2 – Max filter 3 – Combo filter
NumH[a)]	Historical spots number at current position	Uint8	0 – 7 or 0 – 4
NumS[a)]	Spatial neighboring spots number on both sides	Uint8	0 – 7 or 0 – 4

a) Note : Depending on the type of filter, the max value of NumH/NumS is different.

E.g :

cWN SetFilter 0 4 0	
ASCII	<STX>cWN{SPC}SetFilter{SPC}0{SPC}4{SPC}0<ETX>
HEX	02 63 57 4E 20 53 65 74 46 69 6C 74 65 72 20 30 20 34 20 30 03
Binary	02 4C 45 55 5A 45 00 11 63 57 4E 20 53 65 74 46 69 6C 74 65 72 20 00 04 00 1C

Response

Command Syntax: **cWA SetFilter Type NumH NumS**

Command part	Description	Type	Value
Command type	Response	String	cWA
Command	Set filter type	String	SetFilter
Type	Filter type	Enum8	0 – Median filter 1 – Average filter 2 – Max filter 3 – Combo filter
NumH[a)]	Historical spots number at current position	Uint8	0 – 7 or 0 – 4
NumS[a)]	Spatial neighboring spots number on both sides	Uint8	0 – 7 or 0 – 4

a) Note : Depending on the type of filter, the max value of NumH/NumS is different.

E.g :

cWA SetFilter 0 4 0	
ASCII	<STX>cWA{SPC}SetFilter{SPC}0{SPC}4{SPC}0<ETX>
HEX	02 63 57 41 20 53 65 74 46 69 6C 74 65 72 20 30 20 34 20 30 03
Binary	02 4C 45 55 5A 45 00 11 63 57 41 20 53 65 74 46 69 6C 74 65 72 20 00 04 00 13

4.3.13 Command: Reset

Set all user parameters to default.

Supported Firmware

Prototype Number	P1 or higher

Request Write

Command Syntax: **cWN Reset**

Command part	Description	Type	Value
Command type	Request	String	cWN
Command	Reset all parameters	String	Reset

E.g :

cWN Reset	
ASCII	<STX>cWN{SPC}Reset<ETX>
HEX	02 63 57 4E 20 52 65 73 65 74 03
Binary	02 4C 45 55 5A 45 00 09 63 57 4E 20 52 65 73 65 74 0F

Response

Command Syntax: **cWA Reset**

Command part	Description	Type	Value
Command type	Response	String	cWA
Command	Reset all parameters	String	Reset

E.g :

cWA Reset	
ASCII	<STX>cWA{SPC}Reset<ETX>
HEX	02 63 57 41 20 52 65 73 65 74 03
Binary	02 4C 45 55 5A 45 00 09 63 57 41 20 52 65 73 65 74 00

4.3.14 Command: Reboot device

Reboot laser scanner.
There is no response to this command.

Supported Firmware

Prototype Number	P1 or higher

Request Write

Command Syntax: **cWN Reboot**

Command part	Description	Type	Value
Command type	Request	String	cWN
Command	Reboot device	String	Reboot

E.g :

cWN Reboot	
ASCII	<STX>cWN{SPC}Reboot<ETX>
HEX	02 63 57 4E 20 52 65 62 6F 6F 74 03
Binary	02 4C 45 55 5A 45 00 0A 63 57 4E 20 52 65 62 6F 6F 74 7B

4.4 MDI packet

Supported Firmware

Prototype Number	P1 or higher

The format of the MDI data packet is shown as below :

Header	Msg	Footer

Header :

SYNC	Packet type	Packet size	Re-served	Re-served	Re-served	Packet NO.	Total NO.	Sub NO.	Scan freq	Scan spots	First angle	Delta angle	Time-stamp
4 Bytes	1Bytes	2Bytes	2Bytes	2Bytes	2Bytes	2Bytes	1Byte	1Byte	2Bytes	2Bytes	4Bytes	4Bytes	2 Bytes

Msg :

Distance	Intensity
Max 700*2Bytes	Max 700*2Bytes

Footer :

CRC
2Bytes

Header:

