DIGITAL PROJECTION

E-Vision Laser 7500, 8500, 10K & 4K Series
HIGHlite Laser 3D II & 4K Series
M-Vision Laser 18K Series
E-Vision 6900 Series
Mercury Quad Series
INSIGHT Dual Laser 4K Series
INSIGHT 4K Quad & Dual LED Series
INSIGHT 4K Laser Series

PROTOCOL GUIDES

Rev H October 2017
115-482H

About This Document

Follow the instructions in this guide carefully to ensure safe and long-lasting use of the projector.

Symbols used in this guide

Notes

Many pages in this document have a dedicated area for notes. The information in that area is accompanied by the following symbol:

☞ *NOTE: this symbol indicates that there is some important information that you should read.*

Product badges

Product badges are sometimes used to identify information that only applies to specific projectors as opposed to all projectors covered within the section.

Product revision

Because we at Digital Projection continually strive to improve our products, we may change specifications and designs, and add new features without prior notice.

Legal notice

Notes

CONTENTS

CONTENTS (continued)

CONTENTS (continued)

This page is intentionally left blank.

E-Vision Laser 7500 & 8500 Series
E-Vision Laser 10K Series
E-Vision Laser 4K Series

High Brightness Digital Video Projector

PROTOCOL GUIDE

Rev H October 2017

IN THIS GUIDE

Introduction

The projector can be controlled by using an external control system or a PC via an RS232 or LAN interface, using a terminal-emulation program.

Network setup

1. Connect the projector to a LAN network.
2. Open the **Setup > Network** menu and edit network settings. The default IP address is **192.168.0.100** and the TCP port number is **7000**.

Serial Port setup

- Baud rate 9,600 bps
- Data length 8 bits
- Stop bits one
- Parity none
- Flow control none

Notes

For details on connecting the projector to an RS232 or LAN network, or changing network settings, see the user manual.

Only one control path at a time should be used for protocol control. Attempts to send commands to both serial and network ports at the same time may result in unpredictable behavior.

Protocol commands

Commands are used to simulate menu operations and determine the settings of the projector, and use the following format:

- All commands consist of ASCII text strings starting with an asterisk∗ and ending with an ASCII Carriage Return character↵ (code 13): **`*command operator <value>`**↵
- The **`<command>`** string determines which setting the command will affect.
- Spaces are required before the operator and before the value.
- The **`<operator>`** string can take one of the following formats:

Command type	<operator>	Description
Set	`= <value>`	Makes the setting take the `<value>`.
Get	?	Asks what the current value is. The value is returned as an ASCII text string.
Execute		Performs an action. No operator is entered for this type of command.

Examples

`*orientation = 3`↵	sets the orientation to Rear Ceiling (for a ceiling mounted projector positioned behind the screen)
`*aspect.ratio ?`↵	asks what the current aspect ratio is
`*zoom.in`↵	commands the projector to zoom in
`*orientation=3`↵	is an invalid instruction because of the missing spaces before the operator and the value

Responses

If the command has been successful, the projector response begins with `ACK` or `ack` (“acknowledged”). For example, if the command is `*aspect.ratio = 1`↵, the projector will return `ACK aspect.ratio = 1`↵ or `ack aspect.ratio = 1`↵, depending on the model. In either case the projector will then will change the aspect ratio accordingly.

If the command has not been acknowledged, due to a syntax error or another problem, the projector response will be `NAK` or `nack`, followed by a brief description of the problem.

Notes

To set the default value of a command, simply enter the command name and ↵, without an operator. For example `*orientation`↵ *will set the orientation to 0 (Desktop Front).*

You must wait for the complete response to a command before sending another command.

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision Laser 7500 & 8500	E-Vision Laser 10K	E-Vision Laser 4K
Input								
`input`	●	●	✕	✕	✕	0 = HDMI 1 1 = HDMI 2 2 = VGA 3 = COMP 4 = DVI 5 = DisplayPort 6 = HDBaseT 7 = 3G-SDI		0 = DisplayPort 1 = HDMI 1 2 = HDMI 2 3 = HDBaseT 4 = 3G-SDI 5 = HDMI 3 6 = HDMI 4
Test Pattern								
`test.pattern`	●	●	✕	✕	✕	0= Off 1 = White 2 = Black 3 = Red 4 = Green 5 = Blue 6 = Checkerboard 7 = Crosshatch 8 = V Burst 9 = H Burst 10 = Color Bar 11 = Plunge		0 = Off 1 = White 2 = Black 3 = Red 4 = Green 5 = Blue 6 = Cyan 7 = Yellow 8 = Magenta

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision Laser 7500 & 8500	E-Vision Laser 10K	E-Vision Laser 4K
Lens								
`zoom.in`	✕	✕	✕	✕	●	✓	✓	✓
`zoom.out`	✕	✕	✕	✕	●	✓	✓	✓
`focus.near`	✕	✕	✕	✕	●	✓	✓	✓
`focus.far`	✕	✕	✕	✕	●	✓	✓	✓
`lens.up`	✕	✕	✕	✕	●	✓	✓	✓
`lens.down`	✕	✕	✕	✕	●	✓	✓	✓
`lens.left`	✕	✕	✕	✕	●	✓	✓	✓
`lens.right`	✕	✕	✕	✕	●	✓	✓	✓
`lens.center`	✕	✕	✕	✕	●	✓	✓	✓
`lens.load`	●	✕	✕	✕	✕	1 to 10 (integer)		
`lens.save`	●	●	✕	✕	✕	1 to 10 (integer)		
`lens.clear`	●	✕	✕	✕	✕	1 to 10 (integer)		
`lens.type`	●	●	✕	✕	✕	0 = non-UST Lens 1 = UST Lens		
`lens.lock`	●	●	✕	✕	✕	0 = Off 1 = On		

Continues on next page...

Notes

Lens commands only work if the projector is switched on.

To use lens commands, make sure the lens is unlocked. If `lens.lock` *is set to 1, most other lens commands are disabled.*

Exceptions are `lens.type`, `lens.save` *and* `lens.clear`.

When used with a get operator, the `lens.save` *command returns a string of zeroes and ones where each zero is an empty memory slot and each one is an occupied slot.*

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision Laser 7500 & 8500	E-Vision Laser 10K	E-Vision Laser 4K
Image								
pic.mode	●	●	✕	✕	✕	0 = High Bright 1 = Presentation 2 = Video		
db.on	●	●	✕	✕	✕	0 = Off 1 = On		
gamma	●	●	✕	✕	✕	0 = 1.0 1 = 1.8 2 = 2.0 3 = 2.2 4 = 2.35 5 = 2.5 6 = S-curve 7 = DICOM		
brightness	●	●	●	●	✕	0 to 200 (integer)		
contrast	●	●	●	●	✕	0 to 200 (integer)		
saturation	●	●	●	●	✕	0 to 200 (integer)		
hue	●	●	●	●	✕	0 to 200 (integer)		
sharpness	●	●	●	●	✕	0 to 15 (integer)		0 to 20 (integer)

Continues on next page...

Notes

☞ *The values you set of* `pic.mode`, `gamma`, `brightness`, `contrast`, `saturation` *and* `hue` *will only apply to the current image source.*

☞ `db.on` *is not available in 3D.*

☞ `db.on` *cannot be used when edge blending.*

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision Laser 7500 & 8500	E-Vision Laser 10K	E-Vision Laser 4K
nr.level	●	●	✕	✕	✕	n/a		0 to 3 (integer)
nr.temporal	●	●	●	●	✕	0 to 3 (integer)		n/a
nr.block	●	●	●	●	✕	0 to 3 (integer)		n/a
nr.mosquito	●	●	●	●	✕	0 to 3 (integer)		n/a
nr.hori	●	●	●	●	✕	0 to 3 (integer)		n/a
nr.vert	●	●	●	●	✕	0 to 3 (integer)		n/a
nr.reset	●	●	✕	✕	✕	0 to 3 (integer)		n/a
h.position	●	●	●	●	✕	0 to 200 (integer)		n/a
v.position	●	●	●	●	✕	0 to 200 (integer)		n/a
vga.phase	●	●	●	●	✕	0 to 31 (integer)		n/a
tracking	●	●	●	●	✕	0 to 200 (integer)		n/a
sync.level	●	●	●	●	✕	0 to 200 (integer)		n/a
freeze	●	●	✕	✕	✕	0 = Off 1 = On		
resync	✕	✕	✕	✕	●	✓	✓	✓

Continues on next page...

Notes

The commands on this page will only apply to the current image source.

The `vga.phase` *command is identical to the* **Phase** *setting in the* **Image > Position and Phase** *menu.*

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision Laser 7500 & 8500	E-Vision Laser 10K	E-Vision Laser 4K
Color								
color.space	●	●	✕	✕	✕	0 = Auto 1 = YPbPr 2 = YCbCr 3 = RGB-PC 4 = RGB-Video		
color.temp	●	●	✕	✕	✕	0 = 3200K 1 = 5400K 2 = 6500K 3 = 7500K 4 = 9300K 5 = Native		
color.mode	●	●	✕	✕	✕	0 = ColorMax 1 = Manual Color Matching 2 = Color Temperature 3 = Gains and Lifts		
color.max	●	●	✕	✕	✕	n/a		0 = HDTV (REC709) 1 = Peak 2 = User 1 3 = User 2

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision Laser 7500 & 8500	E-Vision Laser 10K	E-Vision Laser 4K
red.lift	●	●	●	●	✕	0 to 200 (integer)		
green.lift	●	●	●	●	✕	0 to 200 (integer)		
blue.lift	●	●	●	●	✕	0 to 200 (integer)		
red.gain	●	●	●	●	✕	0 to 200 (integer)		
green.gain	●	●	●	●	✕	0 to 200 (integer)		
blue.gain	●	●	●	●	✕	0 to 200 (integer)		
gainlift.reset	✕	✕	✕	✕	●	✓	✓	n/a
auto.test.ptrn	●	●	✕	✕	✕	0 = Off 1 = On		
user.std.rx	●	●	✕	✕	✕	550 to 750 (integer)		
user.std.ry	●	●	✕	✕	✕	250 to 450 (integer)		
user.std.gx	●	●	✕	✕	✕	200 to 400 (integer)		
user.std.gy	●	●	✕	✕	✕	400 to 750 (integer)		
user.std.bx	●	●	✕	✕	✕	50 to 250 (integer)		
user.std.by	●	●	✕	✕	✕	0 to 120 (integer)		
user.std.wx	●	●	✕	✕	✕	200 to 400 (integer)		
user.std.wy	●	●	✕	✕	✕	250 to 450 (integer)		
user.std.reset	✕	✕	✕	✕	●	✓	✓	✓

Continues on next page...

Notes

The `user.std` *commands are identical to the settings in the* **Setup > ColorMax > Measured Data** *menu. Protocol values are multiples of 1000.*

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision Laser 7500 & 8500	E-Vision Laser 10K	E-Vision Laser 4K
`user.target.rx`	●	●	✕	✕	✕	550 to 750 (integer)		
`user.target.ry`	●	●	✕	✕	✕	250 to 450 (integer)		
`user.target.gx`	●	●	✕	✕	✕	200 to 400 (integer)		
`user.target.gy`	●	●	✕	✕	✕	400 to 750 (integer)		
`user.target.bx`	●	●	✕	✕	✕	50 to 250 (integer)		
`user.target.by`	●	●	✕	✕	✕	0 to 120 (integer)		
`user.target.wx`	●	●	✕	✕	✕	200 to 400 (integer)		
`user.target.wy`	●	●	✕	✕	✕	250 to 450 (integer)		
`user.target.cx`	●	●	✕	✕	✕	125 to 325 (integer)		
`user.target.cy`	●	●	✕	✕	✕	225 to 425 (integer)		
`user.target.mx`	●	●	✕	✕	✕	200 to 400 (integer)		
`user.target.my`	●	●	✕	✕	✕	50 to 250 (integer)		
`user.target.yx`	●	●	✕	✕	✕	300 to 500 (integer)		
`user.target.yy`	●	●	✕	✕	✕	400 to 600 (integer)		
`user.target.reset`	✕	✕	✕	✕	●	✓	✓	✓

Continues on next page...

Notes

The `user.target` *commands are identical to the settings in the* **Setup > ColorMax > Target Data — User 1** *menu. Protocol values are multiples of 1000.*

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision Laser 7500 & 8500	E-Vision Laser 10K	E-Vision Laser 4K
`user2.target.rx`	●	●	✕	✕	✕	550 to 750 (integer)		
`user2.target.ry`	●	●	✕	✕	✕	250 to 450 (integer)		
`user2.target.gx`	●	●	✕	✕	✕	200 to 400 (integer)		
`user2.target.gy`	●	●	✕	✕	✕	400 to 750 (integer)		
`user2.target.bx`	●	●	✕	✕	✕	50 to 250 (integer)		
`user2.target.by`	●	●	✕	✕	✕	0 to 120 (integer)		
`user2.target.wx`	●	●	✕	✕	✕	200 to 400 (integer)		
`user2.target.wy`	●	●	✕	✕	✕	250 to 450 (integer)		
`user2.target.cx`	●	●	✕	✕	✕	125 to 325 (integer)		
`user2.target.cy`	●	●	✕	✕	✕	225 to 425 (integer)		
`user2.target.mx`	●	●	✕	✕	✕	200 to 400 (integer)		
`user2.target.my`	●	●	✕	✕	✕	50 to 250 (integer)		
`user2.target.yx`	●	●	✕	✕	✕	300 to 500 (integer)		
`user2.target.yy`	●	●	✕	✕	✕	400 to 600 (integer)		
`user2.target.reset`	✕	✕	✕	✕	●	✓	✓	✓
`user.p7.rst`	✕	✕	✕	✕	●	✓	✓	n/a

Continues on next page...

Notes

The `user2.target` commands are identical to the settings in the **Setup > ColorMax > Target Data — User 2** *menu. Protocol values are multiples of 1000.*

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision Laser 7500 & 8500	E-Vision Laser 10K	E-Vision Laser 4K
hsg.hue.r	●	●	●	●	✕	0 to 200 (integer)		
hsg.hue.g	●	●	●	●	✕	0 to 200 (integer)		
hsg.hue.b	●	●	●	●	✕	0 to 200 (integer)		
hsg.hue.c	●	●	●	●	✕	0 to 200 (integer)		
hsg.hue.m	●	●	●	●	✕	0 to 200 (integer)		
hsg.hue.y	●	●	●	●	✕	0 to 200 (integer)		
hsg.sat.r	●	●	●	●	✕	0 to 200 (integer)		0 to 100 (integer)
hsg.sat.g	●	●	●	●	✕	0 to 200 (integer)		0 to 100 (integer)
hsg.sat.b	●	●	●	●	✕	0 to 200 (integer)		0 to 100 (integer)
hsg.sat.c	●	●	●	●	✕	0 to 200 (integer)		0 to 100 (integer)
hsg.sat.m	●	●	●	●	✕	0 to 200 (integer)		0 to 100 (integer)
hsg.sat.y	●	●	●	●	✕	0 to 200 (integer)		0 to 100 (integer)

Continues on next page...

Notes

The `hsg` *commands are identical to the settings in the* **Color > Manual Color Matching** *menu.*

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision Laser 7500 & 8500	E-Vision Laser 10K	E-Vision Laser 4K
hsg.gain.r	●	●	●	●	✕	0 to 200 (integer)		0 to 100 (integer)
hsg.gain.g	●	●	●	●	✕	0 to 200 (integer)		0 to 100 (integer)
hsg.gain.b	●	●	●	●	✕	0 to 200 (integer)		0 to 100 (integer)
hsg.gain.c	●	●	●	●	✕	0 to 200 (integer)		0 to 100 (integer)
hsg.gain.m	●	●	●	●	✕	0 to 200 (integer)		0 to 100 (integer)
hsg.gain.y	●	●	●	●	✕	0 to 200 (integer)		0 to 100 (integer)
hsg.white.r	●	●	●	●	✕	0 to 200 (integer)		0 to 100 (integer)
hsg.white.g	●	●	●	●	✕	0 to 200 (integer)		0 to 100 (integer)
hsg.white.b	●	●	●	●	✕	0 to 200 (integer)		0 to 100 (integer)
hsg.reset	✕	✕	✕	✕	●	✓	✓	✓

Continues on next page...

