

PROTON OS

Reference Manual

Table of Contents

TABLE OF CONTENTS	2
1 INTRODUCTION	5
2 GENERAL OPERATION	5
2.1 POWER AND CONTROL CONNECTOR.....	5
2.2 STATUS LED.....	5
2.3 OVERTEMPERATURE PROTECTION.....	6
3 FIRMWARE UPDATE.....	6
3.1 DATA TRANSFER	7
3.2 INSTALL AND VERIFY.....	7
3.3 PROTON UPDATER GUI.....	7
4 CONTROL SOFTWARE	8
5 COMMAND INTERFACE.....	8
5.1 OPERATIONAL MODES	8
5.1.1 <i>Controller Mode</i>	8
5.1.1.1 RS485 Addressing.....	9
5.1.1.2 RS485 Broadcasting.....	9
5.1.2 <i>Interactive Mode</i>	9
5.2 COMMAND FORMAT	10
5.2.1 <i>Parameter Data Types</i>	10
5.2.2 <i>Hierarchical Command Structure</i>	11
5.2.3 <i>Command Types</i>	11
5.2.3.1 Setter and Getter Commands.....	11
5.2.3.2 Pure Getter Commands.....	11
5.2.3.3 Getter Commands with Arguments	11
5.2.3.4 Special Commands	12
5.2.4 <i>Error Codes</i>	12
5.3 COMMAND ALIAS AND PROVIDEO PROTOCOL COMPATIBILITY	12
5.4 BUILT-IN HELP	13
5.5 AUTO COMPLETION	13
6 SETTINGS HANDLING	13
6.1 RESET	14
6.2 AUTO-SAVE.....	14
7 COMMAND REFERENCE	14
7.1 GENERAL COMMANDS.....	14
7.1.1 <i>alias</i>	15
7.1.2 <i>clear</i>	15
7.1.3 <i>help</i>	15
7.1.4 <i>history</i>	15
7.1.5 <i>rem</i>	15
7.1.6 <i>resize</i>	15
7.2 SETTINGS COMMANDS	15
7.2.1 <i>settings save</i>	15
7.2.2 <i>settings load</i>	15
7.2.3 <i>settings reset</i>	15
7.2.4 <i>settings auto_save</i>	16
7.3 SYSTEM COMMANDS.....	16
7.3.1 <i>system info</i>	16
7.3.2 <i>system name</i>	17
7.3.3 <i>system runtime</i>	18
7.3.4 <i>system reboot</i>	18
7.3.5 <i>system update</i>	18
7.3.6 <i>system identify</i>	18
7.3.7 <i>system error</i>	18

7.3.8	<i>system volatile</i>	18
7.3.9	<i>system rs485</i>	19
7.3.9.1	system rs485 mode	19
7.3.9.2	system rs485 device_address	19
7.3.9.3	system rs485 broadcast_address	19
7.3.9.4	system rs485 broadcast_master	19
7.3.9.5	system rs485 baudrate	20
7.3.9.5.1	system rs485 baudrate list	20
7.3.10	<i>system status_led</i>	20
7.3.11	<i>system tally</i>	20
7.3.12	<i>system temp</i>	20
7.3.12.1	system temp max	21
7.3.12.2	system temp reset	21
7.3.12.3	system temp count	21
7.3.13	<i>system audio</i>	21
7.3.13.1	system audio gain	21
7.3.14	<i>system timecode</i>	22
7.3.14.1	system timecode value	22
7.3.14.2	system timecode pause	22
7.3.14.3	system timecode drop	22
7.4	CAMERA COMMANDS	23
7.4.1	<i>camera gain</i>	23
7.4.2	<i>camera exposure</i>	23
7.4.3	<i>camera gain_mode</i>	23
7.4.3.1	camera gain_mode list	24
7.4.4	<i>camera info</i>	24
7.4.5	<i>camera auto</i>	24
7.4.5.1	camera auto mode	25
7.4.5.1.1	camera auto mode list	26
7.4.5.2	camera auto target	26
7.4.5.3	camera auto speed	26
7.4.5.4	camera auto max_gain	26
7.4.5.5	camera auto anti_flicker	26
7.4.5.5.1	camera auto anti_flicker list	27
7.4.5.6	camera auto custom	27
7.4.5.7	camera auto weight	27
7.5	VIDEO COMMANDS	27
7.5.1	<i>video mode</i>	28
7.5.1.1	video mode list	28
7.5.2	<i>video flip</i>	28
7.5.2.1	video flip list	29
7.5.3	<i>video black_sensor</i>	29
7.5.3.1	video black_sensor red	29
7.5.3.2	video black_sensor green	29
7.5.3.3	video black_sensor blue	29
7.5.4	<i>video lsc</i>	30
7.5.4.1	video lsc preset	30
7.5.4.1.1	video lsc preset list	31
7.5.5	<i>video wb</i>	31
7.5.5.1	video wb gain	31
7.5.5.1.1	video wb gain red	32
7.5.5.1.2	video wb gain green	32
7.5.5.1.3	video wb gain blue	32
7.5.5.1.4	video wb gain offset	32
7.5.5.1.4.1	video wb gain offset red	33
7.5.5.1.4.2	video wb gain offset green	33
7.5.5.1.4.3	video wb gain offset blue	33
7.5.5.2	video wb color_cross	33
7.5.5.2.1	video wb color_cross offset	34
7.5.5.3	video wb temp	34
7.5.5.4	video wb preset	34
7.5.5.5	video wb auto	35
7.5.5.5.1	video wb auto speed	35
7.5.5.5.2	video wb auto offset	36
7.5.6	<i>video bpc</i>	36

7.5.6.1	video bpc calibrate.....	36
7.5.7	video flare.....	36
7.5.7.1	video flare red.....	37
7.5.7.2	video flare green.....	37
7.5.7.3	video flare blue.....	37
7.5.8	video color_space.....	37
7.5.8.1	video color_space list.....	37
7.5.9	video sdi_range.....	37
7.5.10	video filter.....	38
7.5.10.1	video filter detail.....	38
7.5.10.2	video filter denoise.....	38
7.5.11	video mcc.....	38
7.5.11.1	video mcc mode.....	39
7.5.11.1.1	video mcc mode list.....	39
7.5.11.2	video mcc phase.....	39
7.5.11.3	video mcc blink.....	40
7.5.12	video black_master.....	41
7.5.12.1	video black_master red.....	41
7.5.12.2	video black_master green.....	41
7.5.12.3	video black_master blue.....	41
7.5.13	video knee.....	41
7.5.14	video lut.....	42
7.5.14.1	video lut mode.....	43
7.5.14.1.1	video lut mode list.....	43
7.5.14.2	video lut gamma.....	43
7.5.14.3	video lut max_brightness.....	44
7.5.14.4	video lut master_gain.....	44
7.5.14.5	video lut mode_compat.....	44
7.5.15	video post.....	44
7.5.15.1	video post brightness.....	45
7.5.15.2	video post contrast.....	45
7.5.15.3	video post saturation.....	45
7.5.15.4	video post hue.....	45
7.5.16	video osd.....	45
7.5.16.1	video osd list.....	46
8	ALIAS REFERENCE.....	46
	APPENDIX A: SOFTWARE LICENSES.....	49
	APACHE LICENSE VERSION 2.0.....	49
	“FAST APPROX” LIBRARY.....	52
	APPENDIX B: DOCUMENT REVISION HISTORY.....	53

1 Introduction

This manual describes the usage of PROTON cameras running the PROTON Operating System (short: PROTON OS).

Details on the general operation of a camera running PROTON OS can be found in chapter 2. Instructions for firmware updates in chapter 3.

PROTON OS uses a custom control protocol, but it is also compatible with the ProVideo protocol from ATOM one cameras via alias functions (see chapter 5.3). This means PROTON cameras can be controlled with a special version of the ProVideo GUI, for details and a download link see chapter 4.

If you want to control the device via a terminal application or a custom hardware controller, see chapter 4 for tool recommendations and chapter 5 and following for a detailed description of the command protocol.

This manual covers the following PROTON devices:

- PROTON CAM

2 General Operation

The device is connected via two sockets which are either integrated into the housing or detached via cables:

1. Power and control: 6 pin Hirose HR10 connector, see chapter 2.1.
2. SDI video: Mini BNC or Micro BNC connector.

The device will immediately power on when the supply voltage is connected, the boot process takes a few seconds. Once the device is operational the LED on the backside will blink blue. For details on the status LED see chapter 2.2.

The device is controlled via an RS485 half-duplex serial interface. The camera acts as a slave device and will not send data without a request from the host. That makes it possible to connect multiple devices to the same host without data corruption. In this case each device must have a unique device address. For details on how device addressing works, see chapter 5.1.1.1.

To control the device attach it to an RCP (e.g. [CyanView](#)) or connect it to a PC using the bundled RS485 USB dongle.

TODO: Add example schematics of how the device can be connected.

2.1 Power and Control Connector

For power and control a 6 pin Hirose HR10 connector is used. The pin assignment is as follows:

Table 1: Pinning of power and control connector.

Pin	Wire Color	Function
1	White	RS485 A / +
2	Black	RS485 B / -
3		Unused
4		Unused
5	Blue	Ground
6	Brown	Supply Voltage (4.5V to 25V)

TODO: Add image of connector with PIN numbers.

2.2 Status LED

On the back of the device is a RGB status LED that indicates the current device state. You can change the brightness of the status LED with the `system status_led` command. The following blink codes are possible:

Table 2: Status LED blink codes.

State	Blink Code	Description
Boot	Blinks cyan, 2.5x per second (200ms on, 200ms off)	Device is booting after power got connected. This state is very short so normally this code is barely visible for a few milliseconds. If a firmware install is interrupted due to power-loss this state will take longer while the bootloader recovers.
Verify / Upgrade Firmware	Blinks green, 2.5x per second (200ms on, 200ms off)	Device is verifying or installing a firmware. This happens both during normal boot (a few seconds) and during a firmware upgrade (about two minutes).
Boot Error	Flashes red, 1x per second (200ms on, 800ms off)	Device failed to start. This is a critical error that cannot be recovered. Contact PROTON customer support.
Startup	Solid purple	Device is initializing after boot (loading stored settings).
Normal	Blinks blue, 2.5x per second (200ms on, 200ms off)	Device is streaming video and waiting for commands.
Busy	Blinks yellow, 2.5x per second (200ms on, 200ms off)	Device is busy processing a command.
Error	Blinks red, 2.5x per second (200ms on, 200ms off)	Device encountered an error, e.g. while loading settings or processing a command or due to a previous over temperature event. This should not happen during normal operation. To get the error log use the <code>system error</code> command. If the error persists, contact PROTON customer support.
Over Temperature	Blinks red long, 1x per second (800ms on, 200ms off)	Device reached critical temperature and is in cooldown mode. For details see chapter 2.3.

2.3 Overtemperature Protection

Since PROTON cameras are optimized for a small form factor, they will reach high temperatures under operation. You can check the current temperature using the `system temp` command. The device also logs the maximum temperature since the last power cycle which can be read using the `system temp max` command.

To protect the device, it will enter a cooldown mode if the critical system temperature of **90°C** is reached. Cooldown mode is signaled by a red status LED which is turned on long (800ms) and off shortly (200ms). In this mode the camera does not output a video signal to reduce the heat output and reach a save temperature again. The device will still respond to commands, but most camera and video commands will fail or have no effect. This error can be checked remotely using the `system error` command.

Once the system temperature falls below **85°C** video processing is restarted. To make it easier to detect previous over temperature events the LED will continue to blink red (200ms on, 200ms off) and the previous over temperature event can be read using the `system error` command

Every time the camera reaches the critical temperature the over temperature counter is incremented. It can be read using the `system temp count` command. The counter is persistent and is not reset by a power cycle or reboot.

For more details regarding the temperature commands see section 7.3.12.

3 Firmware Update

Firmware updates are performed via the camera's serial interface. The update is a two-step process:

1. Transfer new firmware to the device.
2. Install and verify new firmware.

3.1 Data Transfer

Before data transfer is started, the camera must be switched to firmware update mode, see `system update` command for details. The update process is robust regarding interruptions and data corruption.

The update speed mainly depends on the configured RS485 baud rate which can be changed before starting the update with the `system rs485 baudrate` command.

3.2 Install and Verify

After the firmware has been transferred successfully the camera is restarted with the `system reboot` command. During boot it will detect the new firmware, install and verify it. This process takes about **two minutes**, during install the status LED will blink cyan and green. In case the image cannot be verified by the bootloader (wrong image uploaded, data got corrupted) the update will be aborted, and the camera starts with the previous firmware.

After the firmware has been installed and verified the camera continues to boot. During initialization of the application a self-check is performed. Should the camera not be able to initialize, the status LED will blink red. In this case the camera will revert to the previous firmware when it is power cycled or rebooted with the `system reboot` command.