- *SYNC* : Synchronization pattern : 4C 45 55 5A;
- *Packet type* : Type of scan data packet, 0 = distance only, 1= distance & intensity ;
- *Packet size* : Overall size of this packet in bytes, the max is 1433;
- *Reserved* : Reserved for future use ;
- *Reserved* : Reserved for future use ;
- *Reserved* : Reserved for future use ;
- *Packet NO.* : Sequence number for pac (ket, start counting from sensor startup ;
- *Total NO.* : Totoal frame number of a whole scan ;
- *Sub NO.* : The index of this packet in a whole scan (starting from 1, not greater than Total NO.) ;
- *Scan freq* : Scan frequency (Hz) ;
- *Scan spots* : The number of scan spots within this packet ;
- *First angle* : Absolute angle of the first spot in this packet (1/1000°) ;
- *Delta angle* : Delta angle between two consecutive output spots (1/1000°) ;
- *Timestamp* : Timestamp of the packet. unit is ms ;

Msg

- *Distance* : Measured distance in mm, invalid distance reture max.
  The maximum number of distance is 700 ;
- *Intensity* : Measured intensity. (if intensity output is asked),
  The maximum number of intensity is 700 ;

NOTE
The maximum size of Msg should not be greater than 1400. If the Msg contains both distance and intensity, the maximum number of distance is 350, and the maximum number of intensity is 350. If the Msg contains distance only, the maximum number of distance could reach 700.

Footer

- *CRC* : CRC16 check of the packet, including Header and Msg ;
- **Max length of MDI** : Header + Msg + Footer = 31 + 700*2 +2 = 1433 Bytes

Example

4C 45 55 5A 01 00 35 00 00 00 00 00 00 00 01 05 01 00 50 00 05 FF FF CF 90 00 00 4E 20 00 1A 01 55 01 50 01 00 02 00 01 22 00 60 00 55 01 00 00 20 00 60 DD 2F

4C 45 55 5A:	**SYNC**	
01 :	**Packet type**	Distance & intensity
00 35 :	**Packet size**	Total size of the frame. 53 bytes in this frame
00 00 :	**Reserved**	Reserved
00 00 :	**Reserved**	Reserved
00 00 :	**Reserved**	Reserved
00 01 :	**Packet NO.**	This is the first sequence of the packet
05 :	**Total NO.**	The whole scan has 5 packets
01 :	**Sub NO.**	This is the first packet of the whole 5 scan
00 50 :	**Scan freq**	The scan frequency is 80Hz
00 05 :	**Scan spots**	This packet includes 5 spots information
FF FF CF 90 :	**First angle**	First angle of this packet. This first angle is -12400/1000 = -12.4°
00 00 4E 20 :	**Delta angle**	Delta angle is 20000/1000=20.0°
00 1A :	**Timestamp**	This is the timestamp of this packet Unit is ms. So the time is 26ms
01 55 :	**1st spot distance**	This distance is 341mm
01 50 :	**2nd spot distance**	This distance is 336mm
01 00 :	**3rd spot distance**	This distance is 256mm
02 00 :	**4th spot distance**	This distance is 512mm
01 22 :	**5th spot distance**	This distance is 290mm
00 60 :	**1st spot intensity**	This intensity is 96
00 55 :	**2nd spot intensity**	This intensity is 85
01 00 :	**3rd spot intensity**	This intensity is 256
00 20 :	**4th spot intensity**	This intensity is 32
00 60 :	**5th spot intensity**	This intensity is 96
CB 76 : CRC		**CRC16**

From the frame message, the angle of each spot can be calculated as below :

1st : -12.4°(First angle) + 20.0°(Delta angle) * (1 - 1) = -12.4°
2nd : -12.4°(First angle) + 20.0°(Delta angle) * (2 - 1) = 7.6°
3rd : -12.4°(First angle) + 20.0°(Delta angle) * (3 - 1) = 27.6°
4th : -12.4°(First angle) + 20.0°(Delta angle) * (4 - 1) = 47.6°
5th : -12.4°(First angle) + 20.0°(Delta angle) * (5 - 1) = 67.6°

CRC code

The verification field of the frame is a CRC16. The properties of the CRC processing are presented below: (All bytes are considered for computing the CRC)

The polynomial is 0x90d9

Preset value is 0

A source code for computing the CRC is proposed below:

```
#define BYTE unsigned char
#define WORD unsigned short
#define POLYNOM 0x90d9
WORD CRC16(BYTE *buf_, WORD cnt_)
{
	WORD crc = 0; /* CRC value is 16bit */
	WORD i, j;
	for (i = 0; i < cnt_; i++)
	{
		crc ^= (WORD)(buf_[i] << 8); /* move byte into MSB of 16bit CRC */
		for (j = 0; j < 8; j++)
		{
			if ((crc & 0x8000) != 0) /* test for MSB = bit 15 */
			{
				crc = (WORD)((crc << 1) ^ POLYNOM);
			}
			else
			{
				crc <<= 1;
			}
		}
	}
	return crc;
}
```