Notes

The `hsg` *commands are identical to the settings in the* **Color > Manual Color Matching** *menu.*

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision Laser 7500 & 8500	E-Vision Laser 10K	E-Vision Laser 4K
Geometry								
aspect.ratio	●	●	✕	✕	✕	0 = 5:4 1 = 4:3 2 = 16:10 3 = 16:9 4 = 1.88 5 = 2.35 6 = Theaterscope 7 = Source 8 = Unscaled		
digi.zoom	●	●	✕	✕	✕	0 to 100 (integer)		
digi.pan	●	●	✕	✕	✕	-320 to +320 (integer)		
digi.pan.bound	✕	●	✕	✕	✕	-320 to +320 (integer)		
digi.scan	●	●	✕	✕	✕	-200 to +200 (integer)		
digi.scan.bound	✕	●	✕	✕	✕	-200 to +200 (integer)		
digi.zoom.rst	✕	✕	✕	✕	●	✓	✓	✓
overscan	●	●	✕	✕	✕	0 = Off 1 = Crop 2 = Zoom		
h.keystone	●	●	●	●	✕	-600 to +600 (integer)		
v.keystone	●	●	●	●	✕	-400 to +400 (integer)		
keystone.reset	✕	✕	✕	✕	●	n/a		✓
rotation	●	●	●	●	✕	-100 to +100 (integer)		
rotation.reset	✕	✕	✕	✕	●	n/a		✓

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision Laser 7500 & 8500	E-Vision Laser 10K	E-Vision Laser 4K
h.pin.barrel	●	●	●	●	✕	-150 to +300 (integer)		
v.pin.barrel	●	●	●	●	✕	-150 to +300 (integer)		
pin.barrel.reset	✕	✕	✕	✕	●	n/a		✓
4corner.ulx	●	●	●	●	✕	-192 to +192 (integer)		
4corner.uly	●	●	●	●	✕	-120 to +120 (integer)		
4corner.urx	●	●	●	●	✕	-192 to +192 (integer)		
4corner.ury	●	●	●	●	✕	-120 to +120 (integer)		
4corner.llx	●	●	●	●	✕	-192 to +192 (integer)		
4corner.lly	●	●	●	●	✕	-120 to +120 (integer)		
4corner.lrx	●	●	●	●	✕	-192 to +192 (integer)		
4corner.lry	●	●	●	●	✕	-120 to +120 (integer)		
4corner.reset	✕	✕	✕	✕	●	n/a		✓

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision Laser 7500 & 8500	E-Vision Laser 10K	E-Vision Laser 4K
arc.top	●	●	●	●	✕	-150 to +150 (integer)		n/a
arc.bottom	●	●	●	●	✕	-150 to +150 (integer)		n/a
arc.left	●	●	●	●	✕	-150 to +150 (integer)		n/a
arc.right	●	●	●	●	✕	-150 to +150 (integer)		n/a
arc.t	●	●	✕	✕	✕	n/a		-192 to +192 (integer)
arc.b	●	●	✕	✕	✕	n/a		-120 to +120 (integer)
arc.l	●	●	✕	✕	✕	n/a		-192 to +192 (integer)
arc.r	●	●	✕	✕	✕	n/a		-120 to +120 (integer)
arc.reset	✕	✕	✕	✕	●	n/a		✓
blanking.top	●	●	●	●	✕	0 to 360 (integer)		
blanking.bottom	●	●	●	●	✕	0 to 360 (integer)		
blanking.left	●	●	●	●	✕	0 to 534 (integer)		
blanking.right	●	●	●	●	✕	0 to 534 (integer)		
blanking.reset	✕	✕	✕	✕	●	✓	✓	✓

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision Laser 7500 & 8500	E-Vision Laser 10K	E-Vision Laser 4K
`warp.reset`	✕	✕	✕	✕	●	✓	✓	✓
`active.warp`	●	●	✕	✕	✕	0 = none (no warp function is set) 1 = Keystone 2 = Four Corner 3 = Rotation 4 = Pin/Barrel 5 = Arc		0 = none (no warp function is set) 1 = Keystone 2 = Four Corner 3 = Rotation 4 = Pin/Barrel
`cust.wp.write`	●	✕	✕	✕	✕	1 = User 1 file 2 = User 2 file		n/a
`cust.wp.clear`	●	✕	✕	✕	✕	1 = User 1 file 2 = User 2 file		n/a
`cust.wp.send`	●	●	✕	✕	✕	0 = custom warp transfer mode off 1 = custom warp transfer User 1 file 2 = custom warp transfer User 2 file		n/a
`cust.wp.ck.sum`	✕	●	✕	✕	✕	Returns the unsigned 32 bits check sum by summing all bytes in the current sent warp file when `cust.wp.send` is not zero		n/a
`warp.cust`	●	●	✕	✕	✕	0 = Off 1 = User 1 2 = User 2		

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision Laser 7500 & 8500	E-Vision Laser 10K	E-Vision Laser 4K
Edge Blend								
eb.stat	●	●	✕	✕	✕	0 = Off 1 = On		
eb.adl	●	●	✕	✕	✕	0 = Off 1 = On		
eb.top	●	●	●	●	✕	0, 100 to 500		
eb.bottom	●	●	●	●	✕	0, 100 to 500		
eb.left	●	●	●	●	✕	0, 100 to 500		
eb.right	●	●	●	●	✕	0, 100 to 500		
eb.blu.top	●	●	●	●	✕	0 to 32 (integer)		
eb.blu.btm	●	●	●	●	✕	0 to 32 (integer)		n/a
eb.blu.bottom	●	●	●	●	✕	n/a		0 to 32 (integer)
eb.blu.left	●	●	●	●	✕	0 to 32 (integer)		
eb.blu.right	●	●	●	●	✕	0 to 32 (integer)		
eb.all	●	●	✕	✕	✕	n/a		0 to 32 (integer)
eb.red	●	●	✕	✕	✕	0 to 255 (integer)		0 to 32 (integer)
eb.green	●	●	✕	✕	✕	0 to 255 (integer)		0 to 32 (integer)
eb.blue	●	●	✕	✕	✕	0 to 255 (integer)		0 to 32 (integer)
eb.reset	✕	✕	✕	✕	●	✓	✓	✓

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision Laser 7500 & 8500	E-Vision Laser 10K	E-Vision Laser 4K
3D								
`3d.format`	●	●	✕	✕	✕	0 = Off 1 = Auto 2 = Side-By-Side (Half) 3 = Top-And-Bottom 4 = Dual-Pipe 5 = Frame Sequential		0 = Off 1 = Auto 2 = Dual-Pipe 3 = Frame Sequential
`3d.dlplink`	●	●	✕	✕	✕	0 = Off 1 = On		
`3d.dominance`	●	●	✕	✕	✕	0 = Normal 1 = Reverse		
`3d.darktime`	●	●	✕	✕	✕	0 = 0.65 ms 1 = 1.3 ms 2 = 1.95 ms		
`3d.syncoffset`	●	●	✕	✕	✕	0 to 60 (integer)		
`3d.syncref`	✕	●	✕	✕	✕	0 = Internal 1 = External		
Laser								
`laser.mode`	●	●	✕	✕	✕	0 = Eco 1 = Normal 2 = Custom		
`laser.power`	●	●	✕	✕	✕	20-100 (20%-100% power level; only available when `laser.mode=2`)		
`laser.hours`	✕	●	✕	✕	✕	integer		

Continues on next page...

Notes

☞ `laser.power` *is only effective if* `laser.mode` *is set to custom.*

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision Laser 7500 & 8500	E-Vision Laser 10K	E-Vision Laser 4K
Setup								
altitude	●	●	✕	✕	✕	1 = On 2 = Auto 3 = Quiet		0 = reserved for other applications 1 = On 2 = Auto 3 = Quiet
cooling.condition	●	●	✕	✕	✕	0 = Table 1 = Ceiling 2 = Freetilt 3 = Auto		
orientation	●	●	✕	✕	✕	0 = Desktop Front 1 = Ceiling Front 2 = Desktop Rear 3 = Ceiling Rear 4 = Auto-front		
screen.setting	●	●	✕	✕	✕	0 = 16:10 1 = 16:9 2 = 4:3		
auto.poweroff	●	●	✕	✕	✕	0 = Off 1 = On		
auto.poweron	●	●	✕	✕	✕	0 = Off 1 = On		

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision Laser 7500 & 8500	E-Vision Laser 10K	E-Vision Laser 4K
schedule.power	●	●	✕	✕	✕	0 = Off 1 = On		
schedule1.on.day	●	●	✕	✕	✕	= 76543210 (Bit 6 = Sat, Bit5 = Fri, Bit4 = Thu, Bit3 = Wed, Bit2 = Tue, Bit1 = Mon , Bit0= Sun)		
schedule1.off.day	●	●	✕	✕	✕	= 76543210 (Bit 6 = Sat, Bit5 = Fri, Bit4 = Thu, Bit3 = Wed, Bit2 = Tue, Bit1 = Mon , Bit0= Sun)		
schedule1.on.time	●	●	✕	✕	✕	HH:MM		
schedule1.off.time	●	●	✕	✕	✕	HH:MM		
schedule2.on.day	●	●	✕	✕	✕	= 76543210 (Bit 6 = Sat, Bit5 = Fri, Bit4 = Thu, Bit3 = Wed, Bit2 = Tue, Bit1 = Mon , Bit0= Sun)		
schedule2.off.day	●	●	✕	✕	✕	= 76543210 (Bit 6 = Sat, Bit5 = Fri, Bit4 = Thu, Bit3 = Wed, Bit2 = Tue, Bit1 = Mon , Bit0= Sun)		
schedule2.on.time	●	●	✕	✕	✕	HH:MM		
schedule2.off.time	●	●	✕	✕	✕	HH:MM		
date	●	●	✕	✕	✕	yyyy/MM/dd		DD:MM:YYYY
time.zone	●	●	✕	✕	✕	-11 to +12 (integer)		
time.adjust	●	●	✕	✕	✕	HH:MM		
startup.logo	●	●	✕	✕	✕	0 = Off 1 = On		
blank.screen	●	●	✕	✕	✕	0 = Logo 1 = Black 2 = Blue 3 = White		

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision Laser 7500 & 8500	E-Vision Laser 10K	E-Vision Laser 4K
trig.1	●	●	✕	✕	✕	0 = Off 1 = Screen 2 = 5:4 3 = 4:3 4 = 16:10 5 = 16:9 6 = 1.88 7 = 2.35 8 = Theaterscope 9 = Source 10 = Unscalled 11 = RS232 12 = RS232 on 13 = RS232 off		
trig.2	●	●	✕	✕	✕	0 = Off 1 = Screen 2 = 5:4 3 = 4:3 4 = 16:10 5 = 16:9 6 = 1.88 7 = 2.35 8 = Theaterscope 9 = Source 10 = Unscalled 11 = RS232 12 = RS232 on 13 = RS232 off		
auto.source	●	●	✕	✕	✕	0 = Off 1 = On		
ir.enable	●	●	✕	✕	✕	0 = Off (Disable) 1 = On (Enable)		

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision Laser 7500 & 8500	E-Vision Laser 10K	E-Vision Laser 4K
ir.code	●	●	✕	✕	✕	00 to 99		
ir.code.rst	✕	✕	✕	✕	●	✓	✓	✓
osd.lang	●	●	✕	✕	✕	0 = English 1 = French 2 = German 3 = Spanish 4 = Simplified Chinese		
osd.menupos	●	●	✕	✕	✕	0 = Top Left 1 = Top Right 2 = Bottom Left 3 = Bottom Right 4 = Center		
osd.trans	●	●	✕	✕	✕	0 = 0% 1 = 25% 2 = 50% 3 = 75%		
osd.timer	●	●	✕	✕	✕	0 = Always On 1 = 10 Seconds 2 = 30 Seconds 3 = 60 Seconds		
osd.msgbox	●	●	✕	✕	✕	0 = Off 1 = On		n/a
recall.mem	●	●	✕	✕	✕	0 = Preset A 1 = Preset B 2 = Preset C 3 = Preset D 4 = Default		
save.mem	●	●	✕	✕	✕	0 = Preset A 1 = Preset B 2 = Preset C 3 = Preset D		

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision Laser 7500 & 8500	E-Vision Laser 10K	E-Vision Laser 4K
Network								
network.mode	●	●	✕	✕	✕	0 = Projector Control 1 = Service		n/a
standby.power						n/a		0 = Save 1 = Eco 2 = Normal
lan.power	●	●	✕	✕	✕	0 = Off 1 = On		n/a
lan.dhcp	●	●	✕	✕	✕	0 = Off 1 = On		
lan.ip	●	●	✕	✕	✕	A valid IP address in the following format: xxx.xxx.xxx.xxx		
lan.subnet	●	●	✕	✕	✕	A valid subnet address in the following format: xxx.xxx.xxx.xxx		
lan.gateway	●	●	✕	✕	✕	A valid gateway address in the following format: xxx.xxx.xxx.xxx		
lan.dns	●	●	✕	✕	✕	A valid DNS address in the following format: xxx.xxx.xxx.xxx		
lan.mac	●	●	✕	✕	✕	string		
lan.amx	●	●	✕	✕	✕	0 = Off 1 = On		

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision Laser 7500 & 8500	E-Vision Laser 10K	E-Vision Laser 4K
PIP								
pip.mode	●	●	✕	✕	✕	0 = Off 1 = On		
pip.input	●	●	✕	✕	✕	0 = HDMI 1 1 = HDMI 2 2 = RGB (VGA) 3 = COMP 4 = DisplayPort 5 = HDBaseT 6 = 3G-SDI		0 = DisplayPort 1 = HDMI 1 2 = HDMI 2 3 = HDBaseT 4 = 3G-SDI
pip.position	●	●	✕	✕	✕	0 = TopLeft 1 = TopRight 2 = BottomLeft 3 = BottomRight 4 = PBP		
pip.swap	✕	✕	✕	✕	●	✓	✓	✓
Information								
model.name	✕	●	✕	✕	✕	string		
serial	✕	●	✕	✕	✕	string		
sw.version	✕	●	✕	✕	✕	string		
sw1.version	✕	●	✕	✕	✕	string		
sw2.version	✕	●	✕	✕	✕	string		
sw3.version	✕	●	✕	✕	✕	string		
act.source	✕	●	✕	✕	✕	string		
signal	✕	●	✕	✕	✕	string		

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision Laser 7500 & 8500	E-Vision Laser 10K	E-Vision Laser 4K
h.refresh	✕	●	✕	✕	✕	number		
v.refresh	✕	●	✕	✕	✕	number		
pixel.clock	✕	●	✕	✕	✕	number		
laser.hours	✕	●	✕	✕	✕	integer		
atmos.alti	✕	●	✕	✕	✕	number		
atmos.pressure	✕	●	✕	✕	✕	number		
ac.voltage	✕	●	✕	✕	✕	0 = 90~150 1 = 160~264		
g.ceiling	✕	●	✕	✕	✕	0 = table 1 = ceiling		
g.portrait	✕	●	✕	✕	✕	number		
g.tilt	✕	●	✕	✕	✕	number		
altitude.info	✕	●	✕	✕	✕	0 = Low 1 = High	0 = SEA-1 1 = SEA-2 2 = MODE-1 3 = MODE-2 4 = MODE-3	0 = Low 1 = High
laser.power.info	✕	●	✕	✕	✕	number		

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision Laser 7500 & 8500	E-Vision Laser 10K	E-Vision Laser 4K
ti	✕	●	✕	✕	✕	number		
ti2	✕	●	✕	✕	✕	number		
tc	✕	●	✕	✕	✕	number		
tb1	✕	●	✕	✕	✕	number		
tb2	✕	●	✕	✕	✕	number		
fan1_3	✕	●	✕	✕	✕	xxxx / xxxx / xxxx (speed of FAN 1~3)		n/a
fan4_6	✕	●	✕	✕	✕	xxxx / xxxx / xxxx (speed of FAN 4~6)		n/a
fan7_9	✕	●	✕	✕	✕	xxxx / xxxx / xxxx (speed of FAN 7~9)		n/a
fan10_12	✕	●	✕	✕	✕	xxxx / xxxx / xxxx (speed of FAN 10~12)		n/a
fan13_15	✕	●	✕	✕	✕	xxxx / xxxx / xxxx (speed of FAN 13~15)		n/a
fan16_18	✕	●	✕	✕	✕	xxxx / NA / NA (speed of FAN 16)		n/a
fans	✕	●	✕	✕	✕	All fan & environment status		
water.pump	✕	●	✕	✕	✕	number		
factory.reset	✕	✕	✕	✕	●	✓	✓	✓

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision Laser 7500 & 8500	E-Vision Laser 10K	E-Vision Laser 4K
Miscellaneous								
power	●	●	✕	✕	✕	0 = Off 1 = On		
shutter	●	●	✕	✕	✕	0 = Open 1 = Close		
total.hours	✕	●	✕	✕	✕	integer		
total.minutes	✕	●	✕	✕	✕	n/a		number
laser.minutes	✕	●	✕	✕	✕	n/a		number
laser.normal.hr	✕	●	✕	✕	✕	n/a		number
laser.normal.min	✕	●	✕	✕	✕	n/a		number
laser.eco.hr	✕	●	✕	✕	✕	n/a		number
laser.eco.min	✕	●	✕	✕	✕	n/a		number
laser.reset						n/a		✓
status	✕	●	✕	✕	✕	0 = Standby 1 = Warm Up 2 = Imaging 3 = Cooling 4 = Error		
errcode	✕	●	✕	✕	✕	string		
cw.index	●	●	✕	✕	✕	n/a		0, 100 to 1000
pw.index	●	●	✕	✕	✕	n/a		0, 100 to 1000

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision Laser 7500 & 8500	E-Vision Laser 10K	E-Vision Laser 4K
dlp.pattern	●	✕	✕	✕	✕	n/a		0 = Off 1 = White 2 = Black 3 = Red 4 = Green 5 = Blue 6 = Cyan 7 = Magenta 8 = Yellow 9 = Checkboard 10 = Vramp 11 = Hramp 12 = Grid 13 = Cross, 14 = FPGA_TP_Calibration
pri.reset	✕	✕	✕	✕	●	n/a		✓
mfg.reset	✕	✕	✕	✕	●	n/a		✓
sp.index	●	●	✕	✕	✕	n/a		0, 0 to 4096
sp.index.v sp.index.h	●	●	✕	✕	✕	n/a		0, 0 to 4096
sp.t1	●	●	✕	✕	✕	n/a		0, 0 to 4096
sp.t2	●	●	✕	✕	✕	n/a		0, 0 to 4096
psoc4.ver	✕	●	✕	✕	✕	n/a		string
warp.key	✕	●	✕	✕	✕	n/a		0 = licence fail, timeout expired 1 = licence pass, timeout expired 2 = licence fail, timeout not expired 3 = licence pass, timeout not expired

Notes

HIGHlite Laser II 3D Series
HIGHlite Laser 4K Series
M-Vision Laser 18K Series

High Brightness Digital Video Projector

PROTOCOL GUIDE

Rev H October 2017

IN THIS GUIDE

Introduction

The projector can be controlled by using an external control system or a PC via an RS232 or LAN interface, using a terminal-emulation program.