3.3 PROTON Updater GUI

Firmware updates are performed via the PROTON Updater GUI application which automatically executes the transfer, install and verify steps. The GUI is supplied with every firmware release:

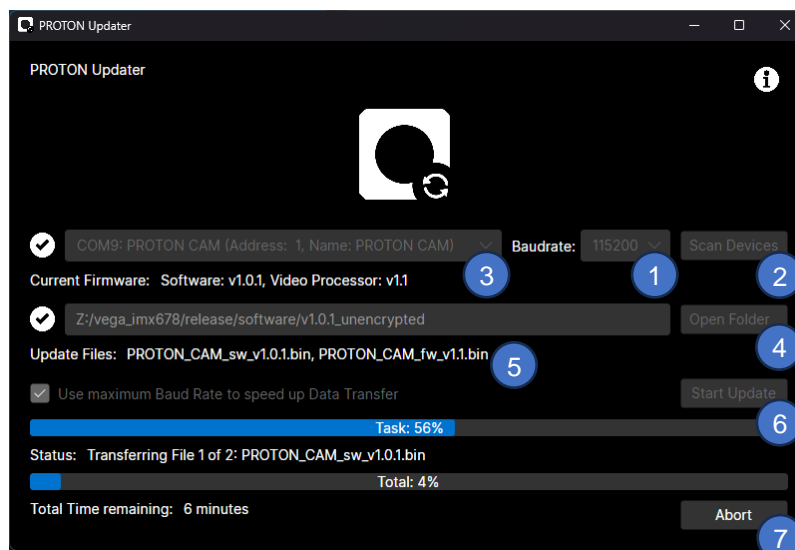


Figure 1: Example for the PROTON Updater GUI

To install a firmware update with the PROTON Updater GUI, follow these steps:

- Connect your camera to your PC using a USB to RS485 adapter. It is possible to have multiple devices on the same RS485 interface.
- Open the "PROTON Updater.exe" from the firmware release folder.
- Select the baud rate which is currently configured in the camera (1) and click *Scan Devices* (2), the GUI will automatically scan all available COM ports for connected PROTON devices and show them in the drop-down menu besides the *Scan Devices* button.
- If only one camera is connected it will automatically be selected. If you have multiple cameras connected (via separate RS485 dongles or on the same bus) select the camera you want to update from the drop-down list (3).
- Click the *Open Folder* button (4) and navigate to the folder which contains the firmware updates you want to install.

- Verify that the GUI has found the correct update files, they are displayed below the update folder path (5).
- Click *Start Update* (6) The whole process takes about 6 minutes. The upper progress bar will fill several times (once for each update file and finally for the install and verify step). The overall progress and the estimated total time remaining are displayed at the bottom of the window.
- The process can be aborted anytime by clicking the *Abort* button (7). Unless you close the GUI or disconnect the camera, the progress is retained, and you can continue by clicking the *Start Update* button (6) again.
- **Note:** Only after all files have been transferred successfully the update is made permanent. If the camera is power cycled before all files are transferred, the progress is lost.

4 Control Software

PROTON cameras are compatible with the ProVideo protocol and can be controlled using a special version of the ProVideo GUI which is available for download here:

<https://gitlab.com/proton-camera/provideo-gui/-/releases>

The above repository also includes the full source code, feel free to adapt it to your needs if you respect the used open-source licenses.

In the future a custom GUI application will be provided which is currently under development.

For direct control via commands, you can also use any terminal application which supports opening serial ports including:

- Putty: <https://www.putty.org/>
- TeraTerm: <https://teratermproject.github.io/index-en.html>

5 Command Interface

PROTON cameras are controlled via a text-based command interface on the RS485 serial port. All commands consist of human readable ASCII characters.

The interface uses the following settings:

- Default baud rate: 115200 baud (adjustable, see `system rs485 baudrate` command)
- 8-bit data, no parity, 1 stop bit (aka 8BitN1)
- No HW flow control

5.1 Operational Modes

The camera has two operational modes:

1. Controller Mode: Default mode, RS485 addressing is enabled, echo is disabled.
2. Interactive Mode: For testing and debugging, RS485 addressing is disabled, echo is enabled.

These are described in detail below.

To switch between the modes, use the `system rs485 mode` command or the `controller` and `interactive` alias commands.

5.1.1 Controller Mode

This is the default mode. In controller mode the camera uses RS485 addressing (see below) so that multiple cameras can be connected to one host on the same RS485 bus.

In this mode the camera has reduced output to keep the RS485 bus as free as possible:

- No prompt printed at new line.

- No echo output.
- No or reduced help and error messages.

This mode is used when controlling the camera via a HW controller or the GUI.

5.1.1.1 RS485 Addressing

In controller mode each command send to the camera must be prefixed with the cameras RS485 address. The address is an integer value from 0 to 99 which can be changed using the `system rs485 device_address` command.

When multiple devices are connected to the same RS485 bus each camera must have a unique address, otherwise multiple devices will reply to the same command causing garbage on the bus. You will have to setup each device separately before connecting them to the same bus.

The address 100 is reserved as the fail-safe address. Every device will always reply to commands send on address 100. This can be used if the current device address is unknown but should only be used if only one device is currently connected to the RS485 bus. Exception: To scan the bus for connected devices you can send the `system identify` command to the fail-safe address and all devices will report back in order of their device addresses. For details see the `identify` command description.

5.1.1.2 RS485 Broadcasting

In addition to a unique device address each camera also has a broadcast address. Multiple cameras that have the same broadcast address form a broadcast group. Within each group one camera acts as the broadcast master. Commands which are send to the broadcast group are processed by all camera which are part of that group, but only the broadcast master will reply keeping the bus clean.

The Broadcast address can be changed using the `system rs485 broadcast_address` command, valid addresses are values from 0 to 99 but it is not allowed to set it to the same value as the device address.

After the broadcast address has been configured select the broadcast master using the `system rs485 broadcast_master` command. The following example shows how to setup the broadcast group 10 which consists of three cameras with the device addresses 1, 2 and 3 where camera 3 becomes the broadcast master:

```
→ 1 system rs485 broadcast_address 10
← OK
→ 2 system rs485 broadcast_address 10
← OK
→ 3 system rs485 broadcast_address 10
← OK
→ 10 system rs485 broadcast_master 3
← OK
```

Note that the last OK was sent from camera 3 while camera 1 and 2 processed the command but stayed silent.

5.1.2 Interactive Mode

Interactive mode is intended for **single device operation** (only one camera on the RS485 bus) using a terminal program.

In this mode the camera provides an interactive console (like a UNIX shell) so local echo must be turned off and the terminal program should support VT100 emulation for the best user experience (see chapter 4 for recommended terminal applications). This mode is mainly used for debugging, but it is also a good way to play around with the command interface and get to know the camera.

Interactive mode disables RS485 addressing, so commands must not be prefixed with the device address. Also, the camera produces extended output:

- At the start of each line the prompt `proton-os:~$` is printed.
- Echo is enabled so all characters send to the device are sent back to the host.
- Extended help and error messages.

Additionally, there are several comfort features enabled which implement a fully featured command shell:

- Command editing: You can navigate in the typed text by using the *Arrow Left* and *Arrow Right* keys, delete text with *Backspace* or *Delete*, insert text with *Insert* and jump to the start or end of the command with the *Home* and *End* keys.
- Tab auto completion: Type only part of a command and hit *Tab* to automatically complete the command. When multiple matches are found all of them are printed.
- Command history: Use the *Arrow Up* and *Arrow Down* keys to quickly scroll through the last typed commands. To print the whole history, use the `history` command.
- Clear: Send the `clear` command to clear the whole screen.
- Resize: Adjust the width of the terminal to your current terminal application's window size with the `resize` command.
- Colored output and cursor control: Messages are colored for better readability (e.g. errors are red, warnings yellow and info messages green) and the cursor is controlled using VT100 commands.
- Extended command output: Some commands (e.g. `video mode list`) will print additional information.

The shell also supports the following meta-keys:

Table 3: Shell Meta-Keys.

Meta-Key	Action
Ctrl + A	Moves the cursor to the beginning of the line.
Ctrl + B	Moves the cursor backward one character.
Ctrl + C	Preserves the last command on the screen and starts a new command in a new line.
Ctrl + D	Deletes the character under the cursor.
Ctrl + E	Moves the cursor to the end of the line.
Ctrl + F	Moves the cursor forward one character.
Ctrl + K	Deletes from the cursor to the end of the line.
Ctrl + L	Clears the screen and leaves the currently typed command at the top of the screen.
Ctrl + N	Moves in history to next entry.
Ctrl + P	Moves in history to previous entry.
Ctrl + U	Clears the currently typed command.
Ctrl + W	Removes the word or part of the word to the left of the cursor. Words separated by period instead of space are treated as one word.
Alt + B	Moves the cursor backward one word.
Alt + F	Moves the cursor forward one word.

5.2 Command Format

A command consists of one or multiple command words followed by no, one, or multiple parameters.

In **controller mode** the general format of a command is:

```
<RS485 address> <command name> <parameters>
```

In **interactive mode** the RS485 address is omitted:

```
<command name> <parameters>
```

The camera stores each received character in an input buffer until a new-line character (Carriage Return (CR) or Line Feed (LF)) is received. Then the command is evaluated and, if valid, executed. During command execution no new commands should be send to the camera. Once execution is done the camera replies with the commands result (nothing or a string) followed by either `OK` or in case of an error with `FAIL` followed by an error code.

Note: In the following examples the camera has the default RS485 address of 1.

5.2.1 Parameter Data Types

The following parameter data types are supported:

- **Signed Decimals**, e.g.: -2947, 40687
- **Signed Hexadecimals**, e.g.: -0x100, 0x123AF7
- **Strings**, e.g.: plain_text_string
- **Booleans**: For commands that accept a Boolean parameter, like an enable flag, the following values can be used:
 - o 0 or 1
 - o true or false
 - o on or off
 - o enable or disable

For an example the following commands have the same effect:

```
settings auto_save 1
settings auto_save true
settings auto_save on
settings auto_save enable
```

5.2.2 Hierarchical Command Structure

Commands are structured hierarchical, that means a command can have subcommands. A command string is built by concatenating command words starting from the top level. For an example the command to get or set the RS485 baud rate is:

```
system rs485 baudrate
```

It is part of the `system` top-level command group which has the `rs485` subcommand group which provides the `baudrate` command.

5.2.3 Command Types

The following chapter list the different command types.

5.2.3.1 Setter and Getter Commands

Most of the commands provide two modes, a “Setter” mode to change a setting and a “Getter” mode to retrieve the current value of the setting.

A setter commands takes one or multiple parameters and applies the given values. The reply only consists of either OK or FAIL <error_code> and no further output.

Example:

```
→ 1 video mode 9
← OK
```

A getter command takes no arguments and replies with the command name followed by one or multiple values and is terminated by either OK or FAIL <error_code>.

The above `video mode` command can be called without arguments to act like a getter command:

```
→ 1 video mode
← video mode 9
← OK
```

5.2.3.2 Pure Getter Commands

There are also commands which are pure getters, that means they do not have a setter function, e.g.:

```
→ 1 system runtime
← system runtime 237700
← OK
```

5.2.3.3 Getter Commands with Arguments

These are special getter commands which require one or multiple arguments. Example:

```
→ 1 system temp 0
```

```

← system temp 0 43.6 CPU
← OK

```

5.2.3.4 Special Commands

Some commands combine multiple of the above modes or are completely unique, for an example the `system info` command will only print information but does not have the leading command name and the `video mcc phase` command can be called as a setter, a getter (which lists all MCC phases) or as a getter with arguments (to only list one MCC phase).

These intricacies are described in detail in the following chapters.

5.2.4 Error Codes

The following table lists the most common error codes which can be returned after the `FAIL` keyword:

Table 4: Common command error codes.

Error Code	Description
1	Help message was printed because the command was malformed.
-8	Command not found: The command is unknown and cannot be executed.
-71	Invalid number of parameters: The number of parameters does not match the given command or the
-22 or -34	Invalid parameter value(s): The given parameters are outside the valid value range.
-134	Operation not supported: The requested operation is not supported.
-140	Operation is currently not allowed because device is in firmware update or over temperature protection mode.

For an example the video mode command expects exactly 1 parameter. If it is called with 2 parameters, the camera replies with error -71:

```

→ 1 video mode 1 2
← FAIL -71

```

5.3 Command Alias and ProVideo Protocol Compatibility

To be compatible with the ProVideo protocol and to support shortened command names a command can have an alias. Instead of using the full command syntax, the alias can be used. For an example the command `system temp max` can also be called by its alias `temp_max`.

For compatibility with the ProVideo protocol, some commands behave slightly different when called via the alias instead of the full command syntax. This is noted in the respective command description, one example is the `system info` command.

