“WAVEGUIDE-172-E4-CONTROL” Protocol Description

Contains

[1. Interface connection 2](#_Toc206078232)

[2. RS232/USB COM port parameters 2](#_Toc206078233)

[3. Protocol description 2](#_Toc206078234)

[3.1. Command format 2](#_Toc206078235)

[4. Common commands 3](#_Toc206078236)

[4.1. Console mode switch request 3](#_Toc206078237)

[4.2. Get protocol version 3](#_Toc206078238)

[4.3. Get hash summ 3](#_Toc206078239)

[4.4. Set new IP address 4](#_Toc206078240)

[4.5. Get current IP address 4](#_Toc206078241)

[4.6. Set new MAC address 4](#_Toc206078242)

[4.7. Get current MAC address 4](#_Toc206078243)

[4.8. Factory reset request 5](#_Toc206078244)

[5. Command set 6](#_Toc206078245)

[5.1. Request to get status 6](#_Toc206078246)

[5.2. Request to set relay state 6](#_Toc206078247)

[5.3. Request to set lamp state 7](#_Toc206078248)

[5.4. Request to set laser control type 7](#_Toc206078249)

[5.5. Request to set laser signal tower state 7](#_Toc206078250)

[5.6. Request to set protection state 8](#_Toc206078251)

[5.7. Request to set out state 8](#_Toc206078252)

# Interface connection

Supported connection types: RS232, USB. Protocol provides simple text commands and responses.

# RS232/USB COM port parameters

* Baud rate: 115200 baud/sec
* Data bits: 8bit
* Parity: no parity
* Stop bits: 1 bit

# Protocol description

Protocol version name is “WAVEGUIDE-172-E4-CONTROL-ДДММГГ”

# Command format

All commands and responses is a simple text string that has the following structure:

<message\_header> <sub\_sys\_type> <id> <param1> <param2> <…> <CR>

Where:

* <message\_header> - message header that contains command keyword
* <sub\_sys\_type> - target subsystem addressed by the command
* <id> - subsys address

NOTE: For this laser configuration, the <id> is always 0

* <paramN> - a bunch of parameters corresponding to command
* <CR> - carriage return symbol

# Common commands

Protocol version name: “WAVEGUIDE-172-E4-CONTROL-ДДММГГ”

## Console mode switch request

“console\r”

Response:

“lConsole mode <value>\r”

Where:

* <value> - result (string), ON, OFF

## Get protocol version

“lgvers\r”

Response:

“lrvers <protocol version name>\r”

Where:

* <protocol version name> - protocol version name (See. Chapter 1)

## Get hash summ

“lghash\r”

Response:

“lrhash <hash information>\r”

Where:

* <hash information> - hash information string

## Set new IP address

“lsip <xxx.xxx.xxx.xxx>\r”

Where:

* <xxx.xxx.xxx.xxx> - current IP address, (192.168.26.220 default value)

Response:

“lrip <xxx.xxx.xxx.xxx>\r”

ERROR response:

“lrip <xxx.xxx.xxx.xxx> ERR\r”

Where:

* <xxx.xxx.xxx.xxx> - current IP address, (192.168.26.220 default value)

## Get current IP address

“lgip\r”

Response:

“lrip <xxx.xxx.xxx.xxx>\r”

Where:

* <xxx.xxx.xxx.xxx> - current IP address, (192.168.26.220 default value)

## Set new MAC address

“lsmac <xx xx xx xx xx xx>\r”

Where:

* < xx xx xx xx xx xx > - new MAC address

Response:

“lrmac <xx xx xx xx xx xx>\r”

ERROR response:

“lrmac <xx xx xx xx xx xx> ERR\r

## Get current MAC address

“lgmac\r”

Response:

“lrmac <xx xx xx xx xx xx>\r”

Where:

* < xx xx xx xx xx xx > - current MAC address

## Factory reset request

“lsfactrst\r”

Response:

“lrfactrst OK\r”

ERROR response:

“lrfactrst ERR\r”

# Command set

## Request to get status

“lgstatus comm <id>\r”

Response:

“lrstatus comm <id> <key> <emergency> <door1> <door2> <door3> <protection> <relay1> <relay2> <relay3> <relay4> <relay5> <lamp\_fl> <lamp\_bl> <lamp\_pl> <laser\_control> <button\_fl> <button\_bl> <button\_pl> <out1> <out2>\r”

ERROR Response:

“lrstatus comm <id> ERR\r”

Where:

* <id> - device id (int), 0 if device is single
* <key> - key value (int), 0 – off, 1 - triggered
* <emergency> - emergency value (int), 0 – off, 1 – triggered
* <doorN> - N door state (int), 0 – off, 1 – triggered
* <protection> - protection value (int) 0 – off, 1 - on
* <relayN> - relay state (int) 0 – off, 1 – on
* <lamp\_Xl> - lamp X state (int), 0 – off, 1 – on
* <laser\_control> - laser control type, (int) 0 – scan head, 1 – acs LCI
* <button\_Xl> - button X state (int), 0 – not pressed, 1 – pressed
* <outN> - N out state (int), 0 – off, 1 - on

## Request to set relay state

“lsrelay comm <id> <index> <value>\r”

Response:

“lrrelay comm <id> <index> <value>\r”

ERROR Response:

“lrrelay comm <id> ERR\r”

Where:

* <id> - device id (int), 0 if device is single
* <index> - relay index (int) 0 = relay1, 1 = relay2 …
* <value> - relay state, (int) 0 – off, 1 – on

## Request to set lamp state

“lslamp comm <id> <index> <value>\r”

Response:

“lrlamp comm <id> <index> <value>\r”

ERROR Response:

“lrlamp comm <id> ERR\r”

Where:

* <id> - device id (int), 0 if device is single
* <index> - lamp index (int) 0 = lamp1, 1 = lamp2 …
* <value> - lamp state, (int) 0 – off, 1 – on

## Request to set laser control type

“lslaser comm <id> <value>\r”

Response:

“lrlaser comm <id> <value>\r”

ERROR Response:

“lrlaser comm <id> ERR\r”

Where:

* <id> - device id (int), 0 if device is single
* <value> - laser control type, (int) 0 – scan head, 1 – acs LCI

## Request to set laser signal tower state

“lslasstat comm <id> <led\_power> <led\_emission> <led\_error>\r”

Response:

“lrlasstat comm <id> <led\_power> <led\_emission> <led\_error>\r”

ERROR Response:

“lrlasstat comm <id> ERR\r”

Where:

* <id> - device id (int), 0 if device is single
* <led\_power> - GREEN led power state, (int) 0 – off, 1 – on
* <led\_emission> - ORANGE led emission state, (int) 0 – off, 1 – on
* <led\_error> - RED led error state, (int) 0 – off, 1 – on

## Request to set protection state

“lsprot comm <id> <value>\r”

Response:

“lrprot comm <id> <value>\r”

ERROR Response:

“lrprot comm <id> ERR\r”

Where:

* <id> - device id (int), 0 if device is single
* <value> - protection state, (int) 0 – off, 1 – on

## Request to set out state

“lsout comm <id> <out1> <out2>\r”

Response:

“lrout comm <id> <out1> <out2>\r”

ERROR Response:

“lrout comm <id> ERR\r”

Where:

* <id> - device id (int), 0 if device is single
* <out1> - out1 state, (int) 0 – off, 1 – on
* <out2> - out2 state, (int) 0 – off, 1 – on