Network setup

1. Connect the projector to a LAN network.
2. Open the **Setup > Network** menu and edit network settings. The default IP address is **192.168.0.100** and the TCP port number is **7000**.

Serial Port setup

- Baud rate 9,600 bps
- Data length 8 bits
- Stop bits one
- Parity none
- Flow control none

Notes

For details on connecting the projector to an RS232 or LAN network, or changing network settings, see the user manual.

Only one control path at a time should be used for protocol control. Attempts to send commands to both serial and network ports at the same time may result in unpredictable behavior.

Protocol commands

Commands are used to simulate menu operations and determine the settings of the projector, and use the following format:

- All commands consist of ASCII text strings starting with an asterisk∗ and ending with an ASCII Carriage Return character↵ (code 13): **`*command operator <value>`**↵
- The **`<command>`** string determines which setting the command will affect.
- Spaces are required before the operator and before the value.
- The **`<operator>`** string can take one of the following formats:

Command type	<operator>	Description
Set	`= <value>`	Makes the setting take the `<value>`.
Get	?	Asks what the current value is. The value is returned as an ASCII text string.
Execute		Performs an action. No operator is entered for this type of command.

Examples

`*orientation = 3`↵	sets the orientation to Rear Ceiling (for a ceiling mounted projector positioned behind the screen)
`*aspect.ratio ?`↵	asks what the current aspect ratio is
`*zoom.in`↵	commands the projector to zoom in
`*orientation=3`↵	is an invalid instruction because of the missing spaces before the operator and the value

Responses

If the command has been successful, the projector response begins with `ACK` or `ack` (“acknowledged”). For example, if the command is `*aspect.ratio = 1`↵, the projector will return `ACK aspect.ratio = 1`↵ or `ack aspect.ratio = 1`↵, depending on the model. In either case the projector will then will change the aspect ratio accordingly.

If the command has not been acknowledged, due to a syntax error or another problem, the projector response will be `NAK` or `nack`, followed by a brief description of the problem.

Notes

To set the default value of a command, simply enter the command name and ↵, without an operator. For example `*orientation`↵ will set the orientation to 0 (Desktop Front).

You must wait for the complete response to a command before sending another command.

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	HL Laser II 3D	HL Laser 4K	M-Vision Laser 18K
Input								
input	●	●	✕	✕	✕	0= HDMI1 1= HDMI2 2= RGB 3= BNC 4= DVI 5= DP 6= HDBT 7= HDSDI	0 = DisplayPort 1 = HDMI 1 2 = HDMI 2 3 = HDBaseT 4 = 3G-SDI	0 = HDMI 1 1 = HDMI 2 2 = DisplayPort 1 3 = DisplayPort 2 4 = HDBaseT 5 = 3G-SDI
Test Pattern								
test.pattern	●	●	✕	✕	✕	0= Off 1 = White 2 = Black 3 = Red 4 = Green 5 = Blue 6 = Checkerboard 7 = Crosshatch 8 = V Burst 9 = H Burst 10 = Color Bar 11 = Plunge	0 = Off 1 = White 2 = Black 3 = Red 4 = Green 5 = Blue 6 = Cyan 7 = Yellow 8 = Magenta	0= Off 1 = White 2 = Black 3 = Red 4 = Green 5 = Blue 6 = Checkerboard 7 = Crosshatch 8 = V Burst 9 = H Burst 10 = Color Bar 11 = Plunge

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	HL Laser II 3D	HL Laser 4K	M-Vision Laser 18K
Lens								
`zoom.in`	✕	✕	✕	✕	●	✓	✓	✓
`zoom.out`	✕	✕	✕	✕	●	✓	✓	✓
`focus.near`	✕	✕	✕	✕	●	✓	✓	✓
`focus.far`	✕	✕	✕	✕	●	✓	✓	✓
`lens.up`	✕	✕	✕	✕	●	✓	✓	✓
`lens.down`	✕	✕	✕	✕	●	✓	✓	✓
`lens.left`	✕	✕	✕	✕	●	✓	✓	✓
`lens.right`	✕	✕	✕	✕	●	✓	✓	✓
`lens.center`	✕	✕	✕	✕	●	✓	✓	✓
`lens.load`	●	✕	✕	✕	✕	n/a	1 to 10 (integer)	
`lens.save`	●	●	✕	✕	✕	n/a	1 to 10 (integer)	
`lens.clear`	●	✕	✕	✕	✕	n/a	1 to 10 (integer)	
`lens.type`	●	●	✕	✕	✕	n/a	1 to 10 (integer)	n/a
`lens.lock`	●	●	✕	✕	✕	n/a	0 = Off 1 = On	

Continues on next page...

Notes

Lens commands only work if the projector is switched on.

To use any lens commands except `lens.type`*, make sure the lens is unlocked. If* `lens.lock` *is set to 1, most other lens commands are disabled.*

When used with a get operator, the `lens.save` *command returns a string of zeroes and ones where each zero is an empty memory slot and each one is an occupied slot.*

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	HL Laser II 3D	HL Laser 4K	M-Vision Laser 18K
Image								
pic.mode	●	●	✕	✕	✕	0 = High Bright 1 = Presentation 2 = Video	n/a	0 = High Bright 1 = Presentation 2 = Video
db.on	●	●	✕	✕	✕	0 = Off 1 = On		
sp.on	●	●	✕	✕	✕	n/a	0 = Off 1 = On	n/a
gamma	●	●	✕	✕	✕	0 = 1.0 1 = 1.8 2 = 2.0 3 = 2.2 4 = 2.35 5 = 2.5	0 = 1.0 1 = 1.8 2 = 2.0 3 = 2.2 4 = 2.35 5 = 2.5 6 = S-Curve 7 = DICOM 8 = HDR PQ-400 9 = HDR PQ-500 10 = HDR PQ-1000 11 = HDR HLG	0 = 1.0 1 = 1.8 2 = 2.0 3 = 2.2 4 = 2.35 5 = 2.5 6 = S-curve 7 = DICOM
brightness	●	●	●	●	✕	0 to 200 (integer)		
contrast	●	●	●	●	✕	0 to 200 (integer)		
saturation	●	●	●	●	✕	0 to 200 (integer)		
hue	●	●	●	●	✕	0 to 200 (integer)		
sharpness	●	●	●	●	✕	0 to 15 (integer)	0 to 10 (integer)	0 to 15 (integer)

Continues on next page...

Notes

The values you set of `pic.mode`, `gamma`, `brightness`, `contrast`, `saturation` *and* `hue` *will only apply to the current image source.*

`db.on` *is not available in 3D.*

`db.on` *cannot be used when edge blending.*

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	HL Laser II 3D	HL Laser 4K	M-Vision Laser 18K
nr	●	●	✕	✕	✕	n/a		0 to 3 (integer)
nr.level	●	●	✕	✕	✕	n/a	0 to 3 (integer)	n/a
nr.temporal	●	●	●	●	✕	0 to 3 (integer)	n/a	
nr.block	●	●	●	●	✕	0 to 3 (integer)	n/a	
nr.mosquito	●	●	●	●	✕	0 to 3 (integer)	n/a	
nr.hori	●	●	●	●	✕	0 to 3 (integer)	n/a	
nr.vert	●	●	●	●	✕	0 to 3 (integer)	n/a	
nr.reset	●	●	✕	✕	✕	0 to 3 (integer)	n/a	
h.position	●	●	●	●	✕	0 to 200 (integer)	n/a	
v.position	●	●	●	●	✕	0 to 200 (integer)	n/a	
vga.phase	●	●	●	●	✕	0 to 31 (integer)	n/a	
tracking	●	●	●	●	✕	0 to 200 (integer)	n/a	
sync.level	●	●	●	●	✕	0 to 200 (integer)	n/a	
freeze	●	●	✕	✕	✕	0 = Off 1 = On		
resync	✕	✕	✕	✕	●	✓	✓	✓

Continues on next page...

Notes

The commands on this page will only apply to the current image source.

The `vga.phase` *command is identical to the* **Phase** *setting in the* **Image > Position and Phase** *menu.*

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	HL Laser II 3D	HL Laser 4K	M-Vision Laser 18K
Color								
color.space	●	●	✕	✕	✕	0 = Auto 1 = YPbPr 2 = YCbCr 3 = RGB-PC 4 = RGB-Video		
color.temp	●	●	✕	✕	✕	0 = 3200K 1 = 5400K 2 = 6500K 3 = 7500K 4 = 9300K 5 = Native		
color.mode	●	●	✕	✕	✕	0 = ColorMax 1 = Manual Color Matching 2 = Color Temperature 3 = Gains and Lifts		
color.max	●	●	✕	✕	✕	0 = REC709 1 = EBU 2 = SMPTE 3 = Native 4 = User 1 5 = User 2	0 = REC709 1 = EBU 2 = SMPTE 3 = Peak 4 = User 1 5 = User 2	0 = HDTV (REC709) 1 = Peak 2 = User 1 3 = User 2

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	HL Laser II 3D	HL Laser 4K	M-Vision Laser 18K
red.lift	●	●	●	●	✕	0 to 200 (integer)		
green.lift	●	●	●	●	✕	0 to 200 (integer)		
blue.lift	●	●	●	●	✕	0 to 200 (integer)		
red.gain	●	●	●	●	✕	0 to 200 (integer)		
green.gain	●	●	●	●	✕	0 to 200 (integer)		
blue.gain	●	●	●	●	✕	0 to 200 (integer)		
gainlift.reset	✕	✕	✕	✕	●	✓	n/a	✓
auto.test.ptrn	●	●	✕	✕	✕	0 = Off 1 = On		
user.std.rx	●	●	✕	✕	✕	550 to 750 (integer)		
user.std.ry	●	●	✕	✕	✕	250 to 450 (integer)		
user.std.gx	●	●	✕	✕	✕	200 to 400 (integer)		
user.std.gy	●	●	✕	✕	✕	400 to 750 (integer)		
user.std.bx	●	●	✕	✕	✕	50 to 250 (integer)		
user.std.by	●	●	✕	✕	✕	0 to 120 (integer)		
user.std.wx	●	●	✕	✕	✕	200 to 400 (integer)		
user.std.wy	●	●	✕	✕	✕	250 to 450 (integer)		
user.std.reset	✕	✕	✕	✕	●	✓	✓	✓

Continues on next page...

Notes

The `user.std` *commands are identical to the settings in the* **Setup > ColorMax > Measured Data** *menu. Protocol values are multiples of 1000.*

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	HL Laser II 3D	HL Laser 4K	M-Vision Laser 18K
`user.target.rx`	●	●	✕	✕	✕	550 to 750 (integer)		
`user.target.ry`	●	●	✕	✕	✕	250 to 450 (integer)		
`user.target.gx`	●	●	✕	✕	✕	200 to 400 (integer)		
`user.target.gy`	●	●	✕	✕	✕	400 to 750 (integer)		
`user.target.bx`	●	●	✕	✕	✕	50 to 250 (integer)		
`user.target.by`	●	●	✕	✕	✕	0 to 120 (integer)		
`user.target.wx`	●	●	✕	✕	✕	200 to 400 (integer)		
`user.target.wy`	●	●	✕	✕	✕	250 to 450 (integer)		
`user.target.cx`	●	●	✕	✕	✕	125 to 325 (integer)		
`user.target.cy`	●	●	✕	✕	✕	225 to 425 (integer)		
`user.target.mx`	●	●	✕	✕	✕	200 to 400 (integer)		
`user.target.my`	●	●	✕	✕	✕	50 to 250 (integer)		
`user.target.yx`	●	●	✕	✕	✕	300 to 500 (integer)		
`user.target.yy`	●	●	✕	✕	✕	400 to 600 (integer)		
`user.target.reset`	✕	✕	✕	✕	●	✓	✓	✓

Continues on next page...

Notes

The `user.target` *commands are identical to the settings in the* **Setup > ColorMax > Target Data — User 1** *menu. Protocol values are multiples of 1000.*

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	HL Laser II 3D	HL Laser 4K	M-Vision Laser 18K
`user2.target.rx`	●	●	✕	✕	✕	550 to 750 (integer)		
`user2.target.ry`	●	●	✕	✕	✕	250 to 450 (integer)		
`user2.target.gx`	●	●	✕	✕	✕	200 to 400 (integer)		
`user2.target.gy`	●	●	✕	✕	✕	400 to 750 (integer)		
`user2.target.bx`	●	●	✕	✕	✕	50 to 250 (integer)		
`user2.target.by`	●	●	✕	✕	✕	0 to 120 (integer)		
`user2.target.wx`	●	●	✕	✕	✕	200 to 400 (integer)		
`user2.target.wy`	●	●	✕	✕	✕	250 to 450 (integer)		
`user2.target.cx`	●	●	✕	✕	✕	125 to 325 (integer)		
`user2.target.cy`	●	●	✕	✕	✕	225 to 425 (integer)		
`user2.target.mx`	●	●	✕	✕	✕	200 to 400 (integer)		
`user2.target.my`	●	●	✕	✕	✕	50 to 250 (integer)		
`user2.target.yx`	●	●	✕	✕	✕	300 to 500 (integer)		
`user2.target.yy`	●	●	✕	✕	✕	400 to 600 (integer)		
`user2.target.reset`	✕	✕	✕	✕	●	✓	✓	✓

Continues on next page...

Notes

The `user2.target` *commands are identical to the settings in the* **Setup > ColorMax > Target Data — User 2** *menu. Protocol values are multiples of 1000.*

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	HL Laser II 3D	HL Laser 4K	M-Vision Laser 18K
`hsg.hue.r`	●	●	●	●	✕	0 to 200 (integer)		
`hsg.hue.g`	●	●	●	●	✕	0 to 200 (integer)		
`hsg.hue.b`	●	●	●	●	✕	0 to 200 (integer)		
`hsg.hue.c`	●	●	●	●	✕	0 to 200 (integer)		
`hsg.hue.m`	●	●	●	●	✕	0 to 200 (integer)		
`hsg.hue.y`	●	●	●	●	✕	0 to 200 (integer)		
`hsg.sat.r`	●	●	●	●	✕	0 to 200 (integer)	0 to 100 (integer)	0 to 200 (integer)
`hsg.sat.g`	●	●	●	●	✕	0 to 200 (integer)	0 to 100 (integer)	0 to 200 (integer)
`hsg.sat.b`	●	●	●	●	✕	0 to 200 (integer)	0 to 100 (integer)	0 to 200 (integer)
`hsg.sat.c`	●	●	●	●	✕	0 to 200 (integer)	0 to 100 (integer)	0 to 200 (integer)
`hsg.sat.m`	●	●	●	●	✕	0 to 200 (integer)	0 to 100 (integer)	0 to 200 (integer)
`hsg.sat.y`	●	●	●	●	✕	0 to 200 (integer)	0 to 100 (integer)	0 to 200 (integer)

Continues on next page...

Notes

The `hsg` *commands are identical to the settings in the* **Color > Manual Color Matching** *menu.*

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	HL Laser II 3D	HL Laser 4K	M-Vision Laser 18K
`hsg.gain.r`	●	●	●	●	✕	0 to 200 (integer)	0 to 100 (integer)	0 to 200 (integer)
`hsg.gain.g`	●	●	●	●	✕	0 to 200 (integer)	0 to 100 (integer)	0 to 200 (integer)
`hsg.gain.b`	●	●	●	●	✕	0 to 200 (integer)	0 to 100 (integer)	0 to 200 (integer)
`hsg.gain.c`	●	●	●	●	✕	0 to 200 (integer)	0 to 100 (integer)	0 to 200 (integer)
`hsg.gain.m`	●	●	●	●	✕	0 to 200 (integer)	0 to 100 (integer)	0 to 200 (integer)
`hsg.gain.y`	●	●	●	●	✕	0 to 200 (integer)	0 to 100 (integer)	0 to 200 (integer)
`hsg.white.r`	●	●	●	●	✕	0 to 200 (integer)	0 to 100 (integer)	0 to 200 (integer)
`hsg.white.g`	●	●	●	●	✕	0 to 200 (integer)	0 to 100 (integer)	0 to 200 (integer)
`hsg.white.b`	●	●	●	●	✕	0 to 200 (integer)	0 to 100 (integer)	0 to 200 (integer)
`hsg.reset`	✕	✕	✕	✕	●	✓	✓	✓

Continues on next page...