To get a list of all available aliases use the `alias` command. Example:

```

→ 1 alias
← Available command alias:
← cam_gain -> camera gain
← cam_exposure -> camera exposure
← ...
← OK

```

Aliases can also be combined with the normal command syntax. For an example to change the MCC phase setting you can use any of:

```

video mcc phase (full syntax)
mcc phase (mcc alias for video mcc)
mcc_set (alias for video mcc phase)

```

5.4 Built-in Help

PROTON OS includes extensive help messages for all commands. To list general help instructions and a list of all top-level commands, use the `help` command.

To get specific help messages for a command send the command name followed by `-h`. Example:

```
→ 1 video lut -h
← lut - [alias: lut_enable] [getter] [getter with args]
← Enable gamma LUT (0 = bypass, 1 = enabled)
← If called via the alias 'lut_enable' this function expects two
← arguments:
← Index and enable flag. Since only one LUT is supported, the index is
← always 0. This is done for compatibility reasons.
← Usage: lut <enable: [0, 1]>
← lut_enable <idx: 0> <enable: [0, 1]>
← Subcommands:
← mode : [alias: log_mode] [getter]
← Set LUT mode. To get supported modes use the 'lut mode
← list'
← subcommand.
← ...
← OK
```

The first line of the reply contains some general information about the command (e.g., does it have an alias). It is followed by a detailed description of the command and its usage. Finally, all of the command's subcommands are listed (if it has any).

5.5 Auto Completion

It is possible to call commands without using their full name if the name is unique. For an example the `system rs485 device_address` command could also be called as:

```
→ system rs485 dev
← system rs485 dev 1
← OK
```

But it is not possible to call the `rs485 broadcast_address` command like this:

```
→ system rs485 br
← FAIL -71
```

Because that would be ambiguous with the `broadcast_master` command.

Note: For getter commands the camera always replies with the same command name that was used to query the value, e.g. `system rs485 dev` instead of `system rs485 device_address` for the above example.

Auto completion also works for aliases, for an example you can use `save`, `load` and `reset` instead of the `save_settings`, `load_settings` and `reset_settings` aliases.

Warning: When new commands are added to camera in the future old commands may become ambiguous. Therefore, it is not recommended to use shortened commands in your controller software!

6 Settings Handling

The device has two types of settings storage:

1. Runtime Storage: Gets reset as soon as the device is rebooted, or power cycled.
2. Persistent Storage: Permanent storage that survives reboot and power-cycle.

By default, all changes made to settings are only stored in the runtime storage. To transfer settings between the two storage types, use the following commands:

- `settings save`: Save all settings to the persistent settings storage.
- `settings load`: Load and apply all settings from persistent storage (restore stored state).

6.1 Reset

The camera can be reset to its default state using the `settings reset` command. When called without parameters it will only reset non-critical settings.

To perform a full reset, use the `settings reset all` command, it will also reset the following critical settings:

- Device name
- RS485 configuration:
 - Baud rate
 - Mode (controller / interactive)
 - Device and broadcast addresses
- Auto-save (reset to disable)

Note: The bad pixel table calibration data is never reset. Should you find the data to be incorrect (defect pixels are visible), recalibrate the camera as described in chapter 7.5.6.1.

6.2 Auto-Save

It is also possible to save settings automatically after a change has been made, to do so enable the auto-save feature using the `settings auto_save` command:

```
→ 1 settings auto_save 1
← OK
```

Saving settings is a slow operation, therefore command execution will take a lot longer when auto-save is enabled. For this reason, auto-save is disabled by default and manual saving is recommended for a responsive user experience.

Note: For usage with **CyanView RCPs** auto-save should always be disabled as the RCP manages all settings internally and enabling auto-save in the camera has no benefit.

7 Command Reference

The following chapters document all commands in detail. Each sub-section describes a command group.

Notes regarding the command tables:

- If a command has no alias, it will be indicated by a forward slash (/)
- Commands that have a “Getter Function” reply with their command name first if they are called with no parameters or the required number of parameters for the getter function. Commands that do not have a getter function either reply with a special string (without sending the command name first) or do not produce any output at all despite the final `OK / FAIL` delimiter. See chapter 5.2.3 for details.
- Most commands that change a setting have a Default value. This is the value which the setting is reset to when calling the `settings reset` command.
- In the command syntax parameters are written in angle brackets, e.g. `<parameter name>`, and optional parameters are additionally wrapped in round brackets like `(<optional parameter>)`.

7.1 General Commands

These commands are called without any parent command. They control basic shell functionality or print help messages.

7.1.1 alias

Command	alias		
Alias	/	Getter Function	No
Description	Print a list of all available command aliases.		

7.1.2 clear

Command	clear		
Alias	/	Getter Function	No
Description	Clear screen / terminal. Only works in interactive mode.		

7.1.3 help

Command	help		
Alias	/	Getter Function	No
Description	Print the top-level help message which lists basic help instructions and a list of the top-level commands.		

7.1.4 history

Command	history		
Alias	/	Getter Function	No
Description	Print a list of the recently used commands.		

7.1.5 rem

Command	rem		
Alias	/	Getter Function	No
Description	Ignore the following command. Can be used when running a script to comment out single lines. When using this in controller mode, "rem " must be added in front of the address.		

7.1.6 resize

Command	resize		
Alias	/	Getter Function	No
Description	Resize terminal output to current terminal window width. Only works in interactive mode.		

7.2 Settings Commands

These commands control the handling of camera settings. They are called with the `settings` command prefix.

7.2.1 settings save

Command	settings save		
Alias	save_settings	Getter Function	No
Description	Save current device settings to persistent storage.		

7.2.2 settings load

Command	settings load		
Alias	load_settings	Getter Function	No
Description	Load device settings from persistent storage and apply them.		

7.2.3 settings reset

Command	settings reset <all>		
Alias	reset_settings	Getter Function	No
Description	Resets all settings to the default value. To save settings afterwards use the <code>save</code> command (not needed when auto-save is enabled).		

	By default, only non-critical settings are reset. If all settings shall be reset (including RS485 configuration) call the command with <code>all</code> as shown below. For a full list of all critical settings see chapter 6.1.
	<code>settings reset</code> Reset non-critical settings.
	<code>settings reset all</code> Reset all settings.

Note: For a full list of settings that are reset by the `reset all` command, see chapter 6.1.

7.2.4 settings auto_save

Command	<code>settings auto_save <enable></code>		
Alias	<code>auto_save</code>	Getter Function	Yes
Description	Enable automatic saving of settings after each change. Saving settings is a slow operation, that means commands will take a lot longer to execute with auto-save enabled. This makes interacting with the camera slower, therefore auto-save is disabled by default and manual saving is recommended. Turning auto-save on or off immediately saves the settings.		
Parameter	<code>enable</code>		
Type	Boolean		
Description	Enable or disable auto-save.		
Default	Off		

7.3 System Commands

These commands control basic system functionality like:

- RS485 configuration
- Temperature monitoring
- Audio
- Timecode
- Status LEDs
- ...

They are called with the `system` command prefix.

7.3.1 system info

Command	<code>system info</code>		
Alias	<code>version</code>	Getter Function	No
Description	Get system information. If called via the alias <code>version</code> this command prints the information in a style that can be parsed by ProVideo protocol compatible controllers.		

Example Output:

```
platform           : vega
device name        : Proton Cam
serial number      : DF-61-9C-58-CB-66-51-2A
video processor version : v1.0-253
software version    : v1.0.0+0
resolution mask     : 00000000-000007FF-00000000
```

Example Output when called via the `version` alias:

```
platform           : vega
device name        : Proton Cam
system-id          : DF-61-9C-58-CB-66-51-2A
hw revision        : 0001000000000253
system validity: LICENSED
feature mask HW: 00000000
feature mask SW: 00000000
```



```
resolution mask: 00000000-000007FF-00000000
loader version : Unknown
sw-release-id   : v1.0.0+0
sw-release-date: Unknown
sw-build-date  : Unknown
```

As you can see most of the extra fields in the ProVideo compatible output are blank as they are not used / supported by PROTON OS.

Notes on the provided information:

- **platform:** The platform string is unique for each PROTON device. It can be used by a controller to determine which device type it is talking to. It is also included in the `system identify` command.
- **device name:** Can be set by the user with the `system name` command.
- **serial number:** Unique device serial number.
- **video processor version:** Version of the programmable video processor.
- **software version:** PROTON OS version.
- **resolution mask:** Shortened list of all supported video modes where each bit represents one mode, see table below.

Table 5: Resolution Mask.

Bit	HD Modes (Left Block)	FHD / 2K Modes (Middle Block)	UHD / 4K Modes (Right Block)
1	720p60	1080p30	UHDp30
2	720p50	1080p25	UHDp25
3	720p59	1080p24	UHDp24
4		1080p23	UHDp23
5		1080p29	UHDp29
6		1080p50	UHDp50
7		1080p60	UHDp60
8		1080i60	UHDp59
9		1080i50	UHDp48
10		1080i59	UHDp47
11		1080p59	4Kp30
12		2Kp30	4Kp25
13		2Kp25	4Kp24
14		2Kp24	4Kp23
15		2Kp23	4Kp29
16		2Kp29	4Kp50
17		2Kp50	4Kp60
18		2Kp60	4Kp59
19		2Kp59	4Kp48
20		2Kp48	4Kp47
21		2Kp47	

Note: A controller may use the resolution mask to determine which modes are supported by a camera. Alternatively use the `video mode list` command to list all supported video modes.

7.3.2 system name

Command	system name <name string>		
Alias	name	Getter Function	Yes
Description	Set device name. The name string may contain up to 5 words which in total and including white spaces have a length of 32 characters.		
Parameter	name string		
Type	String		
Description	Device name to set, max 32 characters.		

7.3.3 system runtime

Command	system runtime		
Alias	runtime	Getter Function	Yes
Description	Print device runtime since boot in seconds. Counter gets reset by a power cycle or reboot.		

7.3.4 system reboot

Command	system reboot		
Alias	reboot	Getter Function	No
Description	Reboot the device.		

7.3.5 system update

Command	system update		
Alias	update	Getter Function	No
Description	Put device into firmware update mode. In update mode the device will respond to firmware update requests. Additionally, video processing is stopped to increase the performance of the update process. The device will respond to commands as usual, but video and camera related commands will fail. Firmware update mode can only be left via a reboot.		

7.3.6 system identify

Command	system identify		
Alias	identify	Getter Function	No
Description	Print essential system information (platform, RS485 configuration, device name) with a delay depending on the RS485 device address. Can be used to quickly identify all devices on an RS485 bus by sending it to the fail-safe address 100. Maximum delay: $99 * 10\text{ms} \approx 1\text{s}$ (99 is the maximum RS485 address). Output: id <platform> <dev_addr> <bc_addr> <is_master> <dev_name>		

Example for an RS485 bus with 3 cameras of which the cameras with the device IDs 20 and 21 are part of broadcast group 2 and camera 20 is the broadcast master (has the `is_master` flag set):

```

→ 100 system identify
← id: vega 1 0 0 Camera Left   After 10ms
← OK
← id: vega 20 2 1 Wide Angle   After 200ms
← OK
← id: vega 21 2 0 Top View     After 210ms
← OK
  
```

7.3.7 system error

Command	system error		
Alias	error	Getter Function	No
Description	Print error log. In case no errors are logged it only returns OK. The status LED blinks red when an error was logged.		

7.3.8 system volatile

Command	system volatile <value>		
Alias	volatile	Getter Function	Yes
Description	Set a 32 bit runtime variable which will keep its value until a reboot is performed. Can be used by a controller to store arbitrary information or check if camera got rebooted (volatile value got reset to 0).		
Parameter	value		
Description	Volatile value to set.		
Min	0		

Max	4294967295 = 0xFFFFFFFF
Default	0

7.3.9 system rs485

Command	system rs485		
Alias	rs485	Getter Function	No
Description	Setup RS485 console interface, see subcommands for details. If called without arguments prints a summary of the current settings.		

7.3.9.1 system rs485 mode

Command	system rs485 mode <mode>		
Alias	prompt	Getter Function	Yes
Description	Set RS485 operational mode: 0 = Controller Mode: Shell addressing is enabled, to allow multiple devices on the same RS485 bus. Prompt and echo are disabled. 1 = Interactive Mode: Shell addressing is enabled, to allow multiple devices on the same RS485 bus. Prompt and echo are disabled.		
Parameter	mode		
Type	Unsigned Integer		
Valid Values	0 = Controller Mode 1 = Interactive Mode		
Alias	controller → system rs485 mode 0 interactive → system rs485 mode 1		
Default	0 = Controller Mode		

7.3.9.2 system rs485 device_address

Command	system rs485 device_address <dev_addr>		
Alias	rs485_addr	Getter Function	Yes
Description	Set RS485 device address. Cannot be identical to broadcast address. The address 100 is the fail-save address, the device will always respond to that address.		
Parameter	dev_addr		
Description	Device address to set.		
Min	0		
Max	99		
Default	1		

7.3.9.3 system rs485 broadcast_address

Command	system rs485 broadcast_address <bc_addr>		
Alias	rs485_bc_addr	Getter Function	Yes
Description	Set RS485 broadcast group address. Cannot be identical to the device address. The device will process commands received on the broadcast address but not reply to them unless it is the broadcast master.		
Parameter	bc_addr		
Description	Broadcast address to set.		
Min	0		
Max	99		
Default	0		

7.3.9.4 system rs485 broadcast_master

Command	system rs485 broadcast_master <dev_addr>		
Alias	rs485_bc_master	Getter Function	Yes
Description	Set RS485 broadcast master. The device with the given address becomes the broadcast master. Should be send to a broadcast group to ensure there is only one active master. Only the master will reply to commands received on the broadcast address, all other devices are silent. To disable the broadcast master, use the special device address -1.		

	When called without arguments this command will not return the address of the current broadcast master but a flag indicating if this device is currently the master.
Parameter	dev_addr
Description	Device address of the device which shall become master of the broadcast group.
Min	-1 (disable broadcast master)
Max	99
Default	-1 (disabled)

For an example see chapter 5.1.1.2.

7.3.9.5 system rs485 baudrate

Command	system rs485 baudrate <rate>		
Alias	rs485_baud	Getter Function	Yes
Description	Set RS485 baud rate. To get supported rates use the baudrate list subcommand.		
Parameter	rate		
Description	New RS485 baud rate to set.		
Valid Values	9600, 1440, 19200, 57600, 115200, 230400, 250000		
Default	115200		

7.3.9.5.1 system rs485 baudrate list

Command	system rs485 baudrate list		
Alias	/	Getter Function	No
Description	List all supported baud rates of the RS485 interface.		

Example:

```
→ 1 system rs485 baudrate list
← 9600
← ...
← 115200
← OK
```

7.3.10 system status_led

Command	system status_led <brightness>		
Alias	status_led	Getter Function	Yes
Description	Set brightness off the status LED from 0 to 10, set to 0 to turn it off.		
Parameter	brightness		
Description	Status LED brightness.		
Min	0 (off)		
Max	10 (100%)		
Default	10 (100%)		

7.3.11 system tally

Command	system tally <brightness>		
Alias	status_led	Getter Function	Yes
Description	Set brightness off the tally light in %, set to 0 to turn it off. The tally light is only controlled by the user, it is not turned on automatically.		
Parameter	brightness		
Description	Tally light brightness in %.		
Min	0 (off)		
Max	100 (100%)		
Default	0 (off)		

7.3.12 system temp

Command	system temp (<sensor_idx>)		
Alias	temp	Getter Function	Yes
Description	List all supported temperature measurements in °C, the number of temperature sensors varies between camera models.		

	To get a single temperature value specify the optional <code>sensor_idx</code> parameter. For temperature logging see subcommands. Output: <code><sensor_idx> <temperature> <description></code>
Parameter	<code>sensor_idx</code>
Description	Optional index of the sensor
Min	0 (off)
Max	100 (100%)

Examples:

```
→ 1 system temp
← system temp 0 63.5 CPU
← system temp 1 57.8 Case
← OK
→ 1 system temp 0
← system temp 0 67.2 CPU
← OK
```

7.3.12.1 system temp max

Command	<code>system temp max</code>		
Alias	<code>max_temp</code>	Getter Function	Yes
Description	Show maximum logged system temperature. It prints the following values: <code>max_temp_user</code> : User resettable maximum temperature in °C since boot. It can be reset with the 'temp reset' command or via a power-cycle or reboot. <code>max_temp</code> : Maximum temperature in °C since boot which can only be reset via power-cycle or reboot. <code>shutdown_temp</code> : Shutdown temperature in °C. When the shutdown temperature is reached, the device enters a safe state and tries to cool down. Video processing is restarted when it is cooled sufficiently. Whenever the shutdown temperature is reached an over temperature event is logged (see <code>temp count</code> command). Output: <code><max_temp_user> <max_temp> <shutdown_temp></code>		

7.3.12.2 system temp reset

Command	<code>system temp reset</code>		
Alias	<code>max_temp_reset</code>	Getter Function	No
Description	Reset the user resettable maximum logged temperature (see <code>temp max</code> command) to the current system temperature. The temperature is also reset when the camera is power-cycled or rebooted.		

7.3.12.3 system temp count

Command	<code>system temp count</code>		
Alias	<code>over_temp_count</code>	Getter Function	Yes
Description	Show number of logged over temperature events. This is the number of times the device had to shut down because the temperature reached the shutdown temperature (see <code>temp max</code> command). This counter is persistent and cannot be reset. Output: <code><over_temp_count></code>		

7.3.13 system audio

Command	<code>system audio <enable></code>		
Alias	<code>audio_enable</code>	Getter Function	Yes
Description	Enable stereo audio. If enabled 2 channels of audio will be embedded into the SDI signal. To change audio volume, see subcommands.		
Parameter	<code>enable</code>		
Type	Boolean		
Description	Enable or disable SDI audio.		
Default	On		

7.3.13.1 system audio gain

Command	<code>system audio gain <factor></code>
----------------	---

Alias	audio_gain	Getter Function	Yes
Description	Set audio gain. Gain is given as a 4.12 fixed point number, range [0.0, 16.0) with a default of 1.0 = 4096. Setting a gain of 0 will mute (but not disable) the audio output.		
Parameter	factor		
Type	Unsigned 4.12 Fixed Point Number		
Description	Audio gain factor to set.		
Min	0.0 = 0		
Max	15.999 = 65535		
Default	0.5 = 2048		

7.3.14 system timecode

Command	system timecode <enable>		
Alias	timecode_enable	Getter Function	Yes
Description	Enable SDI time code insertion. Disabling the time code resets the value to 0. To set or pause the time code, see subcommands.		
Parameter	enable		
Type	Boolean		
Description	Enable or disable SDI time code.		
Default	On		

7.3.14.1 system timecode value

Command	system timecode value <hour> <minute> <second>		
Alias	timecode	Getter Function	Yes
Description	Set time code to given value or get current value. New value is applied with the next frame. If time code insertion is currently disabled, setting a value has no effect and the new value is ignored. When setting the time code, the frame counter is always reset to 0.		
Parameter	hour	minute	second
Type	Unsigned Integer	Unsigned Integer	Unsigned Integer
Description	Hour value.	Minute value.	Second value.
Min	0	0	0
Max	23	59	59

7.3.14.2 system timecode pause

Command	system timecode pause <enable>		
Alias	timecode_hold	Getter Function	Yes
Description	Pause time code. While paused the transmitted time code is no longer incremented which can be used to control an external SDI recorder. Internally the counter keeps running so when pause is released the transmitted time code value jumps to the current time. This setting is not stored in the persistent storage, so pause is always disabled when the camera is powered on.		
Parameter	enable		
Type	Boolean		
Description	Pause timecode at current value.		
Default	Disabled		

7.3.14.3 system timecode drop

Command	system timecode drop <enable>		
Alias	timecode_drop	Getter Function	Yes
Description	Enable 'drop frame' mode for fractional video modes. This feature is enabled by default and causes the camera to occasionally skip frame numbers in the frame counter for fractional video modes to ensure that the time runs as close to a real clock as possible. This setting has no effect for non-fractional (integer) video modes.		
Parameter	enable		
Type	Boolean		

Description	Use 'drop frame' mode for fractional video modes.
Default	Enabled

7.4 Camera Commands

These commands control the image sensor (gain and exposure). They are called via the `camera` command prefix.

7.4.1 camera gain

Command	camera gain <value>		
Alias	cam_gain	Getter Function	Yes
Description	Set camera gain as (linear gain * 1000), use <code>info</code> command to get range. To set gain to minimum or maximum value pass <code>min</code> or <code>max</code> as value.		
Parameter	value		
Type	Unsigned Integer		
Description	Linear gain scaled by 1000 (see examples below).		
Min	1.0 = 1000		
Max	Depends on camera model, use <code>camera info</code> command to get range.		
Default	1.0 = 1000		

This command sets the linear gain scaled by 1000 for fine-grained gain control. Examples:

```
→ 1 camera gain 1000    # Set gain to 1.0
← OK
→ 1 camera gain 5237    # Set gain to 5.237
← OK
```

To convert linear gain to ISO, use the `camera info` command to get the ISO at gain 1.0 and then simply multiply that value with the gain:

$$gain_{ISO} = ISO_{1.0} * gain_{linear}$$

7.4.2 camera exposure

Command	camera exposure <value>		
Alias	cam_exposure	Getter Function	Yes
Description	Set camera exposure time in microseconds, use <code>info</code> command to get range. To set exposure to minimum or maximum value pass <code>min</code> or <code>max</code> as value.		
Parameter	value		
Type	Unsigned Integer		
Description	Exposure time in μ s.		
Min	Depends on camera model, use <code>camera info</code> command to get range.		
Max	Depends on video mode, use <code>camera info</code> command to get range.		
Default	Maximum value for default video mode.		

7.4.3 camera gain_mode

Command	camera gain_mode <mode>		
Alias	gain_mode	Getter Function	Yes
Description	Switch gain mode to adjust the cameras base sensitivity to the light conditions. This changes the ISO value at gain 1.0 and possibly also the allowed gain range. The updated values can be read using the <code>camera info</code> command. To list supported modes, use the <code>gain_mode list</code> subcommand.		
Parameter	mode		
Type	Unsigned Integer		
Description	Gain mode to set.		
Valid Values	0 = Standard 1 = Low-Light		
Default	0 = Standard		

By default, the camera is configured in Standard mode which results in a good gain range for operation in normally lit environments. For dark environments Low-Light mode can be enabled to boost the base gain of the image sensor (if supported by the camera model).

7.4.3.1 camera gain_mode list

Command	camera gain_mode list		
Alias	/	Getter Function	No
Description	List all supported gain modes.		

The supported modes may vary between camera models, so a controller should check which modes are supported.

Each mode is printed in a separate line. Example:

```
→ 1 camera gain_mode list
← 0
← 1
← OK
```

7.4.4 camera info

Command	camera info		
Alias	cam info	Getter Function	Yes
Description	Prints the capabilities of the image sensor: gain_min: Minimum gain setting, fixed. gain_max: Maximum gain setting, fixed, may depend on gain_mode setting. exp_min: Minimum exposure setting, fixed. exp_max: Maximum exposure setting, depends on video mode. base_iso: ISO at gain 1.0 (= 1000), depends on gain_mode setting. Output: <gain_min> <gain_max> <exp_min> <exp_max> <base_iso>		

7.4.5 camera auto

Command	camera auto <enable>		
Alias	aec	Getter Function	Yes
Description	Enable or disable automatic exposure control. For setup of the auto exposure mode and other parameters see subcommands. Settings that are currently controlled by the auto exposure (e.g. gain or exposure) cannot be set manually, but you can use their getter functions to check the value that have been applied by the AEC. When called via the aec alias the command behaves according to the ProVideo protocol that means either 1 parameter (enable) or 10 parameters can be passed (see below). The clm_tolerance and cost_apr settings from the ProVideo protocol are not supported and the passed values are ignored. For gain control set cost_gain > 0 and cost_tint = 0, for exposure control vice versa. If both are enabled or enabled combined gain and exposure control is used. For detailed examples, see below.		
Parameter	enable		
Type	Boolean		
Description	Enable or disable automatic exposure control.		
Default	Enabled		

When called via the full syntax the command is only used to enable or disable the AEC:

Command: camera auto <enable>

```
→ 1 camera auto 1
← OK
```

For all other settings, see the subcommands in the following sections.

ProVideo Protocol Compatibility

For ProVideo protocol compatibility the command can also be called via the `aec` alias. In that mode it either supports 1 or 10 parameters. With 1 parameter the behavior is identical to the `camera auto` command and only the enable state is changed:

Command: `aec <enable>`

```
→ 1 aec 1
← OK
```

When called with 10 parameters the ProVideo protocol settings are converted to PROTON OS settings:

Command: `aec <enable> <set_point> <speed> <clm_tolerance> <const_gain> <cost_tint> <cost_apt> <taf> <max_gain> <use_custom_weights>`

ProVideo Paramater	Value Range	Related PROTON OS Command
enable	[0, 1]	<code>camera auto</code>
set_point	[256, 3000]	<code>camera auto target</code> , range gets converted to [0, 1000]
speed	[3, 30]	<code>camera auto speed</code> , range gets converted to [0, 100]
clm_tolerance	[10, 500]	None, value is ignored, getter always returns 50
cost_gain	0 + [250, 8000]	If set greater 0: <code>camera auto mode 0</code> , getter returns 0 when disabled and 8000 when enabled.
cost_exp	0 + [250, 8000]	If set greater 0: <code>camera auto mode 1</code> , getter returns 0 when disabled and 8000 when enabled.
cost_apt	0 + [250, 8000]	None, value is ignored, getter always returns 0
taf	[5000, 20000]	<code>camera auto anti_flicker</code> , if set to 10000: 50 Hz anti-flicker is used, if set to 8333: 60 Hz anti-flicker is used, for all other settings anti-flicker is disabled.
max_gain	Camera specific	<code>camera auto max_gain</code>
use_custom_weights	[0, 1]	<code>camera auto custom</code>

Note: If both `cost_gain` and `cost_exp` are set to 0, or both are enabled (>0), gain control mode is used.