Notes

The `hsg` *commands are identical to the settings in the* **Color > Manual Color Matching** *menu.*

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	HL Laser II 3D	HL Laser 4K	M-Vision Laser 18K
Geometry								
aspect.ratio	●	●	✕	✕	✕	0 = 5:4 1 = 4:3 2 = 16:10 3 = 16:9 4 = 1.88 5 = 2.35 6 = Theaterscope 7 = Source 8 = Unscaled		
digi.zoom	●	●	✕	✕	✕	0 to 100 (integer)		
digi.pan	●	●	✕	✕	✕	-320 to +320 (integer)		
digi.pan.bound	✕	●	✕	✕	✕	-320 to +320 (integer)		
digi.scan	●	●	✕	✕	✕	-200 to +200 (integer)		
digi.scan.bound	✕	●	✕	✕	✕	-200 to +200 (integer)		
digi.zoom.rst	✕	✕	✕	✕	●	✓	✓	✓
overscan	●	●	✕	✕	✕	0 = Off 1 = Crop 2 = Zoom		
h.keystone	●	●	●	●	✕		-470 to +470 (integer)	-600 to +600 (integer)
v.keystone	●	●	●	●	✕		-400 to +400 (integer)	-400 to +400 (integer)
keystone.reset	✕	✕	✕	✕	●	n/a	✓	n/a
rotation	●	●	●	●	✕	-100 to +100 (integer)		
rotation.reset	✕	✕	✕	✕	●	n/a	✓	n/a

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	HL Laser II 3D	HL Laser 4K	M-Vision Laser 18K
h.pin.barrel	●	●	●	●	✕	-150 to +300 (integer)	-120 to +229 (integer)	-150 to +300 (integer)
v.pin.barrel	●	●	●	●	✕	-150 to +300 (integer)	-120 to +120 (integer)	-150 to +300 (integer)
pin.barrel.reset	✕	✕	✕	✕	●	n/a	✓	n/a
4corner.ulx	●	●	●	●	✕	-192 to +192 (integer)		
4corner.uly	●	●	●	●	✕	-120 to +120 (integer)		
4corner.urx	●	●	●	●	✕	-192 to +192 (integer)		
4corner.ury	●	●	●	●	✕	-120 to +120 (integer)		
4corner.llx	●	●	●	●	✕	-192 to +192 (integer)		
4corner.lly	●	●	●	●	✕	-120 to +120 (integer)		
4corner.lrx	●	●	●	●	✕	-192 to +192 (integer)		
4corner.lry	●	●	●	●	✕	-120 to +120 (integer)		
4corner.reset	✕	✕	✕	✕	●	n/a	✓	n/a
arc.top	●	●	●	●	✕	-150 to +150 (integer)	-60 to +115 (integer)	-150 to +150 (integer)
arc.bottom	●	●	●	●	✕	-150 to +150 (integer)	-60 to +114 (integer)	-150 to +150 (integer)
arc.left	●	●	●	●	✕	-150 to +150 (integer)	-60 to +115 (integer)	-150 to +150 (integer)
arc.right	●	●	●	●	✕	-150 to +150 (integer)	-60 to +114 (integer)	-150 to +150 (integer)
arc.reset	✕	✕	✕	✕	●	n/a	✓	n/a

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	HL Laser II 3D	HL Laser 4K	M-Vision Laser 18K
`blanking.top`	●	●	●	●	✕	0 to 360 (integer)		
`blanking.bottom`	●	●	●	●	✕	0 to 360 (integer)		
`blanking.left`	●	●	●	●	✕	0 to 534 (integer)		
`blanking.right`	●	●	●	●	✕	0 to 534 (integer)		
`blanking.reset`	✕	✕	✕	✕	●	✓	✓	✓
`warp.reset`	✕	✕	✕	✕	●	✓	✓	✓
`active.warp`	●	●	✕	✕	✕	0 = none (no warp function is set) 1 = Keystone 2 = Four Corner 3 = Rotation 4 = Pin/Barrel 5 = Arc		
`cust.wp.write`	●	✕	✕	✕	✕	1 = User 1 file 2 = User 2 file	n/a	1 = User 1 file 2 = User 2 file
`cust.wp.clear`	●	✕	✕	✕	✕	1 = User 1 file 2 = User 2 file	n/a	1 = User 1 file 2 = User 2 file
`cust.wp.send`	●	●	✕	✕	✕	0 = custom warp transfer mode off 1 = custom warp transfer User 1 file 2 = custom warp transfer User 2 file	n/a	0 = custom warp transfer mode off 1 = custom warp transfer User 1 file 2 = custom warp transfer User 2 file
`cust.wp.ck.sum`	✕	●	✕	✕	✕	Returns the unsigned 32 bits check sum by summing all bytes in the current sent warp file when `cust.wp.send` is not zero	n/a	Returns the unsigned 32 bits check sum by summing all bytes in the current sent warp file when `cust.wp.send` is not zero
`warp.cust`	●	●	✕	✕	✕	0 = Off 1 = User 1 2 = User 2		

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	HL Laser II 3D	HL Laser 4K	M-Vision Laser 18K
Edge Blend								
eb.stat	●	●	✕	✕	✕	0 = Off 1 = On		
eb.adl	●	●	✕	✕	✕	0 = Off 1 = On		
eb.top	●	●	●	●	✕	0, 100 to 500		
eb.bottom	●	●	●	●	✕	0, 100 to 500		
eb.left	●	●	●	●	✕	0, 100 to 800		0, 100 to 500
eb.right	●	●	●	●	✕	0, 100 to 800		0, 100 to 500
eb.blu.top	●	●	●	●	✕			0 to 32 (integer)
eb.blu.btm	●	●	●	●	✕	0 to 32 (integer)	n/a	
eb.blu.bottom	●	●	●	●	✕	n/a	0 to 32 (integer)	
eb.blu.left	●	●	●	●	✕	0 to 32 (integer)		
eb.blu.right	●	●	●	●	✕	0 to 32 (integer)		
eb.all	✕	✕	●	●	✕	0 to 255 (integer)		
eb.red	●	●	✕	✕	✕	0 to 255 (integer)	0 to 32 (integer)	0 to 255 (integer)
eb.green	●	●	✕	✕	✕	0 to 255 (integer)	0 to 32 (integer)	0 to 255 (integer)
eb.blue	●	●	✕	✕	✕	0 to 255 (integer)	0 to 32 (integer)	0 to 255 (integer)
eb.reset	✕	✕	✕	✕	●	✓	✓	✓

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	HL Laser II 3D	HL Laser 4K	M-Vision Laser 18K
3D								
`3d.format`	●	●	✕	✕	✕	0 = Off 1 = Auto 2 = Side-By-Side (Half) 3 = Top-And-Bottom 4 = Dual-Pipe 5 = Frame Sequential	n/a	0 = Off 1 = Auto 2 = Side-By-Side (Half) 3 = Top-And-Bottom 4 = Dual-Pipe 5 = Frame Sequential
`3d.dlplink`	●	●	✕	✕	✕	n/a		0 = Off 1 = On
`3d.dominance`	●	●	✕	✕	✕	0 = Normal 1 = Reverse	n/a	0 = Normal 1 = Reverse
`3d.darktime`	●	●	✕	✕	✕	0 = 0.65 ms 1 = 1.3 ms 2 = 1.95 ms 3 = 2.5 ms	n/a	0 = 0.65 ms 1 = 1.3 ms 2 = 1.95 ms
`3d.syncoffset`	●	●	✕	✕	✕	0 to 60 (integer)	n/a	0 to 200 (integer)
`3d.syncref`	✕	●	✕	✕	✕	0 = External 1 = Internal	n/a	0 = External 1 = Internal
Laser								
`laser.mode`	●	●	✕	✕	✕	0 = Eco 1 = Normal 2 = Custom		
`laser.power`	●	●	✕	✕	✕	30-100 (30%-100% power level; only available when `laser.mode=2`)		20-100 (20%-100% power level; only available when `laser.mode=2`)
`laser.hours`	✕	●	✕	✕	✕	integer		

Continues on next page...

Notes

☞ `laser.power` *is only effective if* `laser.mode` *is set to custom.*

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	HL Laser II 3D	HL Laser 4K	M-Vision Laser 18K
Setup								
altitude	●	●	✕	✕	✕	1 = On 2 = Auto	1 = Auto 2 = On	0 = On 1 = Auto 2 = Quiet
cooling.condition	●	●	✕	✕	✕	n/a		0 = Table 1 = Ceiling 2 = Freetilt 3 = Auto
orientation	●	●	✕	✕	✕	0 = Desktop Front 1 = Ceiling Front 2 = Desktop Rear 3 = Ceiling Rear		0 = Desktop Front 1 = Ceiling Front 2 = Desktop Rear 3 = Ceiling Rear 4 = Auto-front
screen.setting	●	●	✕	✕	✕	0 = 16:10 1 = 16:9 2 = 4:3		
auto.poweroff	●	●	✕	✕	✕	0 = Off 1 = On		
auto.poweron	●	●	✕	✕	✕	0 = Off 1 = On		

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	HL Laser II 3D	HL Laser 4K	M-Vision Laser 18K
schedule.power	●	●	✕	✕	✕	0 = Off 1 = On		
schedule1.on.day	●	●	✕	✕	✕	= 76543210 (Bit 6 = Sat, Bit5 = Fri, Bit4 = Thu, Bit3 = Wed, Bit2 = Tue, Bit1 = Mon , Bit0= Sun)		
schedule1.off.day	●	●	✕	✕	✕	= 76543210 (Bit 6 = Sat, Bit5 = Fri, Bit4 = Thu, Bit3 = Wed, Bit2 = Tue, Bit1 = Mon , Bit0= Sun)		
schedule1.on.time	●	●	✕	✕	✕	HH:MM		
schedule1.off.time	●	●	✕	✕	✕	HH:MM		
schedule2.on.day	●	●	✕	✕	✕	= 76543210 (Bit 6 = Sat, Bit5 = Fri, Bit4 = Thu, Bit3 = Wed, Bit2 = Tue, Bit1 = Mon , Bit0= Sun)		
schedule2.off.day	●	●	✕	✕	✕	= 76543210 (Bit 6 = Sat, Bit5 = Fri, Bit4 = Thu, Bit3 = Wed, Bit2 = Tue, Bit1 = Mon , Bit0= Sun)		
schedule2.on.time	●	●	✕	✕	✕	HH:MM		
schedule2.off.time	●	●	✕	✕	✕	HH:MM		
date	●	●	✕	✕	✕	yyyy/MM/dd	DD:MM:YYYY	yyyy/MM/dd
time.zone	●	●	✕	✕	✕	-11 to +12 (integer)		
time.adjust	●	●	✕	✕	✕	HH:MM		
startup.logo	●	●	✕	✕	✕	0 = Off 1 = On		
standby.mode	●	●	✕	✕	✕	n/a		0 = Saving 1 = Eco 2 = Normal
blank.screen	●	●	✕	✕	✕		0 = Black 1 = Blue 2 = White	

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	HL Laser II 3D	HL Laser 4K	M-Vision Laser 18K
trig.1	●	●	✕	✕	✕	0 = Off 1 = Screen 2 = 5:4 3 = 4:3 4 = 16:10 5 = 16:9 6 = 1.88 7 = 2.35 8 = Theaterscope 9 = Source 10 = Unscalled 11 = RS232 12 = RS232 on 13 = RS232 off		
trig.2	●	●	✕	✕	✕	0 = Off 1 = Screen 2 = 5:4 3 = 4:3 4 = 16:10 5 = 16:9 6 = 1.88 7 = 2.35 8 = Theaterscope 9 = Source 10 = Unscalled 11 = RS232 12 = RS232 on 13 = RS232 off		
auto.source	●	●	✕	✕	✕	0 = Off 1 = On		

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	HL Laser II 3D	HL Laser 4K	M-Vision Laser 18K
ir.enable	●	●	✕	✕	✕	n/a	0 = Off (Disable) 1 = On (Enable)	
ir.code	●	●	✕	✕	✕	00 to 99		
ir.code.rst	✕	✕	✕	✕	●	✓	✓	✓
osd.lang	●	●	✕	✕	✕	n/a	0 = English 1 = French 2 = Spanish 3 = German 4 = Spanish 5 = Simplified Chinese 6 = Japanese 7 = Korean	0 = English 1 = French 2 = German 3 = Spanish 4 = Simplified Chinese
osd.menupos	●	●	✕	✕	✕	0 = Top Left 1 = Top Right 2 = Bottom Left 3 = Bottom Right 4 = Center		
osd.trans	●	●	✕	✕	✕	0 = 0% 1 = 25% 2 = 50% 3 = 75%		
osd.timer	●	●	✕	✕	✕	0 = Always On 1 = 10 Seconds 2 = 30 Seconds 3 = 60 Seconds		
osd.msgbox	●	●	✕	✕	✕	0 = Off 1 = On		

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	HL Laser II 3D	HL Laser 4K	M-Vision Laser 18K
recall.mem	●	●	✕	✕	✕	0 = Preset A 1 = Preset B 2 = Preset C 3 = Preset D 4 = Default		
save.mem	●	●	✕	✕	✕	0 = Preset A 1 = Preset B 2 = Preset C 3 = Preset D		
Network								
network.mode	●	●	✕	✕	✕	0 = Projector Control 1 = Service	n/a	0 = Projector Control 1 = Service
lan.power	●	●	✕	✕	✕	0 = Off 1 = On	n/a	
lan.dhcp	●	●	✕	✕	✕	0 = Off 1 = On		
lan.ip	●	●	✕	✕	✕	A valid IP address in the following format: xxx.xxx.xxx.xxx		
lan.subnet	●	●	✕	✕	✕	A valid subnet address in the following format: xxx.xxx.xxx.xxx		
lan.gateway	●	●	✕	✕	✕	A valid gateway address in the following format: xxx.xxx.xxx.xxx		
lan.dns	●	●	✕	✕	✕	A valid DNS address in the following format: xxx.xxx.xxx.xxx		
lan.mac	●	●	✕	✕	✕	string		
lan.amx	●	●	✕	✕	✕	0 = Off 1 = On		

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	HL Laser II 3D	HL Laser 4K	M-Vision Laser 18K
PIP								
pip.mode	●	●	✕	✕	✕	0 = Off 1 = On		
pip.input	●	●	✕	✕	✕	0 = HDMI 1 1 = HDMI 2 2 = RGB (VGA) 3 = COMP 4 = DisplayPort 5 = HDBaseT 6 = 3G-SDI	0 = DisplayPort 1 = HDMI 1 2 = HDMI 2 3 = HDBaseT 4 = 3G-SDI	0 = HDMI 1 1 = HDMI 2 2 = DisplayPort 1 3 = DisplayPort 2 4 = HDBaseT 5 = 3G-SDI
pip.position	●	●	✕	✕	✕	0 = TopLeft 1 = TopRight 2 = BottomLeft 3 = BottomRight 4 = PBP		
Information								
model.name	✕	●	✕	✕	✕	string		
serial	✕	●	✕	✕	✕	string		
sw.version	✕	●	✕	✕	✕	string		
sw1.version	✕	●	✕	✕	✕	n/a	string	n/a
sw2.version	✕	●	✕	✕	✕	n/a	string	n/a
sw3.version	✕	●	✕	✕	✕	n/a	string	n/a
act.source	✕	●	✕	✕	✕	string		
signal	✕	●	✕	✕	✕	string		

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	HL Laser II 3D	HL Laser 4K	M-Vision Laser 18K
h.refresh	✕	●	✕	✕	✕	number		
v.refresh	✕	●	✕	✕	✕	number		
pixel.clock	✕	●	✕	✕	✕	number		
laser.hours	✕	●	✕	✕	✕	integer		
atmos.alti	✕	●	✕	✕	✕	number		
atmos.pressure	✕	●	✕	✕	✕	number		
ac.voltage	✕	●	✕	✕	✕	0 = 90~150 1 = 160~264		
g.ceiling	✕	●	✕	✕	✕	n/a		0 = table 1 = ceiling
g.portrait	✕	●	✕	✕	✕	n/a		number
g.tilt	✕	●	✕	✕	✕	n/a		number
altitude.info	✕	●	✕	✕	✕	n/a	0 = Low 1 = High	
laser.power.info	✕	●	✕	✕	✕	n/a	number	
laser.temp						n/a	number	n/a
ti	✕	●	✕	✕	✕	number		
tc	✕	●	✕	✕	✕	number		
tb1	✕	●	✕	✕	✕	n/a		number
tb2	✕	●	✕	✕	✕	n/a		number

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	HL Laser II 3D	HL Laser 4K	M-Vision Laser 18K
fan1_3	✕	●	✕	✕	✕	n/a		xxxx / xxxx / xxxx (speed of FAN 1~3)
fan4_6	✕	●	✕	✕	✕	n/a		xxxx / xxxx / xxxx (speed of FAN 4~6)
fan7_9	✕	●	✕	✕	✕	n/a		xxxx / xxxx / xxxx (speed of FAN 7~9)
fan10_12	✕	●	✕	✕	✕	n/a		xxxx / xxxx / xxxx (speed of FAN 10~12)
fan13_15	✕	●	✕	✕	✕	n/a		xxxx / xxxx / xxxx (speed of FAN 13~15)
fan16_18	✕	●	✕	✕	✕	n/a		xxxx / xxxx / xxxx (speed of FAN 16)
fans	✕	●	✕	✕	✕	All fan & environment status		n/a
water.pump	✕	●	✕	✕	✕	number	n/a	number
water.pump1	✕	●	✕	✕	✕	n/a	number	n/a
water.pump2	✕	●	✕	✕	✕	n/a	number	number
water.pump3	✕	●	✕	✕	✕	n/a	number	n/a
factory.reset	✕	✕	✕	✕	●	✓	✓	✓

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	HL Laser II 3D	HL Laser 4K	M-Vision Laser 18K
Miscellaneous								
power	●	●	✕	✕	✕	0 = Off 1 = On		
shutter	●	●	✕	✕	✕	0 = Open 1 = Close		n/a
pic.mute	●	●	✕	✕	✕	n/a		0 = Open 1 = Close
total.hours	✕	●	✕	✕	✕	number		
status	✕	●	✕	✕	✕	0 = Standby 1 = Warm Up 2 = Imaging 3 = Cooling 4 = Error		
errcode	✕	●	✕	✕	✕	string		
sys.info	✕	●	✕	✕	✕	n/a		string

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model		
	Set =	Get ?	Inc +	Dec -	Exe	HL Laser II 3D	HL Laser 4K	M-Vision Laser 18K
dlp.pattern	●	✕	✕	✕	✕	n/a	0 = Off 1 = RGB Ramps 2 = Color Bar 3 = Step Bars 4 = Check Board 5 = Grid6 = H Lines 7 = V Lines 8 = D Lines 9 = Ramp H 10 = Ramp V 11 = White 12 = Red 13 = Green 14 = Blue 15 = Black 16 = Cyan 17 = Magenta 18 = Yellow	n/a
pri.reset	✕	✕	✕	✕	●	n/a	✓	n/a
sp.power	✕	●	✕	✕	✕		0 = Off 1 = On	
sp.index.v sp.index.h	●	●	✕	✕	✕	n/a	0, 0 to 4096	n/a
warp.key	✕	●	✕	✕	✕	n/a	0 = licence fail, timeout expired 1 = licence pass, timeout expired 2 = licence fail, timeout not expired 3 = licence pass, timeout not expired	n/a

Notes

This page is intentionally left blank.