The following example shows the returned default settings:

```
→ 1 aec
← 1 1170 17 50 8000 8000 0 0 32000 0
← OK
```

Example to enable AEC in exposure control mode:

```
→ 1 aec 0 1170 17 50 0 8000 0 0 32000 0
← OK
```

7.4.5.1 camera auto mode

Command	<code>camera auto mode <mode></code>		
Alias	<code>aec_mode</code>	Getter Function	Yes
Description	Set auto exposure mode. To get supported modes use the <code>mode list</code> subcommand.		
Parameter	<code>mode</code>		
Type	Unsigned Integer		
Description	Auto exposure mode to set.		
Valid Values	0 = Gain Control: Gain is set automatically; exposure can be set manually using the <code>camera exposure</code> command. The maximum gain used by the algorithm can be limited using the <code>camera auto max_gain</code> command. 1 = Exposure Time Control: Exposure time is set automatically; gain is set manually using the <code>camera gain</code> command.		

	2 = Combined Control: Both gain and exposure time are set automatically. The algorithm prefers increasing the exposure and tries to minimize the gain to reduce noise in the output image. The maximum gain used by the algorithm can be limited using the <code>camera auto max_gain</code> command. In this mode anti-flicker can be enabled with the <code>camera auto anti_flicker</code> command to limit the exposure time to flicker free values when using artificial light sources.		
Default	0 = gain control		

7.4.5.1.1 camera auto mode list

Command	<code>camera auto mode list</code>		
Alias	/	Getter Function	No
Description	List all auto exposure modes.		

This command is mainly intended for interactive use. A controller can expect all modes to be available for every camera model.

7.4.5.2 camera auto target

Command	<code>camera auto target <brightness></code>		
Alias	<code>aec_target</code>	Getter Function	Yes
Description	Set target brightness for the auto exposure algorithm. Valid range [0, 1000] which equals [0.0, 1.0] or [0%, 100%]. The default is 333 = 0.333 = 33.3%.		
Parameter	<code>brightness</code>		
Type	Unsigned Integer		
Description	Relative brightness in percent multiplied by 10 for higher resolution.		
Min	0 = 0%		
Max	1000 = 100%		
Default	333 = 33.3%		

7.4.5.3 camera auto speed

Command	<code>camera auto speed <speed></code>		
Alias	<code>aec_speed</code>	Getter Function	Yes
Description	Set auto exposure control speed in range [1, 100] which equals [1%, 100%]. Bigger values result in a faster reaction to scene brightness changes. The default is 50 = 50%.		
Parameter	<code>speed</code>		
Type	Unsigned Integer		
Description	AEC control speed (reaction speed to brightness changes).		
Min	0 = 0%		
Max	100 = 100%		
Default	50 = 50%		

7.4.5.4 camera auto max_gain

Command	<code>camera auto max_gain <value></code>		
Alias	<code>aec_max_gain</code>	Getter Function	Yes
Description	Set the maximum gain that the AEC algorithm is allowed to use. The gain range is identical to the <code>camera gain</code> command and can be fetched with the <code>camera info</code> command. To set the maximum gain to the minimum or maximum value pass <code>min</code> or <code>max</code> as value.		
Parameter	<code>value</code>		
Type	Unsigned Integer		
Description	Maximum linear gain that the AEC algorithm is allowed to use.		
Min	1.0 = 1000		
Max	Depends on camera model, use <code>camera info</code> command to get range.		
Default	Maximum gain.		

7.4.5.5 camera auto anti_flicker

Command	<code>camera auto anti_flicker <mode></code>		
Alias	<code>aec_anti_flicker</code>	Getter Function	Yes
Description	Set the anti-flicker mode for the combined (gain and exposure) auto exposure mode. In all other modes this setting has no effect.		

	When anti-flicker is enabled the auto exposure algorithm tries to use only exposure times that allow for a flicker free output image when using artificial light sources. To get supported modes use the <code>anti_flicker_list</code> subcommand.
Parameter	<code>mode</code>
Type	Unsigned Integer
Description	Auto exposure anti-flicker mode to set.
Valid Values	0 = Disabled 1 = 50 Hz (EU) 2 = 60 Hz (US)
Default	0 = Disabled

7.4.5.5.1 camera auto anti_flicker list

Command	<code>camera auto anti_flicker list</code>		
Alias	/	Getter Function	No
Description	List all auto exposure anti-flicker modes.		

This command is mainly intended for interactive use. A controller can expect all modes to be available for every camera model.

7.4.5.6 camera auto custom

Command	<code>camera auto custom <enable></code>		
Alias	<code>aec_custom</code>	Getter Function	Yes
Description	Enable to use custom weights for the 25 measurement areas of the auto exposure algorithm. Weights can be configured with the <code>auto_weight</code> command. If disabled (default), all measurements are weighted equally with a weight of 1.		
Parameter	<code>enable</code>		
Type	Boolean		
Description	Enable or disable usage of custom weights for the measurement areas.		
Default	Disabled		

7.4.5.7 camera auto weight

Command	camera auto weight <index> (<weight>)																												
Alias	aec_weight		Getter Function	Yes																									
Description	<p>The AEC algorithm measures the brightness in an evenly split 5x5 grid. Each area of the grid has a weight which determines how much this area shall influence the algorithm. A higher weight results in a stronger focus on that area, a weight of 0 completely ignores the measurement for that area.</p> <p>The index parameter selects the measurement area for which the weight is set. Index 1 is the top-left measurement area while 25 is bottom-right:</p> <table><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr><tr><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr><tr><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr><tr><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td></tr><tr><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td></tr></table> <p>When the command is called with the index parameter only, the weight for that index is printed. If it is called without parameters all weights are printed.</p> <p>These weights are only used if custom weights are enabled with the <code>auto custom</code> command, otherwise all areas are weighted equally.</p>				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	2	3	4	5																									
6	7	8	9	10																									
11	12	13	14	15																									
16	17	18	19	20																									
21	22	23	24	25																									
Parameter	index		weight																										
Type	Unsigned Integer		Unsigned Integer																										
Description	Index of the weight to set.		Weight for selected measurement window.																										
Min	1		0 = Ignore this area																										
Max	25		25 = Very strong focus on this area																										
Default	/		1																										

7.5 Video Commands

These commands control the video processing features of the camera. They are called with the `video` command prefix.

7.5.1 video mode

Command	video mode <mode>		
Alias	video_mode	Getter Function	Yes
Description	Set image format of the video pipeline. This will stop video processing, reconfigure for the new format and restart processing. To list supported modes, use the <code>mode list</code> subcommand or evaluate the resolution mask from the <code>system info</code> command. If the current exposure setting is bigger than the maximum of the new mode it will be clipped.		
Parameter	mode		
Description	Video mode ID.		
Valid Values	Depends on camera model, see Table 6 below.		
Default			

The following table lists the supported video modes and the default video mode for each camera model:

Table 6: Supported Video Modes.

ID	Mode	PROTON CAM
4	1080p30	✓
5	1080p25	✓
6	1080p24	✓
7	1080p23	✓
8	1080p29	✓
9	1080p50	✓ Default
10	1080p60	✓
11	1080i60	✓
12	1080i50	✓
13	1080i59	✓
14	1080p59	✓

7.5.1.1 video mode list

Command	video mode list		
Alias	/	Getter Function	No
Description	List all supported video modes.		

This command can be used by a controller to check which video modes are supported by the device (instead of evaluating the resolution mask from the `system info` command or hardcoding the values).

Each mode is printed in a separate line. Example:

```

→ 1 video mode list
← 4
← 5
← ...
← 14
← OK
  
```

7.5.2 video flip

Command	video flip <mode>		
Alias	flip	Getter Function	Yes
Description	Flip output image. To list all modes, use the <code>flip list</code> subcommand.		
Parameter	mode		
Type	Unsigned Integer		
Description	Flip mode to set.		
Valid Values	0 = Normal 1 = Vertical 2 = Horizontal 3 = Rotated (H+V)		
Default	0 = Normal		

7.5.2.1 video flip list

Command	video flip list		
Alias	/	Getter Function	No
Description	List all flip modes.		

This command is mainly intended for interactive use. A controller can expect all modes to be available for every camera model.

7.5.3 video black_sensor

Command	video black_sensor <red/all> (<green> <blue>)		
Alias	black_sensor	Getter Function	Yes
Description	Set all sensor black level offsets. All values are 17 bit signed integers with a range of [-65536, 65535]. The offsets are subtracted, that means a negative offset results in an addition. Usually, positive values will be programmed to subtract the sensor black level from the image signal. If only one value is given, all offsets are set to the same value. Otherwise, all three values must be given. Changing the sensor black level is usually not recommended, use the flare compensation or master black level instead. To set a single offset value use the subcommands.		
Parameter	red/all	green	blue
Type	Signed Integer		
Description	Red or all components black level offset.	Optional green offset.	Optional blue offset.
Min	-65536		
Max	65535		
Default	0		

7.5.3.1 video black_sensor red

Command	video black_sensor red <offset>		
Alias	black_red	Getter Function	Yes
Description	Set red black level offset as 17 bit signed integer.		
Parameter	factor		
Type	Signed Integer		
Description	Red component black level offset.		
Min	-65536		
Max	65535		
Default	0		

7.5.3.2 video black_sensor green

Command	video black_sensor green <offset>		
Alias	black_green	Getter Function	Yes
Description	Set green black level offset as 17 bit signed integer.		
Parameter	factor		
Type	Signed Integer		
Description	Green component black level offset.		
Min	-65536		
Max	65535		
Default	0		

7.5.3.3 video black_sensor blue

Command	video black_sensor blue <offset>		
Alias	black_blue	Getter Function	Yes
Description	Set blue black level offset as 17 bit signed integer.		
Parameter	factor		
Type	Signed Integer		
Description	Blue component black level offset.		
Min	-65536		

Max	65535
Default	0

7.5.4 video lsc

Command	video lsc <enable> (<k> <offset> <slope>)			
Alias	lsc		Getter Function	Yes
Description	Configure lens shading correction. The parameters k, offset and slope are optional and given as 2.30 fixed point numbers. If any of the 3 parameters shall be changed, all 3 must be specified! For details on how to calibrate LSC for a specific lens see below. To set a preset for an officially supported lens use the <code>lsc preset</code> subcommand.			
Parameter	enable	k	offset	slope
Type	Boolean	Unsigned 2.30 Fixed Point Number		
Description	Enable or disable correction.	Natural vignetting correction factor.	Artificial vignetting correction offset.	Artificial vignetting correction slope.
Min	0 (Off)	0.0 = 0	0.0 = 0	0.0 = 0
Max	1 (On)	2.0 = 2147483648	1.0= 1073741824	2.0 = 2147483648
Default	Off	0.0 = 0	0.0 = 0	0.0 = 0

Due to the physical properties of the lenses used in optical imaging a reduction of the luminescence occurs from the middle of the image sensor to its borders. This is commonly known as vignetting. This effect can be separated into natural and artificial vignetting, both can be corrected using the lens shading correction function.

Please note that this is an advanced feature that requires the right measurement equipment to be setup correctly (see setup instructions at the end of this chapter). The lens shading correction uses a correction function to increase the gain in the outer image areas. The parameter `k` configures the natural vignetting compensation (cos4 compensation), a higher value will result in a higher compensation. The parameters `offset` and `slope` configure the artificial vignetting compensation. If the value for `offset` is increased, the radius where the compensation starts will be increased (moved to the image border). The higher the `slope`, the stronger is the compensation.

To setup the lens shading correction, follow these steps:

1. Point the camera at a homogeneous light source.
2. Connect the camera to a wave monitor and select a line in the middle of the image, you should see a decrease in luminescence towards the edges of the image.
3. Enable lens shading correction, start with the parameters `k`, `offset` and `slope` set to 0.
4. Now turn up the `k` factor, this should correct the lens shading in the middle area of the image, the edges will probably still be not ideally illuminated. Make sure to not overcompensate, this will create a wavelike appearance of the luminescence on the monitor.
5. Set the `offset` to 0.5 and set to `slope` to a high value like 1.5, you should now clearly see where the compensation starts. Now adjust the `slope` until you hit the point where the luminescence starts decreasing. Finally decrease the `slope` until the result is not overcompensated anymore.
6. Make fine adjustments until you are satisfied with the result. Please note that, depending on the optical lens used, the aperture and focal length have influence on the lens shading, so you should use your default setup for configuring the compensation. Also, it might be helpful to turn of the auto exposure during the setup, see `camera auto` command.

Each PROTON camera comes pre-calibrated for officially supported lenses. To select a calibrated lens preset, use the `lsc preset` command described below.