DIGITAL PROJECTION

E-Vision 6900 Series
Mercury Quad Series

High Brightness Digital Video Projector

PROTOCOL GUIDE

Rev H October 2017

IN THIS GUIDE

Introduction

The projector can be controlled by using an external control system or a PC via an RS232 or LAN interface, using a terminal-emulation program.

Network setup

1. Connect the projector to a LAN network.
2. Open the **Setup > Network** menu and edit network settings. The default IP address is **192.168.0.100** and the TCP port number is **7000**.

Serial Port setup

- Baud rate 9,600 bps
- Data length 8 bits
- Stop bits one
- Parity none
- Flow control none

Notes

For details on connecting the projector to an RS232 or LAN network, or changing network settings, see the user manual.

Only one control path at a time should be used for protocol control. Attempts to send commands to both serial and network ports at the same time may result in unpredictable behavior.

Protocol commands

Commands are used to simulate menu operations and determine the settings of the projector, and use the following format:

- All commands consist of ASCII text strings starting with an asterisk∗ and ending with an ASCII Carriage Return character↵ (code 13): **`*command operator <value>`**↵
- The **`<command>`** string determines which setting the command will affect.
- Spaces are required before the operator and before the value.
- The **`<operator>`** string can take one of the following formats:

Command type	<operator>	Description
Set	`= <value>`	Makes the setting take the `<value>`.
Get	?	Asks what the current value is. The value is returned as an ASCII text string.
Execute		Performs an action. No operator is entered for this type of command.

Examples

`*orientation = 3`↵	sets the orientation to Rear Ceiling (for a ceiling mounted projector positioned behind the screen)
`*aspect.ratio ?`↵	asks what the current aspect ratio is
`*zoom.in`↵	commands the projector to zoom in
`*orientation=3`↵	is an invalid instruction because of the missing spaces before the operator and the value

Responses

If the command has been successful, the projector response begins with `ACK` or `ack` (“acknowledged”). For example, if the command is `*aspect.ratio = 1`↵, the projector will return `ACK aspect.ratio = 1`↵ or `ack aspect.ratio = 1`↵, depending on the model. In either case the projector will then will change the aspect ratio accordingly.

If the command has not been acknowledged, due to a syntax error or another problem, the projector response will be `NAK` or `nack`, followed by a brief description of the problem.

Notes

To set the default value of a command, simply enter the command name and ↵, without an operator. For example `*orientation`↵ will set the orientation to 0 (Desktop Front).

You must wait for the complete response to a command before sending another command.

Command	Operators allowed					Values accepted / Format of response – per model	
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision 6900	Mercury Quad
Input							
`input`	●	●	✕	✕	✕	0 = HDMI I 1 = HDMI II 2 = DVI-D 3 = VGA 4 = Component 5 = HDBaseT	0 = HDMI 1 1 = HDMI 2 2 = DisplayPort 3 = HDBaseT 4 = 3G-SDI
Test Pattern							
`test.pattern`	●	●	✕	✕	✕	0 = Off 1 = White 2 = Black 3 = Red 4 = Green 5 = Blue 6 = Checkboard 7 = CrossHatch 8 = V Burst 9 = H Burst 10 = Color Bar 11 = Hramp	0 = Off 1 = White 2 = Black 3 = Red 4 = Green 5 = Blue 6 = Checkerboard 7 = Crosshatch 8 = V Burst 9 = H Burst 10 = Color Bar 11 = Plunge

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model	
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision 6900	Mercury Quad
Lens							
`zoom.in`	✕	✕	✕	✕	●	✓	✓
`zoom.out`	✕	✕	✕	✕	●	✓	✓
`focus.near`	✕	✕	✕	✕	●	✓	✓
`focus.far`	✕	✕	✕	✕	●	✓	✓
`lens.up`	✕	✕	✕	✕	●	✓	✓
`lens.down`	✕	✕	✕	✕	●	✓	✓
`lens.left`	✕	✕	✕	✕	●	✓	✓
`lens.right`	✕	✕	✕	✕	●	✓	✓
`lens.center`	✕	✕	✕	✕	●	✓	✓
`lens.load`	●	✕	✕	✕	✕	0 to 9 (integer)	1 to 10 (integer)
`lens.save`	●	✕	✕	✕	✕	0 to 9 (integer)	1 to 10 (integer)
`lens.clear`	●	✕	✕	✕	✕	0 to 9 (integer)	1 to 10 (integer)
`lens.type`	●	●	✕	✕	✕	0 = non-UST Lens 1 = UST Lens	n/a
`lens.lock`	●	●	✕	✕	✕	0 = Off 1 = On	

Continues on next page...

Notes

Lens commands only work if the projector is switched on.

To use lens commands, make sure the lens is unlocked. If `lens.lock` *is set to 1, most other lens commands are disabled.*

Exceptions are:

- `lens.type` *- for all models*
- `lens.save` *and* `lens.clear` *- for* ***Mercury Quad****.*

When used with a get operator, the `lens.save` *command returns a string of zeroes and ones where each zero is an empty memory slot and each one is an occupied slot.*

Command	Operators allowed					Values accepted / Format of response – per model	
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision 6900	Mercury Quad
Image							
pic.mode	●	●	✕	✕	✕	0 = High Bright 1 = Presentation 2 = Video	n/a
dblack	●	●	✕	✕	✕	0 = Off 1 = On	n/a
gamma	●	●	✕	✕	✕	0 = 1.0 1 = 1.8 2 = 2.0 3 = 2.2 4 = 2.35 5 = 2.5 6 = S-curve	0 = 1.0 1 = 1.8 2 = 2.0 3 = 2.2 4 = 2.35 5 = 2.5
brightness	●	●	●	●	✕	n/a	0 to 200 (integer)
	●	●	✕	✕	✕	0 to 200 (integer)	n/a
contrast	●	●	●	●	✕	n/a	0 to 200 (integer)
	●	●	✕	✕	✕	0 to 200 (integer)	n/a
saturation	●	●	●	●	✕	n/a	0 to 200 (integer)
	●	●	✕	✕	✕	0 to 200 (integer)	n/a
hue	●	●	●	●	✕	n/a	0 to 200 (integer)
	●	●	✕	✕	✕	0 to 200 (integer)	n/a
sharpness	●	●	●	●	✕	n/a	0 to 15 (integer)
	●	●	✕	✕	✕	0 to 31 (integer)	n/a

Continues on next page...

Notes

The values you set of `pic.mode`, `gamma`, `brightness`, `contrast`, `saturation` *and* `hue` *will only apply to the current image source.*

`dblack` *is not available in 3D.*

E-Vision 6900 *will only accept* `saturation` *and* `hue` *values if the input is YUV.*

Command	Operators allowed					Values accepted / Format of response – per model	
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision 6900	Mercury Quad
nr	●	●	✕	✕	✕	0 to 15 (integer)	n/a
nr.temporal	●	●	●	●	✕	n/a	0 to 3 (integer)
nr.block	●	●	●	●	✕	n/a	0 to 3 (integer)
nr.mosquito	●	●	●	●	✕	n/a	0 to 3 (integer)
nr.hori	●	●	●	●	✕	n/a	0 to 3 (integer)
nr.vert	●	●	●	●	✕	n/a	0 to 3 (integer)
nr.reset	●	●	✕	✕	✕	n/a	0 to 3 (integer)
h.position	●	●	●	●	✕	0 to 200 (integer)	
v.position	●	●	●	●	✕	0 to 200 (integer)	
vga.phase	●	●	●	●	✕	0 to 31 (integer)	
tracking	●	●	●	●	✕	0 to 200 (integer)	
sync.level	●	●	●	●	✕	n/a	0 to 200 (integer)
freeze	●	●	✕	✕	✕	0 = Off 1 = On	
resync	✕	✕	✕	✕	●	n/a	✓
vga.auto	✕	✕	✕	✕	●	✓	n/a

Continues on next page...

Notes

The commands on this page will only apply to the current image source.

The `vga.phase` *command is identical to the* **Phase** *setting in the* **Image > Position and Phase** *menu.*

Command	Operators allowed					Values accepted / Format of response – per model	
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision 6900	Mercury Quad
Color							
color.space	●	●	✕	✕	✕	0 = Auto 1 = YPbPr 2 = YCbCr 3 = RGB-PC 4 = RGB-Video	
color.temp	●	●	✕	✕	✕	0 = Native 1 = 5400K 2 = 6500K 3 = 7500K 4 = 9300K	0 = 3200K 1 = 5400K 2 = 6500K 3 = 7500K 4 = 9300K 5 = Native
color.mode	●	●	✕	✕	✕	n/a	0 = ColorMax 1 = Manual Color Matching 2 = Color Temperature 3 = Gains and Lifts
color.max	●	●	✕	✕	✕	n/a	0 = REC709 1 = EBU 2 = SMPTE 3 = Native 4 = User 1 5 = User 2

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model	
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision 6900	Mercury Quad
red.lift	●	●	●	●	✕	0 to 200 (integer)	
green.lift	●	●	●	●	✕	0 to 200 (integer)	
blue.lift	●	●	●	●	✕	0 to 200 (integer)	
red.gain	●	●	●	●	✕	0 to 200 (integer)	
green.gain	●	●	●	●	✕	0 to 200 (integer)	
blue.gain	●	●	●	●	✕	0 to 200 (integer)	
gainlift.reset	✕	✕	✕	✕	●	n/a	✓
auto.test.ptrn	●	●	✕	✕	✕	n/a	0 = Off 1 = On
user.std.rx	●	●	✕	✕	✕	n/a	550 to 750 (integer)
user.std.ry	●	●	✕	✕	✕	n/a	250 to 450 (integer)
user.std.gx	●	●	✕	✕	✕	n/a	200 to 400 (integer)
user.std.gy	●	●	✕	✕	✕	n/a	400 to 750 (integer)
user.std.bx	●	●	✕	✕	✕	n/a	50 to 250 (integer)
user.std.by	●	●	✕	✕	✕	n/a	0 to 120 (integer)
user.std.wx	●	●	✕	✕	✕	n/a	200 to 400 (integer)
user.std.wy	●	●	✕	✕	✕	n/a	250 to 450 (integer)
user.std.reset	✕	✕	✕	✕	●	n/a	✓

Continues on next page...

Notes

The `user.std` *commands are identical to the settings in the* **Setup > ColorMax > Measured Data** *menu. Protocol values are multiples of 1000.*

Command	Operators allowed					Values accepted / Format of response – per model	
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision 6900	Mercury Quad
`user.target.rx`	●	●	✕	✕	✕	n/a	550 to 750 (integer)
`user.target.ry`	●	●	✕	✕	✕	n/a	250 to 450 (integer)
`user.target.gx`	●	●	✕	✕	✕	n/a	200 to 400 (integer)
`user.target.gy`	●	●	✕	✕	✕	n/a	400 to 750 (integer)
`user.target.bx`	●	●	✕	✕	✕	n/a	50 to 250 (integer)
`user.target.by`	●	●	✕	✕	✕	n/a	0 to 120 (integer)
`user.target.wx`	●	●	✕	✕	✕	n/a	200 to 400 (integer)
`user.target.wy`	●	●	✕	✕	✕	n/a	250 to 450 (integer)
`user.target.cx`	●	●	✕	✕	✕	n/a	125 to 325 (integer)
`user.target.cy`	●	●	✕	✕	✕	n/a	225 to 425 (integer)
`user.target.mx`	●	●	✕	✕	✕	n/a	200 to 400 (integer)
`user.target.my`	●	●	✕	✕	✕	n/a	50 to 250 (integer)
`user.target.yx`	●	●	✕	✕	✕	n/a	300 to 500 (integer)
`user.target.yy`	●	●	✕	✕	✕	n/a	400 to 600 (integer)
`user.target.reset`	✕	✕	✕	✕	●	n/a	✓

Continues on next page...

Notes

The `user.target` *commands are identical to the settings in the* **Setup > ColorMax > Target Data — User 1** *menu. Protocol values are multiples of 1000.*

Command	Operators allowed					Values accepted / Format of response – per model	
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision 6900	Mercury Quad
`user2.target.rx`	●	●	✕	✕	✕	n/a	550 to 750 (integer)
`user2.target.ry`	●	●	✕	✕	✕	n/a	250 to 450 (integer)
`user2.target.gx`	●	●	✕	✕	✕	n/a	200 to 400 (integer)
`user2.target.gy`	●	●	✕	✕	✕	n/a	400 to 750 (integer)
`user2.target.bx`	●	●	✕	✕	✕	n/a	50 to 250 (integer)
`user2.target.by`	●	●	✕	✕	✕	n/a	0 to 120 (integer)
`user2.target.wx`	●	●	✕	✕	✕	n/a	200 to 400 (integer)
`user2.target.wy`	●	●	✕	✕	✕	n/a	250 to 450 (integer)
`user2.target.cx`	●	●	✕	✕	✕	n/a	125 to 325 (integer)
`user2.target.cy`	●	●	✕	✕	✕	n/a	225 to 425 (integer)
`user2.target.mx`	●	●	✕	✕	✕	n/a	200 to 400 (integer)
`user2.target.my`	●	●	✕	✕	✕	n/a	50 to 250 (integer)
`user2.target.yx`	●	●	✕	✕	✕	n/a	300 to 500 (integer)
`user2.target.yy`	●	●	✕	✕	✕	n/a	400 to 600 (integer)
`user2.target.reset`	✕	✕	✕	✕	●	n/a	✓

Continues on next page...

Notes

The `user2.target` *commands are identical to the settings in the* **Setup > ColorMax > Target Data — User 2** *menu. Protocol values are multiples of 1000.*

Command	Operators allowed					Values accepted / Format of response – per model	
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision 6900	Mercury Quad
hsg.hue.r	●	●	●	●	✕	0 to 200 (integer)	
hsg.hue.g	●	●	●	●	✕	0 to 200 (integer)	
hsg.hue.b	●	●	●	●	✕	0 to 200 (integer)	
hsg.hue.c	●	●	●	●	✕	0 to 200 (integer)	
hsg.hue.m	●	●	●	●	✕	0 to 200 (integer)	
hsg.hue.y	●	●	●	●	✕	0 to 200 (integer)	
hsg.sat.r	●	●	●	●	✕	0 to 200 (integer)	
hsg.sat.g	●	●	●	●	✕	0 to 200 (integer)	
hsg.sat.b	●	●	●	●	✕	0 to 200 (integer)	
hsg.sat.c	●	●	●	●	✕	0 to 200 (integer)	
hsg.sat.m	●	●	●	●	✕	0 to 200 (integer)	
hsg.sat.y	●	●	●	●	✕	0 to 200 (integer)	

Continues on next page...

Notes

The `hsg` *commands are identical to the settings in the* **Color > Manual Color Matching** *menu.*

Command	Operators allowed					Values accepted / Format of response – per model	
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision 6900	Mercury Quad
`hsg.gain.r`	●	●	●	●	✕	0 to 200 (integer)	
`hsg.gain.g`	●	●	●	●	✕	0 to 200 (integer)	
`hsg.gain.b`	●	●	●	●	✕	0 to 200 (integer)	
`hsg.gain.c`	●	●	●	●	✕	0 to 200 (integer)	
`hsg.gain.m`	●	●	●	●	✕	0 to 200 (integer)	
`hsg.gain.y`	●	●	●	●	✕	0 to 200 (integer)	
`hsg.white.r`	●	●	●	●	✕	0 to 200 (integer)	
`hsg.white.g`	●	●	●	●	✕	0 to 200 (integer)	
`hsg.white.b`	●	●	●	●	✕	0 to 200 (integer)	
`hsg.reset`	✕	✕	✕	✕	●	n/a	✓

Continues on next page...

Notes

The `hsg` *commands are identical to the settings in the* **Color > Manual Color Matching** *menu.*

Command	Operators allowed					Values accepted / Format of response – per model	
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision 6900	Mercury Quad
Geometry							
aspect.ratio	●	●	✕	✕	✕	0 = 5:4 1 = 4:3 2 = 16:10 3 = 16:9 4 = 1.88 5 = 2.35 6 = Theaterscope 7 = Source 8 = Unscaled	
digi.zoom	●	●	✕	✕	✕	n/a	0 to 100 (integer)
digi.pan	●	●	✕	✕	✕	n/a	-320 to +320 (integer)
digi.pan.bound	✕	●	✕	✕	✕	n/a	-320 to +320 (integer)
digi.scan	●	●	✕	✕	✕	n/a	-200 to +200 (integer)
digi.scan.bound	✕	●	✕	✕	✕	n/a	-200 to +200 (integer)
digi.zoom.rst	✕	✕	✕	✕	●	n/a	✓
overscan	●	●	✕	✕	✕	0 = Off 1 = On	0 = Off 1 = Crop 2 = Zoom
h.keystone	●	●	●	●	✕	-30 to +30 (integer)	-600 to +600 (integer)
v.keystone	●	●	●	●	✕	-30 to +30 (integer)	-400 to +400 (integer)
rotation	●	●	●	●	✕	n/a	-100 to +100 (integer)
h.pin.barrel	●	●	●	●	✕	n/a	-150 to +300 (integer)
v.pin.barrel	●	●	●	●	✕	n/a	-150 to +300 (integer)

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model	
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision 6900	Mercury Quad
4corner.ulx	●	●	●	●	✕	n/a	-192 to +192 (integer)
4corner.uly	●	●	●	●	✕	n/a	-120 to +120 (integer)
4corner.urx	●	●	●	●	✕	n/a	-192 to +192 (integer)
4corner.ury	●	●	●	●	✕	n/a	-120 to +120 (integer)
4corner.llx	●	●	●	●	✕	n/a	-192 to +192 (integer)
4corner.lly	●	●	●	●	✕	n/a	-120 to +120 (integer)
4corner.lrx	●	●	●	●	✕	n/a	-192 to +192 (integer)
4corner.lry	●	●	●	●	✕	n/a	-120 to +120 (integer)
arc.top	●	●	●	●	✕	n/a	-150 to +150 (integer)
arc.bottom	●	●	●	●	✕	n/a	-150 to +150 (integer)
arc.left	●	●	●	●	✕	n/a	-150 to +150 (integer)
arc.right	●	●	●	●	✕	n/a	-150 to +150 (integer)
blanking.top	●	●	●	●	✕	n/a	0 to 360 (integer)
blanking.bottom	●	●	●	●	✕	n/a	0 to 360 (integer)
blanking.left	●	●	●	●	✕	n/a	0 to 534 (integer)
blanking.right	●	●	●	●	✕	n/a	0 to 534 (integer)
blanking.reset	✕	✕	✕	✕	●	n/a	✓

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model	
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision 6900	Mercury Quad
`warp.reset`	✕	✕	✕	✕	●	n/a	✓
`active.warp`	●	●	✕	✕	✕	n/a	0 = none (no warp function is set) 1 = Keystone 2 = Four Coner 3 = Rotation 4 = Pin/Barrel 5 = Arc
`cust.wp.write`	●	✕	✕	✕	✕	n/a	1 = User 1 file 2 = User 2 file
`cust.wp.clear`	●	✕	✕	✕	✕	n/a	1 = User 1 file 2 = User 2 file
`cust.wp.send`	●	●	✕	✕	✕	n/a	0 = custom warp transfer mode off 1 = custom warp transfer User 1 file 2 = custom warp transfer User 2 file
`cust.wp.ck.sum`	✕	●	✕	✕	✕	n/a	Returns the unsigned 32 bits checksum by summing all bytes in the current sent warp file when `cust.wp.send` is not zero
`warp.cust`	●	●	✕	✕	✕	n/a	0 = Off 1 = User 1 2 = User 2

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model	
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision 6900	Mercury Quad
Edge Blend							
eb.stat	●	●	✕	✕	✕	n/a	0 = Off 1 = On
eb.adl	●	●	✕	✕	✕	n/a	0 = Off 1 = On
eb.top	●	●	●	●	✕	n/a	0, 100 to 500
eb.bottom	●	●	●	●	✕	n/a	0, 100 to 500
eb.left	●	●	●	●	✕	n/a	0, 100 to 800
eb.right	●	●	●	●	✕	n/a	0, 100 to 800
eb.blu.top	●	●	●	●	✕	n/a	0 to 32 (integer)
eb.blu.btm	●	●	●	●	✕	n/a	0 to 32 (integer)
eb.blu.left	●	●	●	●	✕	n/a	0 to 32 (integer)
eb.blu.right	●	●	●	●	✕	n/a	0 to 32 (integer)
eb.all	✕	✕	●	●	✕	n/a	0 to 255 (integer)
eb.red	●	●	✕	✕	✕	n/a	0 to 255 (integer)
eb.green	●	●	✕	✕	✕	n/a	0 to 255 (integer)
eb.blue	●	●	✕	✕	✕	n/a	0 to 255 (integer)
eb.reset	✕	✕	✕	✕	●	n/a	✓

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model	
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision 6900	Mercury Quad
3D							
`3d.format`	●	●	✕	✕	✕	0 = Off 1 = Auto 2 = Side-By-Side (Half) 3 = Top-And-Bottom 4 = Frame Sequential	0 = Off 1 = Auto 2 = Side-By-Side (Half) 3 = Top-And-Bottom 4 = Dual-Pipe 5 = Frame Sequential
`3d.dlplink`	●	●	✕	✕	✕	0 = Off 1 = On	n/a
`3d.dominance`	●	●	✕	✕	✕	0 = Normal 1 = Reverse	
`3d.darktime`	●	●	✕	✕	✕	n/a	0 = 0.65 ms 1 = 1.3 ms 2 = 1.95 ms 3 = 2.5 ms
`3d.syncoffset`	●	●	✕	✕	✕	0 to 200 (integer)	0 to 60 (integer)
`3d.syncref`	●	●	✕	✕	✕	0 = Internal 1 = External	n/a
	✕	●	✕	✕	✕	n/a	0 = Internal 1 = External

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model	
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision 6900	Mercury Quad
Lamp							
lamp.mode	●	●	✕	✕	✕	0 = Dual 1 = Single 2 = Lamp 1 3 = Lamp 2	0 = Eco mode 1 = Normal mode 2 = dimming mode
lamps	●	●	✕	✕	✕	n/a	0 = Dual Lamps 1 = Triple Lamps 2 = Quad Lamps
power.mode	●	●	✕	✕	✕	0 = Normal 1 = Eco 2 = Custom Power Level	n/a
lamp.power	●	●	✕	✕	✕	0-26 (80%~100%)	n/a
lamp.pow	●	●	✕	✕	✕	n/a	77-100 (77%-100%)
lamp1.hours lamp2.hours	✕	●	✕	✕	✕	number	

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model	
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision 6900	Mercury Quad
Setup							
altitude	●	●	✕	✕	✕	1 = Off 2 = On	1 = On 2 = Auto
cooling.condition	●	●	✕	✕	✕	0 = Table 1 = Ceiling 2 = Upward 3 = Downward	n/a
orientation	●	●	✕	✕	✕	0 = Desktop Front 1 = Ceiling Front 2 = Desktop Rear 3 = Ceiling Rear	0 = Desktop Front 1 = Ceiling Front 2 = Desktop Rear 3 = Ceiling Rear 4 = Vertical Up 5 = Vertical Down
screen.setting	●	●	✕	✕	✕	n/a	0 = 16:10 1 = 16:9 2 = 4:3
screen.format	●	●	✕	✕	✕	0 = 16:10 1 = 16:9 2 = 4:3	n/a
screen.shift	●	●	✕	✕	✕	If screen.format = 16:10 => 0 = 16:9 => -60 ~ 60 = 4:3 => -160 ~160	n/a
auto.poweroff	●	●	✕	✕	✕	0 = Off 1 = On	
auto.poweron	●	●	✕	✕	✕	0 = Off 1 = On	

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model	
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision 6900	Mercury Quad
schedule.power	●	●	✕	✕	✕	n/a	0 = Off 1 = On
schedule1.on.day	●	●	✕	✕	✕	n/a	= 76543210 (Bit 6 = Sat, Bit5 = Fri, Bit4 = Thu, Bit3 = Wed, Bit2 = Tue, Bit1 = Mon, Bit0= Sun)
schedule1.off.day	●	●	✕	✕	✕	n/a	= 76543210 (Bit 6 = Sat, Bit5 = Fri, Bit4 = Thu, Bit3 = Wed, Bit2 = Tue, Bit1 = Mon, Bit0= Sun)
schedule1.on.time	●	●	✕	✕	✕	n/a	HH:MM
schedule1.off.time	●	●	✕	✕	✕	n/a	HH:MM
schedule2.on.day	●	●	✕	✕	✕	n/a	= 76543210 (Bit 6 = Sat, Bit5 = Fri, Bit4 = Thu, Bit3 = Wed, Bit2 = Tue, Bit1 = Mon, Bit0= Sun)
schedule2.off.day	●	●	✕	✕	✕	n/a	= 76543210 (Bit 6 = Sat, Bit5 = Fri, Bit4 = Thu, Bit3 = Wed, Bit2 = Tue, Bit1 = Mon, Bit0= Sun)
schedule2.on.time	●	●	✕	✕	✕	n/a	HH:MM
schedule2.off.time	●	●	✕	✕	✕	n/a	HH:MM
date	●	●	✕	✕	✕	n/a	yyyy/MM/dd
time.zone	●	●	✕	✕	✕	n/a	-11 to +12 (integer)
time.adjust	●	●	✕	✕	✕	n/a	HH:MM
startup.logo	●	●	✕	✕	✕	0 = Off 1 = On	
blank.screen	●	●	✕	✕	✕	0 = Logo 1 = Black 2 = Blue	0 = Logo 1 = Black 2 = Blue 3 = White

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model	
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision 6900	Mercury Quad
trig.1	●	●	✕	✕	✕	0 = Off 1 = On	0 = Off 1 = Screen 2 = 5:4 3 = 4:3 4 = 16:10 5 = 16:9 6 = 1.88 7 = 2.35 8 = Theaterscope 9 = Source 10 = Unscalled 11 = RS232 12 = RS232 on 13 = RS232 off
trig.2	●	●	✕	✕	✕	n/a	0 = Off 1 = Screen 2 = 5:4 3 = 4:3 4 = 16:10 5 = 16:9 6 = 1.88 7 = 2.35 8 = Theaterscope 9 = Source 10 = Unscalled 11 = RS232 12 = RS232 on 13 = RS232 off
auto.source	●	●	✕	✕	✕	n/a	0 = Off 1 = On
auto.src	●	●	✕	✕	✕	0 = Off 1 = On	n/a

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model	
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision 6900	Mercury Quad
ir.enable	●	●	✕	✕	✕	n/a	0 = Off (Disable) 1 = On (Enable)
ir.code	●	●	✕	✕	✕	n/a	00 to 99
ir.code.rst	✕	✕	✕	✕	●	n/a	✓
control.id	●	●	✕	✕	✕	00~99 (0=Disable, 1~99=Enable)	n/a
osd.lang	●	●	✕	✕	✕	0 = English 1 = French 2 = Spanish 3 = German 4 = Portuguese 5 = CHS 6 = CHT 7 = Janpanese 8 = Korean	n/a
osd.menupos	●	●	✕	✕	✕	0 = Center 1 = Top Left 2 = Top Right 3 = Bottom Left 4 = Bottom Right	0 = Top Left 1 = Top Right 2 = Bottom Left 3 = Bottom Right 4 = Center
osd.trans	●	●	✕	✕	✕	n/a	0 = 0% 1 = 25% 2 = 50% 3 = 75%
osd.timer	●	●	✕	✕	✕	0 = Always On 1 = 10 Seconds 2 = 30 Seconds 3 = 60 Seconds	
osd.msgbox	●	●	✕	✕	✕	0 = Off 1 = On	

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model	
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision 6900	Mercury Quad
`recall.mem`	●	●	✕	✕	✕	n/a	0 = Preset A 1 = Preset B 2 = Preset C 3 = Preset D 4 = Default
`save.mem`	●	●	✕	✕	✕	n/a	0 = Preset A 1 = Preset B 2 = Preset C 3 = Preset D

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model	
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision 6900	Mercury Quad
Network							
network.mode	●	●	✕	✕	✕	n/a	0 = Projector Control 1 = Service
lan.power	●	●	✕	✕	✕	0 = On 1 = Off	
lan.dhcp	●	●	✕	✕	✕	0 = On 1 = Off	
lan.ip	●	●	✕	✕	✕	A valid IP address in the following format: *xxx.xxx.xxx.xxx*	
lan.subnet	●	●	✕	✕	✕	A valid subnet address in the following format: *xxx.xxx.xxx.xxx*	
lan.gateway	●	●	✕	✕	✕	A valid gateway address in the following format: *xxx.xxx.xxx.xxx*	
lan.dns	●	●	✕	✕	✕	A valid DNS address in the following format: *xxx.xxx.xxx.xxx*	
lan.mac	●	●	✕	✕	✕	n/a	string
lan.amx	●	●	✕	✕	✕	n/a	0 = On 1 = Off
PIP							
pip.mode	●	●	✕	✕	✕	n/a	0 = On 1 = Off
pip.input	●	●	✕	✕	✕	n/a	0 = HDMI 1 1 = HDMI 2 2 = RGB (VGA) 3 = COMP 4 = DisplayPort 5 = HDBaseT 6 = 3G-SDI
pip.position	●	●	✕	✕	✕	n/a	0 = TopLeft 1 = TopRight 2 = BottomLeft 3 = BottomRight 4 = PBP

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model	
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision 6900	Mercury Quad
Information							
model.name	✕	●	✕	✕	✕	string	
serial	✕	●	✕	✕	✕	string	
sw.version	✕	●	✕	✕	✕	string	
act.source	✕	●	✕	✕	✕	string	
signal	✕	●	✕	✕	✕	string	
h.refresh	✕	●	✕	✕	✕	number	
v.refresh	✕	●	✕	✕	✕	number	
pixel.clock	✕	●	✕	✕	✕	number	
lamp1.hours	✕	●	✕	✕	✕	n/a	integer
lamp2.hours	✕	●	✕	✕	✕	n/a	integer
lamp3.hours	✕	●	✕	✕	✕	n/a	integer
lamp4.hours	✕	●	✕	✕	✕	n/a	integer
brt.lock.pw	●	✕	✕	✕	✕	n/a	XXXX(4 digits = user or supervisor mode password)
brt.lock.pw.set	●	✕	✕	✕	✕	n/a	XXXX(4 digits = new user mode password)
brt.lock.level	●	●	✕	✕	✕	n/a	0 = Dual Lamps 1 = Triple Lamps 2 = Quad Lamps
brt.lock.rst	✕	✕	✕	✕	●	n/a	✓

Continues on next page...

Notes

Command	Operators allowed					Values accepted / Format of response – per model	
	Set =	Get ?	Inc +	Dec -	Exe	E-Vision 6900	Mercury Quad
atmos.alti	✕	●	✕	✕	✕	n/a	number
atmos.pressure	✕	●	✕	✕	✕	n/a	number
ac.voltage	✕	●	✕	✕	✕	n/a	0 = 90~150 1 = 160~264
ti	✕	●	✕	✕	✕	number	
tc	✕	●	✕	✕	✕	number	
fans	✕	●	✕	✕	✕	All fan & environment status	
factory.reset	✕	✕	✕	✕	●	n/a	✓
Miscellaneous							
power	●	●	✕	✕	✕	0 = Off 1 = On	
shutter	●	●	✕	✕	✕	n/a	0 = Open 1 = Close
pic.mute	●	●	✕	✕	✕	0 = Open 1 = Close	n/a
total.hours	✕	●	✕	✕	✕	n/a	integer
status	✕	●	✕	✕	✕	0 = Power Off 1 = Power On	0 = Standby 1 = Warm Up 2 = Imaging 3 = Cooling 4 = Error
errcode	✕	●	✕	✕	✕	n/a	string

Notes

INSIGHT Dual Laser 4K Series
INSIGHT 4K Quad Series
INSIGHT 4K Dual LED Series

High Brightness Digital Video Projector

PROTOCOL GUIDE

Rev H October 2017

IN THIS GUIDE

Introduction

The projector can be controlled by using an external control system or a PC via an RS232 or LAN interface, using a terminal-emulation program.