7.5.4.1 video lsc preset

Command	video lsc preset <idx>		
Alias	lsc_preset	Getter Function	Yes

Description	Apply preset for selected lens. Setting a preset automatically enables the lens shade correction. Use the <code>preset list</code> subcommand to show all supported lenses and their corresponding indices. When called as a getter the index of the currently configured preset is returned. If a custom LSC configuration is used -1 is returned.
Parameter	<code>idx</code>
Type	Unsigned Integer
Description	LSC preset to set.
Valid Values	Depend on camera model, use <code>preset list</code> subcommand to get list.
Default	-1 (no preset configured)

7.5.4.1.1 video lsc preset list

Command	<code>video lsc preset list</code>		
Alias	<code>/</code>	Getter Function	No
Description	List all available LSC presets.		

The available presets depend on the camera model and new firmware versions may add new calibration presets to the list. It is guaranteed that the index of a preset does not change between firmware releases (that means new entries are always appended to the list, the list is not reordered).

Each entry of the list consists of the preset's index followed by:

- Horizontal angle of aperture in degree
- Focal length in mm
- Aperture in F-stops
- Length of the lens in mm
- The PROTON reference number.

The following example shows the supported presets for a PROTON CAM with firmware v1.1.1:

```

→ 1 video lsc preset list
← 0 - 124 degree: 2.2mm, F/2.2, 24mm length (PCI-LENS-22-22)
← 1 - 110 degree: 2.7mm, F/2.8, 22mm length (PCI-LENS-27-28)
← 2 - 97 degree: 3.2mm, F/2.3, 22.5mm length (PCI-LENS-32-23)
← 3 - 88 degree: 3.9mm, F/2.8, 22.2mm length (PCI-LENS-39-28)
← 4 - 85 degree: 4.2mm, F/1.9, 30mm length (PCI-LENS-42-19)
← 5 - 68 degree: 6mm, F/1.7, 31.3mm length (PCI-LENS-60-17)
← 6 - 52 degree: 7.8mm, F/2.0, 34.4mm length (PCI-LENS-78-20)
← 7 - 35 degree: 12mm, F/2.0, 24mm length (PCI-LENS-120-20)
← OK

```

7.5.5 video wb

Command	<code>video wb</code>		
Alias	<code>wb</code>	Getter Function	No
Description	Run auto white balance once and adjust color temperature at the next frame end. Has no effect if continuous auto white balance is currently enabled. This resets gain and color cross matrix offsets to 0. For manual white balance and configuration of the auto white balance, see subcommands.		

7.5.5.1 video wb gain

Command	<code>video wb gain <red> <green> <blue></code>		
Alias	<code>wb_gain</code>	Getter Function	Yes
Description	Set RGB white balance gains. All values are in 4.8 fixed point format with a range of [0.0 = 0, 15.999 = 4095] (1.0 = 256). The gain values will be overridden when a color temperature or white balance preset is set. To set a single gain value use the subcommands.		

	When auto white balance is enabled, this command returns the currently configured gains and setting it is not possible. Instead of specifying the absolute gains you can also use the <code>gain offset</code> subcommand to specify offsets which are applied on top of the gains from the selected color temperature or preset.		
Parameter	red	green	blue
Type	Unsigned 4.8 Fixed Point Number		
Description	Red gain.	Green gain.	Blue gain.
Min	0.0 = 0		
Max	15.999 = 4095		
Default	Values for color temperature 6500K, depends on camera calibration.		

7.5.5.1.1 video wb gain red

Command	video wb gain red <gain>		
Alias	gain_red	Getter Function	Yes
Description	Set red white balance gain in 4.8 fixed point format (1.0 = 256).		
Parameter	gain		
Type	Unsigned 4.8 Fixed Point Number		
Description	Red white balance gain.		
Min	0.0 = 0		
Max	15.999 = 4095		
Default	Value for color temperature 6500K, depends on camera calibration.		

7.5.5.1.2 video wb gain green

Command	video wb gain green <gain>		
Alias	gain_green	Getter Function	Yes
Description	Set green white balance gain in 4.8 fixed point format (1.0 = 256).		
Parameter	gain		
Type	Unsigned 4.8 Fixed Point Number		
Description	Green white balance gain.		
Min	0.0 = 0		
Max	15.999 = 4095		
Default	Value for color temperature 6500K, depends on camera calibration.		

7.5.5.1.3 video wb gain blue

Command	video wb gain blue <gain>		
Alias	gain_blue	Getter Function	Yes
Description	Set blue white balance gain in 4.8 fixed point format (1.0 = 256).		
Parameter	gain		
Type	Unsigned 4.8 Fixed Point Number		
Description	Blue white balance gain.		
Min	0.0 = 0		
Max	15.999 = 4095		
Default	Value for color temperature 6500K, depends on camera calibration.		

7.5.5.1.4 video wb gain offset

Command	video wb gain offset <red> <green> <blue>		
Alias	wb_gain_offset	Getter Function	Yes
Description	Fine tune the RGB white balance gains by applying offsets. The offsets are added to the gain values which result from the selected white balance temperature or preset. All values are in signed 5.8 fixed point format with a range of [-15.999 = -4095, 15.999 = 4095] (0.0 = 0). To use the original (calibrated) values, set all offsets to the default value of 0. The offsets are reset to 0 when a preset is selected using the <code>wb preset</code> command or if the auto white balance is run. The offsets cannot be changed while auto white balance is enabled. To set offsets individually use the subcommands.		
Parameter	red	green	blue
Type	Signed 5.8 Fixed Point Number		
Description	Red gain.	Green gain.	Blue gain.

Min	-15.999 = -4095
Max	15.999 = 4095
Default	0.0 = 0

7.5.5.1.4.1 video wb gain offset red

Command	video wb gain offset red <offset>		
Alias	gain_red_offset	Getter Function	Yes
Description	Set red white balance gain offset in signed 5.8 fixed point format.		
Parameter	gain		
Type	Signed 5.8 Fixed Point Number		
Description	Red white balance gain offset.		
Min	-15.999 = -4095		
Max	15.999 = 4095		
Default	0.0 = 0		

7.5.5.1.4.2 video wb gain offset green

Command	video wb gain offset green <offset>		
Alias	gain_green_offset	Getter Function	Yes
Description	Set green white balance gain offset in signed 5.8 fixed point format.		
Parameter	gain		
Type	Signed 5.8 Fixed Point Number		
Description	Green white balance gain offset.		
Min	-15.999 = -4095		
Max	15.999 = 4095		
Default	0.0 = 0		

7.5.5.1.4.3 video wb gain offset blue

Command	video wb gain offset blue <offset>		
Alias	gain_blue_offset	Getter Function	Yes
Description	Set blue white balance gain offset in signed 5.8 fixed point format.		
Parameter	gain		
Type	Signed 5.8 Fixed Point Number		
Description	Blue white balance gain offset.		
Min	-15.999 = -4095		
Max	15.999 = 4095		
Default	0.0 = 0		

7.5.5.2 video wb color_cross

Command	video wb color_cross <c0> ... <c8>		
Alias	color_cross	Getter Function	Yes
Description	<p>Manually set color cross-talk matrix. The matrix consists of 9 coefficients which are in signed 4.12 fixed point format with a range of [-8.0 = -32768, 8.0 = 32767] (1.0 = 4096).</p> <p>Usually, the matrix is set automatically by selecting a white balance temperature or preset.</p> <p>When auto white balance is enabled, this command returns the currently configured matrix and setting it is not possible.</p> <p>Instead of specifying the absolute matrix coefficients you can also use the <code>color_cross offset</code> subcommand to specify offsets which are applied on top of the matrix from the selected color temperature or preset.</p>		
Parameter	c0 ... c8		
Type	Signed 4.12 Fixed Point Number		
Description	Color cross-talk matrix coefficients.		
Min	-8.0 = -32768		
Max	7.999 = 32767		
Default	Values for color temperature 6500K, depends on camera calibration.		

The cross-talk matrix is applied in RGB domain:

$$red_{out} = red_{in} * c_0 + green_{in} * c_1 + blue_{in} * c_2$$

$$green_{out} = red_{in} * c_3 + green_{in} * c_4 + blue_{in} * c_5$$

$$blue_{out} = red_{in} * c_6 + green_{in} * c_7 + blue_{in} * c_8$$

To disable the cross-talk module completely, the identity matrix must be programmed:

$$\begin{matrix} c_0 & c_1 & c_2 & 1.0 & 0 & 0 & 4096 & 0 & 0 \\ c_3 & c_4 & c_5 & 0 & 1.0 & 0 & 0 & 4096 & 0 \\ c_6 & c_7 & c_8 & 0 & 0 & 1.0 & 0 & 0 & 4096 \end{matrix}$$

Example:

```
→ 1 video wb color_cross 4096 0 0 0 4096 0 0 0 4096
← OK
```

7.5.5.2.1 video wb color_cross offset

Command	video wb color_cross offset <c0> ... <c8>		
Alias	color_cross_offset	Getter Function	Yes
Description	Fine tune the cross-talk matrix by applying offsets. The offsets are added to the matrix coefficients which result from the selected white balance temperature or preset. The offset matrix consists of 9 coefficients which are in signed 4.12 fixed point format with a range of [-8.0 = -32768, 8.0 = 32767]. To use the original (calibrated) values, set all offsets to the default value of 0. The offsets are reset to 0 when a preset is selected using the <code>wb preset</code> command or if the auto white balance is run. The offsets cannot be changed while auto white balance is enabled.		
Parameter	c0 ... c8		
Type	Signed 4.12 Fixed Point Number		
Description	Color cross-talk matrix coefficient offsets.		
Min	-8.0 = -32768		
Max	7.999 = 32767		
Default	Values for color temperature 6500K, depends on camera calibration.		

7.5.5.3 video wb temp

Command	video wb temp <color_temp>		
Alias	wb_temp	Getter Function	Yes
Description	Set white balance temperature in Kelvin. If a custom color temperature is configured by manually changing the white balance gains or color cross-talk matrix or their offsets, this command will return the last known temperature setting. When auto white balance is enabled, this command returns the temperature which is currently set by the auto algorithm and manually setting it is not possible. Changing the temperature preserves the gain and matrix offsets. To return to a clean state they must be set to 0 manually using the 'gain offset' and 'color_cross offset' commands.		
Parameter	color_temp		
Type	Unsigned Integer		
Description	Color temperature in Kelvin.		
Min	3000		
Max	10000		
Default	6500		

7.5.5.4 video wb preset

Command	video wb preset <id>		
Alias	wb_preset	Getter Function	No
Description	Set white balance preset. To get supported presets call command without arguments. This command is for compatibility with the ProVideo protocol, it is recommended to use the <code>wb temp</code> command instead which allows for continuous white balance control. Setting a preset is not possible when auto white balance is enabled. Selecting a preset automatically resets the gain and matrix offsets to 0.		

Parameter	id
Type	Unsigned Integer
Description	White balance preset ID.
Valid Values	0 = Tungsten 3200 (3200K) 1 = Fluorescent TL84 (4200K) 2 = Flash (5000K) 3 = Daylight D55 (5500K) 4 = Daylight D60 (6000K) 5 = Daylight D65 (6500K) 6 = Shady (7500K) 7 = Cloudy (8500K) 8 = Blue Sky (10000K)
Default	5 = Daylight D65 (6500K)

When called without arguments it prints a list of all supported white balance presets:

```

→ 1 wb_preset
← 0= Tungsten 2800 (2800K)
← ...
← 9= Blue Sky (10000K)
← OK
  
```

Note that the command name is not printed (it is not a typical getter command) and that there are no spaces in front of the “=” and between the description and the “(“. This is according to the ProVideo protocol.

7.5.5.5 video wb auto

Command	video wb auto <enable>		
Alias	awb	Getter Function	Yes
Description	Enable continuous auto white balance. The camera will automatically determine the current color temperature and adjust the white balance gains and color cross matrix accordingly. The detected color temperature and resulting settings can be read back using the <code>wb temp</code> , <code>wb gain</code> and <code>wb color_cross</code> commands, it is not possible to set them manually while the automatic is enabled. Enabling the AWB resets the gain and matrix offsets to 0 and they cannot be changed. Use the <code>auto offset</code> command instead to configure a color temperature offset. When auto white balance is switched from on to off the color temperature stays at the last measured value. To run single-shot white balance or configure AWB behavior, see subcommands.		
Parameter	enable		
Type	Boolean		
Description	Enable or disable continuous auto white balance.		
Default	Enabled		

7.5.5.5.1 video wb auto speed

Command	video wb auto speed <speed>		
Alias	awb_speed	Getter Function	Yes
Description	Set control speed of the auto white balance algorithm: 0: Slow (default) 1: Medium 2: Fast		
Parameter	speed		
Type	Unsigned Integer		
Description	Auto white balance control speed.		
Allowed Values	0: Slow 1: Medium 2: Fast		
Default	0: Slow		

7.5.5.5.2 video wb auto offset

Command	video wb auto offset <offset>		
Alias	awb_offset	Getter Function	Yes
Description	Set color temperature offset in Kelvin which is added to the color temperature which is measured by the auto white balance algorithm. A negative offset results in a blueish output image, a positive offset in a reddish image. The default is 0 (neutral).		
Parameter	offset		
Type	Signed Integer		
Description	Color temperature offset in Kelvin.		
Min	-2000		
Max	2000		
Default	0		

7.5.6 video bpc

Command	video bpc		
Alias	bpc	Getter Function	Yes
Description	Show current bad pixel calibration status: 0: Uncalibrated 1: Calibrated		

PROTON cameras do not have an on-the-fly bad pixel detection mechanism as these typically find a lot of false positives which degrades the image quality. Instead, a static calibration is used.