Network setup

The projector must have an IP address assigned to it before it can be controlled via protocol commands. The IP address can be user assigned or DHCP assigned..

Setting a user assigned IP address

Set a static IP address by using the OSD. Navigate to the **Network** page on the OSD. Make sure that the **DHCP** checkbox is unchecked and set your desired IP address. See the ***User Manual*** for details.

Setting a DHCP assigned IP address

To enable a DHCP assigned IP address to be set, navigate to the **Network** page on the OSD and make sure that the **DHCP** checkbox is checked. See the ***User Manual*** for details

Most home and workplace networks have a DHCP server built into the network router. If this is the case, connect the projector and the control PC to the network and log into the router's DHCP client list to find the IP address assigned to the projector. Alternatively, launch the DP Projector Controller software and use it to display the projector's IP address as shown in ***Fig. 1***.

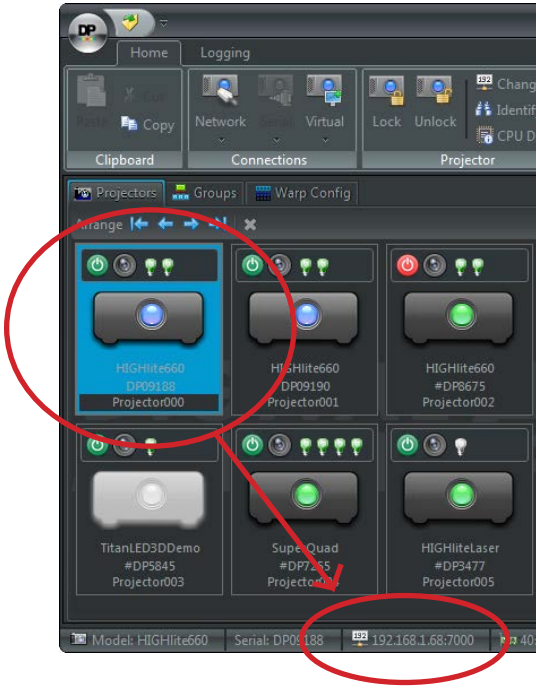

Fig. 1

Network Port setup

- IP address assigned by DHCP
- Port 7000

Serial Port setup

- Baud rate 38,400 bps
- Data length 8 bits
- Stop bits one
- Parity none
- Flow control none

Notes

Only one control path at a time should be used for protocol control. Attempts to send commands to both serial and network ports at the same time may result in unpredictable behavior.

Protocol commands

Commands are used to simulate menu operations and determine the settings of the projector, and use the following format:

- All commands consist of ASCII text strings starting with an asterisk∗ and ending with an ASCII Carriage Return character↵ (code 13):
  `*command operator <value>`↵
- The **`<command>`** string determines which setting the command will affect.
- Spaces are required before the operator and before the value.
- The **`<operator>`** string can take one of the following formats:

Command type	<operator>	Description
Set	`= <value>`	Makes the setting take the `<value>`.
Get	`?`	Asks what the current value is. The value is returned as an ASCII text string.
Increment	`+`	Adds 1 to the current value.
Increment by x	`+x`	Adds x to the current value, where x is an integer
Decrement	`-`	Subtracts 1 from the current value.
Decrement by x	`-x`	Subtracts x from the current value, where x is an integer
Default	`#`	Assigns the default value.
Execute		Performs an action. No operator is entered for this type of command.

Notes

You must wait for the complete response to a command before sending another command.

You cannot increment or decrement past the maximum / minimum value.

Examples

`*orientation = 3↵`	sets the orientation to Rear Ceiling (for a ceiling mounted projector positioned behind the screen)
`*orientation ?↵`	asks what the current orientation is
`*brightness +↵`	increases the current brightness value by 1
`*brightness +10↵`	increases the current brightness value by 10
`*brightness -↵`	decreases the current brightness value by 1
`*brightness -10↵`	decreases the current brightness value by 10
`*brightness #↵`	sets the default brightness value
`*zoom.in↵`	commands the projector to zoom in
`*orientation=3↵`	is an invalid instruction because of the missing spaces before the operator and the value

Responses

If the command has been successful, the projector response begins with `ack` (“acknowledged”). For example, if the command is `*orientation = 1↵`, the projector will return `ack orientation = 1↵`. The projector will then change the orientation accordingly.

If the command has not been acknowledged, due to a syntax error or another problem, the projector response will be `nack`, followed by a brief description of the problem.

Notes

You cannot increment or decrement past the maximum / minimum value.

Power

<command>	<operator>	<values>
power	= ?	on off
standby.mode	= ?	normal super

Inputs

<command>	<operator>	<values>
input	= ?	0 = HDMI A 1 = HDMI B 2 = DisplayPort A 3 = DisplayPort B 6 = DisplayPort A+B Dual Pipe (East/West) 7 = DisplayPort A+B Dual Pipe (Left/Right)
input.next	(execute)	
input.prev	(execute)	
input.max	?	

Test Patterns

<command>	<operator>	<value>
formatter.pattern	= ?	13 = native white 14 = native black 15 = native green 16 = native red 17 = native blue 21 = off

Notes

In `normal` *standby the projector will consume more power in standby mode but will start up more quickly.*
In `super` *standby the projector will consume less power in standby mode but will start up more slowly.*

`input` *values 6 and 7 are only available on video processor software version 29.00 and above.*

`input.next` *selects the next input in the list as follows:*
...HDMI A, HDMI B, DisplayPort A, DisplayPort B...
Alternatively, enter `input.prev` *to select the previous input from the list.*

`input.max` *returns the number of the highest available input.*

Lens

<command>	<operator>	<value>
zoom.in	(execute)	
zoom.out	(execute)	
focus.near	(execute)	
focus.far	(execute)	
lens.center	(execute)	
lens.up	=	0 - 3 (integer, movement speed: 0 = slowest, 3 = fastest)
lens.down	=	0 - 3 (integer, movement speed: 0 = slowest, 3 = fastest)
lens.left	=	0 - 3 (integer, movement speed: 0 = slowest, 3 = fastest)
lens.right	=	0 - 3 (integer, movement speed: 0 = slowest, 3 = fastest)
lens.stop	(execute)	
nudge.up	=	0 - 3 (integer, nudge time: 0 = shortest, 3 = longest)
nudge.down	=	0 - 3 (integer, nudge time: 0 = shortest, 3 = longest)
nudge.left	=	0 - 3 (integer, nudge time: 0 = shortest, 3 = longest)
nudge.right	=	0 - 3 (integer, nudge time: 0 = shortest, 3 = longest)
calibrate.zoom	(execute)	
calibrate.focus	(execute)	
lensmemory.save	=	0 - 9 (integer)
lensmemory.recall	=	0 - 9 (integer)

Image

<command>	<operator>	<value>
brightness	= ? + - #	-50 to 50 (integer)
contrast	= ? + - #	-50 to 50 (integer)
gamma	= ? #	10 to 30 (integer)
freeze	= ?	on, off

Notes

When `lens.up`, `lens.down`, `lens.left` *or* `lens.right` *is sent, the movement will continue until either a* `lens.stop` *command is sent or the limit is reached. Use a* `nudge` *command to produce a brief movement of the lens in the specified direction.*

The `gamma` *values correspond to gamma values of 1.0 to 3.0.*

The `get` *operator always returns a parametric value.*

When `freeze` *is switched on, the image freezes and the projector will keep displaying the frozen frame until* `*freeze = off`↵ *is sent. The frozen image will persist even if you disconnect the source.*

Color

<command>	<operator>	<values>
mcgd.data	= ?	green-x, green-y, red-x ,red-y, blue-x, blue-y, white-x, white-y
mcgd.factory	(execute)	
tcgd.data	= ?	green-x, green-y, red-x ,red-y, blue-x, blue-y, white-x, white-y
gamut	=	0 = Peak 1 = Rec. 709 2 = Rec. 601 3 = 3200K 4 = 5400K 5 = 6500K 6 = 8000K 7 = 9000K
red.lift	= ? + - #	-50 to +50 (integer)
green.lift	= ? + - #	-50 to +50 (integer)
blue.lift	= ? + - #	-50 to +50 (integer)
red.gain	= ? + - #	-50 to +50 (integer)
green.gain	= ? + - #	-50 to +50 (integer)
blue.gain	= ? + - #	-50 to +50 (integer)
csc.matrix	= ?	c1, c2, c3, c4, c5, c6, c7, c8, c9, Y, Cb, Cr
csc.preset	=	auto, user, rgb, yuvsd, yuvhd
pic.mute	= ? #	on, off
sample.format	= ? #	auto, rgb, 444, 422, 420

Notes

☞ `mcgd.data` *and* `tcgd.data` *allow for MCGD data or user TCGD data to be sent as comma separated x and y co-ordinates in the specified order. Must be preceded by leading 0, e.g. 0.663,0.332.*

☞ `mcgd.factory` *recovers the factory set MCGD values.*

☞ `gamut` *cannot be used as a get-type command. Once a gamut has been set, use* `*tcgd.data ?↵` *to query the values.*

☞ *Adjusting the* `brightness` *value will reset* `red.lift`, `green.lift` *and* `blue.lift` *to zero.*

☞ *Adjusting the* `contrast` *value will reset* `red.gain`, `green.gain` *and* `blue.gain` *to zero.*

Geometry

<command>	<operator>	<value>
blanking.top	= ? + - #	0 to 500 (integer)
blanking.bottom	= ? + - #	0 to 500 (integer)
blanking.left	= ? + - #	0 to 500 (integer)
blanking.right	= ? + - #	0 to 500 (integer)
blanking.coordinates	= ? #	ulx, uly, lrx, lry

3D

<command>	<operator>	<values>
3d.enable	= ?	On, off
3d.format	= ?	off – turn 3D off auto – the system will decide the best setting based on the incoming signal, if possible seq – sequential tab – top-and-bottom sbs – side-by-side (half) fpack – frame packing dplr – dual pipe left/right dpew – dual pipe east/west
3d.frmultiplier	= ? + -	1 = x1, 2 = x2, 3 = x3
3d.darktime	= ? + -	0 to 8000, steps of 50 (in µs)
3d.syncoffset	= ? + -	-1500 to 1500, steps of 100
3d.syncinpolarity	= ?	pos, neg
3d.syncoutpolarity	= ?	pos, neg
3d.syncoutenable	= ?	on, off
3d.dominance	= ?	left, right

Notes

Lamps / Lasers

<command>	<operator>	<values>

`laser1.hours`
`laser2.hours`
`lamp1.hours`
`lamp2.hours`
`lamp3.hours`
`lamp4.hours`
	?	

`laser1.strikes`
`laser2.strikes`
`lamp1.strikes`
`lamp2.strikes`
`lamp3.strikes`
`lamp4.strikes`
	?	

`laser1.serial`
`laser2.serial`
`lamp1.serial`
`lamp2.serial`
`lamp3.serial`
`lamp4.serial`
	?	

`laser.power` `lamp.power`	= ?	1 to 100 (integer)

Notes

For projectors with lamp and LED light sources, use `lamp` *in the command (*`lamp1` *to* `lamp4` *for* ***Quad****,* `lamp1` *and* `lamp2` *for* ***Dual LED****). For laser projectors, replace with* `laser`*.*

The `lampX.hours` *command (where X is the lamp number) returns the lamp hours in HH:MM format.*

Depending on the projector model, the `lamp.power` *command has a different value range as follows:*

- *For* ***INSIGHT 4K Dual Laser,*** *the range is between 30 and 100. Any value lower than 30 will be interpreted as 30 by the projector.*
- *For* ***INSIGHT 4K Quad****, the range is between 80 and 100. Any value lower than 80 will be interpreted as 80 by the projector.*
- *For* ***INSIGHT 4K Dual LED****, lamp power cannot be changed.*

Network

<command>	<operator>	<values>
lan.ip	= ?	xxx.xxx.xxx.xxx
lan.dhcp	= ?	on, off
lan.subnet	= ?	xxx.xxx.xxx.xxx

System

<command>	<operator>	<values>
orientation	= ? #	0 = Desktop Front 1 = Ceiling Front 2 = Desktop Rear 3 = Ceiling Rear
shutter	= ?	on or open off or close
ir.address	= ?	0 to 255
power	= ?	on, off
factory.reset	(execute)	
identify	(execute)	
ir.enable	= ? #	on, off
ir.key	=	0 to 127
convergence	= ? #	redX, greenX, blueX, redY, greenY, blueY (each parameter ranges from 0 to 3)

Notes

The `lan.ip` *command can only be set if* `lan.dhcp` *is set to* `off`*.*

Do not set the third octet to `254`*. You will be unable to control the projector with this setting.*

`factory.reset` *takes a long time to execute.* `'ack'` *is returned when it finishes.*

`identify` *flashes the keypad lights for 10 seconds to identify the projector.*

`ir.enable` *always defaults to* `on` *after a power cycle.*

For `ir.key` *see* ***Keypad and remote control keycode table*** *on the next page.*

Keypad and remote control keycode table

KEY_POWER_ON	= 120
KEY_POWER_OFF	= 121
KEY_SHUTTER_OPEN	= 2
KEY_SHUTTER_CLOSE	= 5
KEY_MENU	= 9
KEY_EXIT	= 40
KEY_UP	= 11
KEY_LEFT	= 18
KEY_RIGHT	= 26
KEY_DOWN	= 33
KEY_OK	= 25
KEY_INPUT_PLUS	= 10
KEY_INPUT_MINUS	= 41
KEY_0	= 82
KEY_1	= 42
KEY_2	= 46
KEY_3	= 50
KEY_4	= 55
KEY_5	= 59
KEY_6	= 63
KEY_7	= 68
KEY_8	= 72
KEY_9	= 76
KEY_10_PLUS	= 81
KEY_OSD_ON	= 1
KEY_OSD_OFF	= 4
KEY_CONTROL	= 6
KEY_AUTO	= 7
KEY_INFO	= 8
KEY_TEST	= 87
KEY_BRIGHTNESS	= 88
KEY_CONTRAST	= 89
KEY_GAMMA	= 109
KEY_RED	= 54

KEY_GREEN	= 67
KEY_BLUE	= 80
KEY_3D_ON_OFF	= 110
KEY_3D_EYE_SWAP	= 111
KEY_PIP_ON_OFF	= 112
KEY_MAIN_PIP_SWAP	= 113
KEY_UP_FOCUS	= 12
KEY_LEFT_FOCUS	= 19
KEY_RIGHT_FOCUS	= 27
KEY_DOWN_FOCUS	= 34
KEY_UP_SHIFT	= 13
KEY_LEFT_SHIFT	= 20
KEY_RIGHT_SHIFT	= 28
KEY_DOWN_SHIFT	= 35
KEY_UP_ZOOM	= 14
KEY_LEFT_ZOOM	= 21
KEY_RIGHT_ZOOM	= 29
KEY_DOWN_ZOOM	= 36
KEY_UP_ROLL	= 15
KEY_LEFT_ROLL	= 22
KEY_RIGHT_ROLL	= 30
KEY_DOWN_ROLL	= 37
KEY_UP_PITCH	= 16
KEY_LEFT_PITCH	= 23
KEY_RIGHT_PITCH	= 31
KEY_DOWN_PITCH	= 38
KEY_UP_YAW	= 17
KEY_LEFT_YAW	= 24
KEY_RIGHT_YAW	= 32
KEY_DOWN_YAW	= 39
KEY_0_LOAD	= 83
KEY_0_SAVE	= 84
KEY_0_ALT	= 85
KEY_1_LOAD	= 43

KEY_1_SAVE	= 44
KEY_1_ALT	= 45
KEY_2_LOAD	= 47
KEY_2_SAVE	= 48
KEY_2_ALT	= 49
KEY_3_LOAD	= 51
KEY_3_SAVE	= 52
KEY_3_ALT	= 53
KEY_4_LOAD	= 56
KEY_4_SAVE	= 57
KEY_4_ALT	= 58
KEY_5_LOAD	= 60
KEY_5_SAVE	= 61
KEY_5_ALT	= 62
KEY_6_LOAD	= 64
KEY_6_SAVE	= 65
KEY_6_ALT	= 66
KEY_7_LOAD	= 69
KEY_7_SAVE	= 70
KEY_7_ALT	= 71
KEY_8_LOAD	= 73
KEY_8_SAVE	= 74
KEY_8_ALT	= 75
KEY_9_LOAD	= 77
KEY_9_SAVE	= 78
KEY_9_ALT	= 79
KEY_10_PLUS_LOAD	= 101
KEY_10_PLUS_SAVE	= 107
KEY_HASH	= 86
KEY_HASH_LOAD	= 102
KEY_HASH_SAVE	= 108

Notes

Key assignments through `ir.key` *can be used with custom applications.*

OSD

<command>	<operator>	<values>
osd.enable	= ? #	on, off
osd.position	= ? #	0 to 9
osd.timeout	= ? #	0 to 120 (seconds)
osd.notifications	= ? #	on, off
osd.inputsource	= ? #	on, off
osd.version	?	
osd.zoom	= ? #	on, off
osd.pin	= ? #	0000 to 9999, the default is 1234
osd.pin.reset	(execute)	
osd.pin.enable	= ? #	on, off

Notes

OSD commands are available only where an OSD option is fitted.

When osd.timeout is set to a value of 0, the OSD will never time out.

The `osd.pin` *is used as a security measure on the web. Applies to served OSD only.*

Information

<command>	<operator>	<values>
sw.version	?	
board.id	?	
videoboard.id	?	
fw.version	?	
from.version	?	
lens.version	?	
seq.version	?	
model.name	?	
serial	?	
videosw.version	?	
standby.hours	?	
power.cycles	?	
inlet.temp	?	
dmd.temp	?	
laser.module.temp	?	

Notes

- `sw.version` *returns the software release version.*
- `board.id` *returns the CPU hardware version.*
- `videoboard.id` *returns the video hardware version.*
- `fw.version` *returns the firmware version.*
- `from.version` *returns the factory ROM version.*
- `lens.version` *returns the lens mount version.*
- `seq.version` *returns the formatter sequences version.*
- `model.name` *returns the projector model name.*
- `serial` *returns the projector serial number.*
- `videosw.version` *returns the software release version of the video processor.*
- `standby.hours` *returns the total nuimber of hours spent on standby.*
- `power.cycles` *returns the number of times the projector has been switched on at the mains inlet.*
- `inlet.temp`, `laser.module.temp` *and* `dmd.temp` *return temperature readings at the air inlet, DMD and laser module respectively.*

INSIGHT 4K Laser Series

High Brightness Digital Video Projector

PROTOCOL GUIDE

Rev H October 2017

IN THIS GUIDE

Introduction

Notes

Network Port setup

- IP address assigned by user
- Port 43728

Serial Port setup

- Baud rate 38,400 bps
- Data length 8 bits
- Stop bits one
- Parity none
- Flow control none

Command structure

Byte	Description	Comments
0	Command One (1)	First level commands
1	Command Two (2)	Second level commands
2	Command Three (3)	Third level commands
3	Length (MSB)	Number of bytes of command data
4	Length (LSB)	Number of bytes of command data
5 … n	Command Data 0 … n	
n + 1	Checksum	The LSB of the sum of all preceding bytes

In this section bytes are shown separated for clarity: each string is placed in a table where each byte occupies a separate cell.

All values are in hexadecimal format unless explicitly stated otherwise.

Checksum must be correctly calculated. The projector does not respond to an incorrect checksum.

Command example: Laser On

Byte	Value	Description
0	03	Command 1
1	2f	Command 2
2	00	Command 3
3	00	Length (MSB)
4	02	Length (LSB)
5	12	Data
6	01	
7	47	Checksum (see ***Example 1*** in the note for calculation)

The control system should wait for the full response to a command before transmitting the next command.

Notes

If the checksum is greater than 100, then only the least significant byte shall be sent.

Example 1

*The **Laser On** command looks like this:*

*03 2f 00 00 02 12 01 **47**,*

where the checksum 47 is obtained by adding up all preceding bytes:

3 + 2f + 0 + 0 + 2 + 12 + 1 = 47

Example 2

*The checksum of the **Set Light Power Leve**l command with a light power value of 1e (30%) is obtained by adding up all preceding bytes:*

*3 + 10 + 0 + 0 + 5 + c1 + ff + 0 + 1e + 0 = **1f6***

*The checksum **1f6** contains more than one byte, therefore only the LSB will be sent with the command:*

*03 10 00 00 05 c1 ff 00 1e 00 **f6***

*Similarly, if the light power value is 64 (100%), the checksum will be **23c** and the actual command will look like this:*

*03 10 00 00 05 c1 ff 00 64 00 **3c***

Protocol Commands

Control commands

Power On

Send

Command 1	Command 2	Command 3	Length MSB	Length LSB	Checksum
02	00	00	00	00	02

Response (example)

Response 1	Response 2	Response 3	Response 4	Length	Checksum
22	00	00	c0	00	e2

Notes

Power Off

Send

Command 1	Command 2	Command 3	Length MSB	Length LSB	Checksum
02	01	00	00	00	03

Response (example)

Response 1	Response 2	Response 3	Response 4	Length	Checksum
22	01	00	c0	00	e3

Notes

Light On

Send

Command 1	Command 2	Command 3	Length MSB	Length LSB	Command Data		Checksum
03	2f	00	00	02	12	01	47

Response (example)

Response 1	Response 2	Response 3	Response 4	Length	Data 1	Data 2	Checksum
23	2f	00	c0	02	12	00	26

Notes

Light Off

Send

Command 1	Command 2	Command 3	Length MSB	Length LSB	Command Data		Checksum
03	2f	00	00	02	12	02	48

Response (example)

Response 1	Response 2	Response 3	Response 4	Length	Data 1	Data 2	Checksum
23	2f	00	c0	02	12	00	26

Notes

Set Light Power Level

Send

Command 1	Command 2	Command 3	Length MSB	Length LSB	Command Data					Checksum
								Light Power Level		
03	10	00	00	05	c1	ff	00	**1e**	00	f6

Response (example)

Response 1	Response 2	Response 3	Response 4	Length	Data 1	Data 2	Checksum
23	10	00	c0	02	00	00	f5

Notes

Light Power Level *is represented as a percentage between 30% and 100% (in hex), as in the following examples:*

1e = 30 decimal
63 = 99 decimal
64 = 100 decimal

Change the byte in bold and recalculate the checksum.

Get Light Power Level

Send

Command 1	Command 2	Command 3	Length MSB	Length LSB	Command Data			Checksum
03	05	00	00	03	c1	00	00	cc

Response (example)

Response 1	Response 2	Response 3	Response 4	Length	Response Data																Checksum
												Light Power Level									
23	05	00	c0	10	02	64	00	1e	00	00	00	**63**	00	07	00	01	00	00	ff	ff	e5

Notes

Light Power Level *is represented as a percentage between 30% and 100% (in hex), as in the following examples:*

1e = 30 decimal
63 = 99 decimal
64 = 100 decimal

Get Light Status

Send

Command 1	Command 2	Command 3	Length MSB	Length LSB	Command Data	Checksum
03	2f	00	00	01	1e	51

Response (example)

Response 1	Response 2	Response 3	Response 4	Length	Response Data															Checksum
						Light Hours		Light Warning Time		Light % Remaining	Light Strike Count									
23	2f	00	c0	0f	1e	**42**	**00**	**20**	**4e**	**64**	**47**	**00**	00	00	20	4e	64	00	00	6c

Notes

Light Hours: *0042 = 66 hours*

Light Warning Time: *4e20 = 20,000 hours*

Light % Remaining: *64 = 100%*

Light Strike Count: *0047 = 71 strikes*

Douser Close

Send

Command 1	Command 2	Command 3	Length MSB	Length LSB	Checksum
02	16	00	00	00	18

Response (example)

Response 1	Response 2	Response 3	Response 4	Length	Checksum
22	16	00	c0	00	f8

Notes

Douser Open

Send

Command 1	Command 2	Command 3	Length MSB	Length LSB	Checksum
02	17	00	00	00	19

Response (example)

Response 1	Response 2	Response 3	Response 4	Length	Checksum
22	17	00	c0	00	f9

Notes

Douser Status

Send

Command 1	Command 2	Command 3	Length MSB	Length LSB	Command Data	Checksum
00	85	00	00	01	03	89

Response (example)

Response 1	Response 2	Response 3	Response 4	Length	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7	Data 8	Data 9	Data 10	Data 11	Data 12	Data 13	Data 14	Data 15	Data 16	Checksum
20	85	00	c0	10	**81**	00	00	00	00	00	00	00	00	00	ff	ff	ff	ff	ff	ff	f0

Notes

*In the **Douser Status** response:*

81 = Douser closed
00 = Douser open

Running Status

Send

Command 1	Command 2	Command 3	Length MSB	Length LSB	Command Data	Checksum
00	85	00	00	01	01	87

Response (example)

Response 1	Response 2	Response 3	Response 4	Length	Response Data															Checksum
						External Control	Power	Light Cooling Status	Power Processing	Projector Status / Mode				Light Status	Light On / Off Status					
20	85	00	c0	10	00	**00**	**01**	**00**	**00**	**0c**	00	00	00	**00**	**00**	ff	00	00	00	00	81

Notes

External Control:

00 = Off, 01 = On

Power:

00 = Off, 01 = On

Light Cooling Status:

00 = Normal, 01 = On

Power Processing:

00 = Normal
01 = Powering up or down

Projector Status / Mode:

00 = Standby
01 = Power on protect
02 = Ignition first attempt
03 = Power on running
04 = Running: power on, light on
05 = Cooling
06 = ----- (reserved)
07 = Reset wait
08 = Fan stop error
09 = Light ignition retry
0a = Light error
0c = Running: power on, light off

Light Status:

00 = Off, 01 = On

Light On / Off Status:

00 = Normal
01 = Powering up or down

Lens commands

Move Up

Send

Command 1	Command 2	Command 3	Length MSB	Length LSB	Command Data		Checksum
02	18	00	00	02	03	7f	9e

Response (example)

Response 1	Response 2	Response 3	Response 4	Length	Data	Checksum
22	18	00	c0	01	00	fb

Notes

Move Down

Send

Command 1	Command 2	Command 3	Length MSB	Length LSB	Command Data		Checksum
02	18	00	00	02	03	81	a0

Response (example)

Response 1	Response 2	Response 3	Response 4	Length	Data	Checksum
22	18	00	c0	01	00	fb

Notes

Stop Up/Down Movement

Send

Command 1	Command 2	Command 3	Length MSB	Length LSB	Command Data		Checksum
02	18	00	00	02	03	00	1f

Response (example)

Response 1	Response 2	Response 3	Response 4	Length	Data	Checksum
22	18	00	c0	01	00	fb

Notes

Move Left

Send

Command 1	Command 2	Command 3	Length MSB	Length LSB	Command Data		Checksum
02	18	00	00	02	02	81	9f

Response (example)

Response 1	Response 2	Response 3	Response 4	Length	Data	Checksum
22	18	00	c0	01	00	fb

Notes

Move Right

Send

Command 1	Command 2	Command 3	Length MSB	Length LSB	Command Data		Checksum
02	18	00	00	02	02	7f	9d

Response (example)

Response 1	Response 2	Response 3	Response 4	Length	Data	Checksum
22	18	00	c0	01	00	fb

Notes

Stop Left/Right Movement

Send

Command 1	Command 2	Command 3	Length MSB	Length LSB	Command Data		Checksum
02	18	00	00	02	02	00	1e

Response (example)

Response 1	Response 2	Response 3	Response 4	Length	Data	Checksum
22	18	00	c0	01	00	fb

Notes

Zoom In

Send

Command 1	Command 2	Command 3	Length MSB	Length LSB	Command Data		Checksum
02	18	00	00	02	00	7f	9b

Response (example)

Response 1	Response 2	Response 3	Response 4	Length	Data	Checksum
22	18	00	c0	01	00	fb

Notes

Zoom Out

Send

Command 1	Command 2	Command 3	Length MSB	Length LSB	Command Data		Checksum
02	18	00	00	02	00	81	9d

Response (example)

Response 1	Response 2	Response 3	Response 4	Length	Data	Checksum
22	18	00	c0	01	00	fb

Notes

Stop Zoom

Send

Command 1	Command 2	Command 3	Length MSB	Length LSB	Command Data		Checksum
02	18	00	00	02	00	00	1c

Response (example)

Response 1	Response 2	Response 3	Response 4	Length	Data	Checksum
22	18	00	c0	01	00	fb

Notes

Focus In

Send

Command 1	Command 2	Command 3	Length MSB	Length LSB	Command Data		Checksum
02	18	00	00	02	01	7f	9c

Response (example)

Response 1	Response 2	Response 3	Response 4	Length	Data	Checksum
22	18	00	c0	01	00	fb

Notes

Focus Out

Send

Command 1	Command 2	Command 3	Length MSB	Length LSB	Command Data		Checksum
02	18	00	00	02	01	81	9e

Response (example)

Response 1	Response 2	Response 3	Response 4	Length	Data	Checksum
22	18	00	c0	01	00	fb

Notes

Stop Focus

Send

Command 1	Command 2	Command 3	Length MSB	Length LSB	Command Data		Checksum
02	18	00	00	02	01	00	1d

Response (example)

Response 1	Response 2	Response 3	Response 4	Length	Data	Checksum
22	18	00	c0	01	00	fb

Notes

Lens Memory commands

Store Position

Send

Command 1	Command 2	Command 3	Command Data 1			Memory Number	Command 4	Command Data 2			Command Data 3							Memory Number	Command Data 4			Command Data 5			Checksum
03	b1	00	00	42	ee	**00**	01	00	...	00	6d	65	6d	6f	72	79	20	**30**	00	...	00	ff	...	ff	ba

Response (example)

Response 1	Response 2	Response 3	Response Data 1			Memory Number	Response 4	Response 5	Checksum
23	b1	00	c0	04	ee	**00**	01	00	87

Notes

*The first **Memory Number** byte will accept a value between 00 and 09.*

*The second **Memory Number** byte will accept a value between 30 and 39.*

***Command Data 2** contains thirteen identical 00 bytes.*

***Command Data 4** contains twenty-two identical 00 bytes.*

***Command Data 5** contains twenty identical ff bytes.*

Recall Position

Send

Command 1	Command 2	Command 3	Command Data 1			Memory Number	Command 4	Command Data 2			Command Data 3							Memory Number	Command Data 4			Command Data 5			Checksum
03	b1	00	00	42	ee	**00**	02	00	...	00	6d	65	6d	6f	72	79	20	**30**	00	...	00	ff	...	ff	bb

Response (example)

Response 1	Response 2	Response 3	Response Data 1			Memory Number	Response 4	Response 5	Checksum
23	b1	00	c0	04	ee	**00**	02	00	88

Notes

*The first **Memory Number** byte will accept a value between 00 and 09.*

*The second **Memory Number** byte will accept a value between 30 and 39.*

***Command Data 2** contains thirteen identical 00 bytes.*

***Command Data 4** contains twenty-two identical 00 bytes.*

***Command Data 5** contains twenty identical ff bytes.*

Delete Position

Send

Command 1	Command 2	Command 3	Command Data 1			Memory Number	Command 4	Command Data 2			Command Data 3							Memory Number	Command Data 4			Command Data 5			Checksum
03	b1	00	00	42	ee	**00**	00	00	...	00	6d	65	6d	6f	72	79	20	**30**	00	...	00	ff	...	ff	b9

Response (example)

Response 1	Response 2	Response 3	Response Data 1			Memory Number	Response 4	Response 5	Checksum
23	b1	00	c0	04	ee	**00**	00	00	86

Notes

*The first **Memory Number** byte will accept a value between 00 and 09.*

*The second **Memory Number** byte will accept a value between 30 and 39.*

***Command Data 2** contains thirteen identical 00 bytes.*

***Command Data 4** contains twenty-two identical 00 bytes.*

***Command Data 5** contains twenty identical ff bytes.*

Title selection commands (Preset buttons)

Set Title

Send

Command 1	Command 2	Command 3	Length MSB	Length LSB	Command Data		Checksum
						Title Number	
02	03	00	00	02	06	**00**	0d

Response (example)

Response 1	Response 2	Response 3	Response 4	Length	Data	Checksum
22	03	00	c0	01	00	e6

Notes

***Title number**: 00 to 63 (i.e. between 0 and 99 decimal).*

Title count in the protocol begins from 00. To set the correct title number, convert the decimal value to hex and then decrease the result by 1. For example, if you want to set title 12, send a value of 0b, the hex equivalent of (decimal) 11.

Get Current Title

Send

Command 1	Command 2	Command 3	Length MSB	Length LSB	Command Data	Checksum
02	85	00	00	01	02	88

Response (example)

Response 1	Response 2	Response 3	Response 4	Length	Response Data															Checksum
						Title Number														
20	85	00	c0	10	00	**0e**	04	0d	02	00	00	00	ff	00	00	00	00	00	00	95

Notes

***Title number**: 00 to 63 (i.e. between 0 and 99 decimal).*

*Title count in the protocol begins from 00. To get the real title number, convert the protocol value to decimal and then increase the result by 1. For example, if **Get Current Title** returns a value of 0e (14 in decimal), the current title is 15.*

This page is intentionally left blank.

Contact Information:

Digital Projection Limited

Greenside Way, Middleton
Manchester M24 1XX, UK

Registered in England No. 2207264
Registered Office: as above

Tel (+44) 161 947 3300
Fax (+44) 161 684 7674

enquiries@digitalprojection.co.uk
service@digitalprojection.co.uk

www.digitalprojection.co.uk

Digital Projection Inc.

55 Chastain Road, Suite 115
Kennesaw, GA 30144, USA

Tel (+1) 770 420 1350
Fax (+1) 770 420 1360

powerinfo@digitalprojection.com
www.digitalprojection.com

Digital Projection China

中国 北京市 朝阳区 芍药居北里101号
世奥国际中心A座2006室(100029)

Rm A2006
ShaoYaoJu 101 North Lane
Shi Ao International Center
Chaoyang District
Beijing 100029, PR CHINA

Tel (+86) 10 84888566
Fax (+86) 10 84888566-805

techsupport@dp-china.com.cn
www.dp-china.com.cn

Digital Projection Asia

16 New Industrial Road
#02-10 Hudson Technocentre
Singapore 536204

Tel (+65) 6284-1138
Fax (+65) 6284-1238

www.digitalprojectionasia.com