Each camera is calibrated during production. If new defects become visible during operation, you can re-run the calibration anytime using the `video bpc calibrate` command described below.

7.5.6.1 video bpc calibrate

Command	video bpc calibrate		
Alias	dpc_auto_load	Getter Function	No
Description	Run automatic bad pixel calibration. For correct results this must be done with the lens cap installed! The calibration data is considered critical, so it is not cleared by the <code>settings reset</code> command.		

Caution: For the calibration to work properly it is crucial that the lens cap is installed, otherwise the detected bad pixels will not be correct!

Note: Although `dpc` would be a legal shorthand of the `dpc_auto_load` alias (since it is unambiguous) the camera will not accept it to avoid confusion with the `dpc` command from the ProVideo protocol which is not supported by PROTON OS.

7.5.7 video flare

Command	video flare <red/all> (<green> <blue>)		
Alias	flare	Getter Function	Yes
Description	Set RGB flare compensation values. All values are 16 bit unsigned integers with a range of [0, 65535]. If only one value is given, all offsets are set to the same value. Otherwise, all three values must be given. To set a single value use the subcommands.		
Parameter	red/all	green	blue
Type	Unsigned Integer		
Description	Red or all components compensation value.	Optional green compensation value.	Optional blue compensation value.
Min	0		
Max	65535		
Default	0		

7.5.7.1 video flare red

Command	video flare red <offset>		
Alias	flare_red	Getter Function	Yes
Description	Set red flare compensation value as 16 bit integer.		
Parameter	factor		
Type	Unsigned Integer		
Description	Red component flare compensation value.		
Min	0		
Max	65535		
Default	0		

7.5.7.2 video flare green

Command	video flare green <offset>		
Alias	flare_green	Getter Function	Yes
Description	Set green flare compensation value as 16 bit integer.		
Parameter	factor		
Type	Unsigned Integer		
Description	Green component flare compensation value.		
Min	0		
Max	65535		
Default	0		

7.5.7.3 video flare blue

Command	video flare blue <offset>		
Alias	flare_blue	Getter Function	Yes
Description	Set blue flare compensation value as 16 bit integer.		
Parameter	factor		
Type	Unsigned Integer		
Description	Blue component flare compensation value.		
Min	0		
Max	65535		
Default	0		

7.5.8 video color_space

Command	video color_space <id>		
Alias	color_space	Getter Function	Yes
Description	Set color space. To list all color spaces, use the color_space list subcommand. Depending on the color space the RGB-to-YUV color conversion matrix is set.		
Parameter	id		
Type	Unsigned Integer		
Description	Color space to set.		
Valid Values	0 = BT.709 (HD / SDR) 1 = BT.2020 (UHD / HDR)		
Default	0 = BT.709 (HD / SDR)		

7.5.8.1 video color_space list

Command	video color_space list		
Alias	/	Getter Function	No
Description	List all color spaces.		

This command is mainly intended for interactive use. A controller can expect all modes to be available for every camera model.

7.5.9 video sdi_range

Command	video sdi_range <range>		
Alias	sdi_range	Getter Function	Yes
Description	Set SDI output range: 0 = Limited or Legal range: SDI standard. Y values are limited to the range [64, 940] and chroma values to [64, 960].		

	1 = Full range: Maximum dynamic. Y and chroma values are limited to the range [4, 1019]. Note that the output range is forced to full range if the <code>lut_mode</code> is set to S-Log3 as the standard requires it. You can still change the output range, but the setting will only be applied once S-Log3 mode is deactivated.
Parameter	<code>range</code>
Type	Unsigned Integer
Description	SDI range to set.
Valid Values	0 = Limited (SDI Standard) 1 = Full (Maximum Dynamic)
Default	0 = Limited

7.5.10 video filter

Command	<code>video filter <enable></code>		
Alias	<code>filter_enable</code>	Getter Function	Yes
Description	Enable detail (sharpening) and denoise post processing filters (0 = bypass, 1 = enabled). To achieve bypass behavior with the filter enabled, the detail level must be set to 10% and the denoise level to 0%.		
Parameter	<code>enable</code>		
Type	Boolean		
Description	Enable or bypass post processing filters.		
Default	On		

7.5.10.1 video filter detail

Command	<code>video filter detail <level></code>		
Alias	<code>filter_detail</code>	Getter Function	Yes
Description	Set detail (sharpening) level of the post processing filter from 0 to 100%.		
Parameter	<code>level</code>		
Type	Unsigned Integer		
Description	Detail level in %.		
Min	0		
Max	100		
Default	10		

7.5.10.2 video filter denoise

Command	<code>video filter denoise <level></code>		
Alias	<code>filter_denoise</code>	Getter Function	Yes
Description	Set denoise level of the post processing filter from 0 to 100%.		
Parameter	<code>level</code>		
Type	Unsigned Integer		
Description	Denoise level in %.		
Min	0		
Max	100		
Default	0		

7.5.11 video mcc

Command	<code>video mcc <enable></code>		
Alias	<code>mcc</code>	Getter Function	Yes
Description	Enable multi matrix color correction (0 = bypass, 1 = enabled).		
Parameter	<code>enable</code>		
Type	Boolean		
Description	Enable or bypass multi matrix.		
Default	Off		

For details on how the multi matrix color correction works, see description of the `mcc` mode and `mcc` phase commands below.

7.5.11.1 video mcc mode

Command	video mcc mode <mode>		
Alias	mcc_opmode	Getter Function	Yes
Description	Set MCC operating mode which defines the number of active phases. To get supported modes use the <code>mcc mode list</code> subcommand.		
Parameter	mode		
Type	Unsigned Integer		
Description	Set MCC mode which defines number of active phases.		
Valid Values	0 = 12 phases 1 = 16 phases 2 = 24 phases 3 = 32 phases		
Default	2 = 24 phases		

The multi matrix module splits the color of the input image into equally sized parts depending on the hue which we call phases. Below are examples for 12 and 24 phases. The numbers in the segments are the IDs of the according MCC phases.

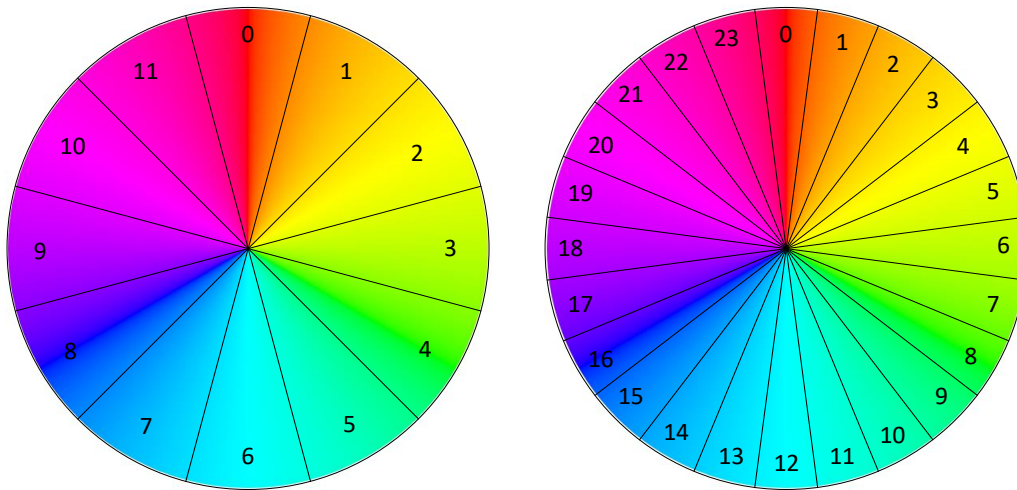


Figure 2: Example of 12 and 24 MCC Phases.

Note that phase 0 is always at the top in the red part of the color circle.

7.5.11.1.1 video mcc mode list

Command	video mcc mode list		
Alias	/	Getter Function	No
Description	List all MCC operational modes.		

This command is mainly intended for interactive use. A controller can expect all modes to be available for every camera model.

7.5.11.2 video mcc phase

Command	video mcc phase <id> (<saturation> <hue>)		
Alias	mcc_set	Getter Function	Yes
Description	Set one phase of the multi matrix consisting of a saturation and a hue value. The number of phases that can be set depends on the currently configured MCC mode. The saturation is an unsigned fixed point integer in 2.14 format, range [0.0 = 0, 3.999 = 35535], default: 1.0 = 16384. The hue is a signed fixed point integer in 1.15 format, range [-1.0 = -32768, 0.999 = 32767], default: 0.0 = 0. When saturation and hue are not specified the phase with the given ID is printed. When no parameter is given all active segments are printed.		
Parameter	id	saturation	hue

Type	Unsigned Integer	Unsigned 2.14 Fixed Point Number	Signed 1.15 Fixed Point Number
Description	ID of the phase to configure.	Saturation factor of given phase.	Hue angle of given phase.
Min	0	0.0 = 0	-1.0 = -32768 → -180°
Max	(Number of Phases) – 1 Example: 23 for 24 phases	3.999 = 65535	0.999 = 32767 → 179.999°
Default	/	1.0 = 16384	0.0 = 0 → 0°

Each phase controls one of the pie segments shown in Figure 2. The number of segments depends on the `mcc mode` setting.

For each phase / segment the saturation and hue can be adjusted independently. The saturation value is a factor by which the saturation is either increased (if >1.0) or decreased (if <1.0). The hue specifies the angle by which the color in the selected phase is shifted. Negative values shift the color counterclockwise, positive values clockwise. A value of +1.0 or -1.0 result in a +/- 180° rotation. The example below shows how the hue works for phase 0:

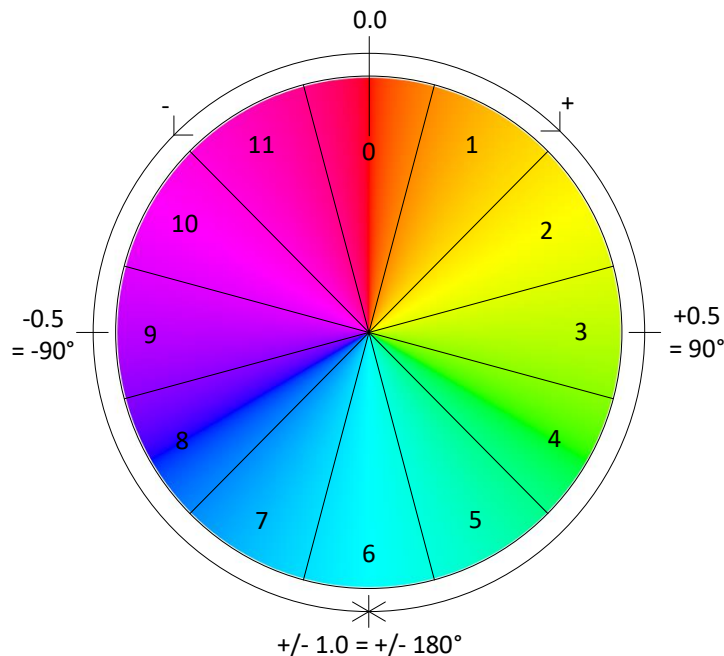


Figure 3: Hue Example for Phase 0.

7.5.11.3 video mcc blink

Command	video mcc blink <mask> (<period>)		
Alias	mcc_blink	Getter Function	Yes
Description	<p>Toggle blinking of masked multi matrix phases. Can be used to highlight areas in the image affected by certain phases.</p> <p>The mask parameter is a bit-mask where each bit represents one phase of the multi matrix.</p> <p>The optional period parameter defines the blink period in ms. The blink state will toggle every (period / 2) ms. If no period is specified, the default of 1s will be used, the minimum period is 100ms, the maximum 10s.</p> <p>This setting is not saved, that means after a reboot or power cycle blinking will always be disabled!</p>		
Parameter	mask	period	
Type	Unsigned Integer	Unsigned Integer	
Description	Each bit selects one phase for blinking.	Optional blinking period in ms.	
Min	0x00000000	100	
Max	Depends on number of active phases: 12 phases: 0x00000FFF 16 phases: 0x0000FFFF	10000	

	24 phases: 0x00FFFFFF 32 phases: 0xFFFFFFFF	
Default	0	1000

7.5.12 video_black_master

Command	video_black_master <red/all> (<green> <blue>)		
Alias	black_master	Getter Function	Yes
Description	Set all master black offsets. All values are 12 bit signed integers with a range of [-2048, 2047]. The offsets are added, that means positive values increase the black level and negative values decrease the black level. If only one value is given, all offsets are set to the same value. Otherwise, all three values must be given. To set a single offset value use the subcommands.		
Parameter	red/all	green	blue
Type	Signed Integer		
Description	Red or all components master black offset.	Optional green offset.	Optional blue offset.
Min	-2048		
Max	2047		
Default	0		

7.5.12.1 video_black_master red

Command	video_black_master red <offset>		
Alias	black_master_red	Getter Function	Yes
Description	Set red master black offset as 12 bit signed integer.		
Parameter	factor		
Type	Signed Integer		
Description	Red component master black offset.		
Min	-2048		
Max	2047		
Default	0		

7.5.12.2 video_black_master green

Command	video_black_master green <offset>		
Alias	black_master_green	Getter Function	Yes
Description	Set green master black offset as 12 bit signed integer.		
Parameter	factor		
Type	Signed Integer		
Description	Green component master black offset.		
Min	-2048		
Max	2047		
Default	0		

7.5.12.3 video_black_master blue

Command	video_black_master blue <offset>		
Alias	black_master_blue	Getter Function	Yes
Description	Set blue master black offset as 12 bit signed integer.		
Parameter	factor		
Type	Signed Integer		
Description	Blue component master black offset.		
Min	-2048		
Max	2047		
Default	0		

7.5.13 video_knee

Command	video_knee <enable> (<point> <slope>)		
Alias	knee	Getter Function	Yes

Description	Configure knee function for highlight limiting defined by the starting point in percent and the slope in degree. Until the starting point a 1:1 mapping of input to output value is used. After the starting point a root function is used. For a slope of 45° it also has a 1:1 mapping which equals bypass behavior. For angles <45° highlights get limited, for angles >45° highlights get boosted. If either point or slope shall be changed both must be specified, if only enable shall be changed both can be omitted.		
Parameter	enable	point	slope
Type	Boolean	Unsigned Integer	Unsigned Integer
Description	Enable or bypass knee module.	Knee starting point in percent.	Knee angle in degree.
Min	0 = Bypass	0	0
Max	1 = Enable	100	90
Default	0 = Bypass	85	45

The following figure visualizes how the knee function works:

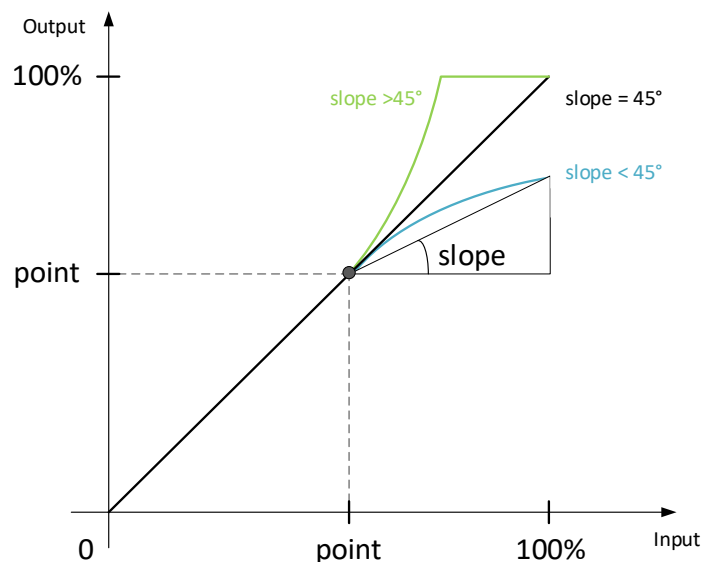


Figure 4: Knee Function Diagram.

Up to the knee `point` the output function is linear with a 1:1 mapping. From there on the function depends on the `slope` which describes the opening angle of the triangle that connects the knee point and output value at 100% input value.

- For a `slope` of 45° the output stays linear with a 1:1 mapping causing bypass behavior.
- The blue line shows a `slope` <45° which limits highlights and reduces the maximum output value.
- The green line shows a `slope` >45° which boosts highlights and causes big input values to get clipped to white.

Typical values for the knee `point` range from 60% to 90% while the `slope` is usually set to 20° to 40° for highlight limiting.

7.5.14 video lut

Command	video lut (<index>) <enable>		
Alias	lut_enable	Getter Function	Yes
Description	Enable gamma LUT (0 = bypass, 1 = enabled). The gamma LUT transforms the linear RGB signal into the gamma space that is expected by a monitor or recorder.		

	If called via the alias <code>lut_enable</code> this function expects two arguments: Index and enable flag. Since only one LUT is supported, the index is always 0. This is done for compatibility with the ProVideo protocol.	
Parameter	<code>index</code>	<code>enable</code>
Type	Unsigned Integer	Boolean
Description	Optional LUT index for compatibility with ProVideo protocol.	Enable or bypass gamma LUT.
Valid Values	0	0 / 1
Default	/	1 = On

Examples with and without the optional index that are functionally identical:

```
→ 1 video lut 1
← OK
→ 1 lut_enable 0 1
← OK
```

7.5.14.1 video lut mode

Command	<code>video lut mode <mode></code>		
Alias	<code>log_mode</code>	Getter Function	Yes
Description	Set LUT mode. To list all modes, use the <code>lut mode list</code> subcommand. Note that in S-Log3 mode the SDI output range (see <code>video sdi_range</code> command) is forced to full range as this is required by the standard. When a different mode is selected the previous SDI range setting is restored.		
Parameter	<code>mode</code>		
Type	Unsigned Integer		
Description	Gamma LUT mode.		
Valid Values	0 = REC.709 1 = BT.2100 HLG 2 = BT.2100 PQ 3 Sony S-Log3		
Default	0 = REC.709		

For **SDR content** the default REC.709 gamma curve is used. For **HDR productions** you can select from HLG, PQ or S-Log3 gamma curves which have a higher dynamic compression.

While the HLG curve is fixed you can make adjustments in the other modes:

- REC.709: Adjust gamma value (`lut gamma` subcommand).
- PQ: Set maximum display brightness (`lut max_brightness` subcommand).
- S-Log3: Set master gain (`lut master_gain` subcommand).

7.5.14.1.1 video lut mode list

Command	<code>video lut mode list</code>		
Alias	/	Getter Function	No
Description	List all LUT operational modes.		

This command is mainly intended for interactive use. A controller can expect all modes to be available for every camera model.

7.5.14.2 video lut gamma

Command	<code>video lut gamma <value></code>		
Alias	<code>lut_fast_gamma</code>	Getter Function	Yes
Description	Only applicable if <code>lut mode</code> is set to 0 = REC.709. Set gamma value of gamma curve. The value includes the scaling factor 1000 which results in a range of [1100, 3000] which equals [1.1, 3.0] in floating point numbers. The default is 2222 = 2.222 which results in a gamma curve according to the REC.709 standard. A bigger value results in a steep gamma curve, a smaller value in a flat curve.		

Parameter	value
Type	Unsigned Integer
Description	REC.709 gamma value.
Min	1.1 = 1100
Max	3.0 = 3000
Default	2.222 = 2222

7.5.14.3 video lut max_brightness

Command	video lut max_brightness <value>		
Alias	pq_max_brightness	Getter Function	Yes
Description	Only applicable if lut mode is set to 2 = PQ. Set maximum display brightness of the PQ curve in cd/ m ² . Valid range [100, 10000], default: 1000. The PQ curve uses an absolute mapping of output value to display brightness, that means the full SDI code range will be used for the maximum setting of 10000 cd/ m ² . For lower settings the curve will be compressed so that highlights are mapped to the desired maximum brightness (which does reduce the used SDI code range!).		
Parameter	value		
Type	Unsigned Integer		
Description	Maximum display brightness in cd/m ²		
Min	100		
Max	10000		
Default	1000		

7.5.14.4 video lut master_gain

Command	video lut master_gain <value>		
Alias	slog3_master_gain	Getter Function	Yes
Description	Only applicable if lut mode is set to 3 = S-Log3. Set master gain of the S-Log3 curve in dB. Valid range [-6, 12], default: 0. The master gain compresses (or expands) the gamma curve to decrease or increase the output code values. The full SDI code range will be used for 10 dB but most workflows use 0 or -3 dB. For everything >10 dB highlights will be clipped.		
Parameter	value		
Type	Unsigned Integer		
Description	S-Log3 master gain in dB.		
Min	-6		
Max	12		
Default	0		

7.5.14.5 video lut mode_compat

Command	video lut mode_compat <mode>		
Alias	lut_mode	Getter Function	Yes
Description	This command is only for compatibility with the ProVideo protocol to provide the lut_mode alias. The only allowed mode is '1' which is equal to "fast gamma" mode in the ProVideo protocol.		
Parameter	mode		
Type	Unsigned Integer		
Description	ProVideo LUT mode for compatibility.		
Valid Values	1 = Fast Gamma		
Default	1 = Fast Gamma		

7.5.15 video post

Command	video post		
Alias	/	Getter Function	No
Description	Configure post processing, see subcommands for details.		

This command has no functionality, it is only used to group the post-processing subcommands (see below).

7.5.15.1 video post brightness

Command	video post brightness <value>		
Alias	post_bright	Getter Function	Yes
Description	Set post processing brightness value in signed 1.7 fixed point format. Valid range [-128, 127] = [-1.0, 1.0), default (neutral) is 0 = 0.0. Instead of changing the brightness during post processing it is recommended to change exposure or gain for improved image quality.		
Parameter	value		
Type	Signed 1.7 Fixed Point Number		
Description	Brightness factor.		
Min	-1.0 = -128		
Max	0.999 = 127		
Default	0.0 = 0		

7.5.15.2 video post contrast

Command	video post contrast <value>		
Alias	post_cont	Getter Function	Yes
Description	Set post processing contrast value in unsigned 1.7 fixed point format. Valid range [0, 255] = [0.0, 2.0), default (neutral) is 128 = 1.0. Instead of changing the contrast during post processing it is recommended to turn on the flare compensation for improved image quality.		
Parameter	value		
Type	Unsigned 1.7 Fixed Point Number		
Description	Contrast factor.		
Min	0.0 = 0		
Max	1.999 = 255		
Default	1.0 = 128		

7.5.15.3 video post saturation

Command	video post saturation <value>		
Alias	post_sat	Getter Function	Yes
Description	Set post processing saturation value in 1.7 fixed point format. Valid range [0, 255] = [0.0, 2.0), default (neutral) is 128 = 1.0.		
Parameter	value		
Type	Unsigned 1.7 Fixed Point Number		
Description	Saturation factor.		
Min	0.0 = 0		
Max	1.999 = 255		
Default	1.0 = 128		

7.5.15.4 video post hue

Command	video post hue <value>		
Alias	post_hue	Getter Function	Yes
Description	Set post processing hue value in signed 1.7 fixed point format. Valid range [-128, 127] = [-1.0, 1.0) = [-90°, 90°), default (neutral) is 0 = 0.0 = 0°.		
Parameter	value		
Type	Signed 1.7 Fixed Point Number		
Description	Brightness factor.		
Min	-1.0 = -128 → -90°		
Max	0.999 = 127 → 90°		
Default	0.0 = 0 → 0°		

7.5.16 video osd

Command	video osd <mode>		
Alias	osd	Getter Function	Yes
Description	Set on-screen display mode. To get a list of supported modes use the <code>osd list</code> subcommand.		
Parameter	mode		
Type	Unsigned Integer		

Description	Gamma LUT mode.
Valid Values	0 = Disabled 1 = Show PROTON logo in top-right corner
Default	0 = Disabled

Note that while the command syntax is identical to the ProVideo protocol, the available OSDs modes differ.

7.5.16.1 video osd list

Command	video osd list		
Alias	/	Getter Function	No
Description	List all OSD modes.		

This command is mainly intended for interactive use. A controller can expect all modes to be available for every camera model.

8 Alias Reference

This chapter lists all the available command alias which also provide compatibility with the ProVideo protocol. Note that some aliases are not part of the ProVideo protocol but have been added for convenience.

The list below can also be generated by the camera with the `alias` command.

Table 7: Command alias overview.

Alias	Full Command Name	ProVideo Command
cam_gain	camera gain	Yes
cam_exposure	camera exposure	Yes
gain_mode	camera gain_mode	No
cam_info	camera info	Yes
aec	camera auto	Yes
aec_mode	camera auto mode	No
aec_target	camera auto target	No
aec_speed	camera auto speed	No
aec_max_gain	camera auto max_gain	No
aec_anti_flicker	camera auto anti_flicker	No
aec_custom	camera auto custom	No
aec_weight	camera auto weight	Yes
save_settings	settings save	Yes
load_settings	settings load	Yes
reset_settings	settings reset	Yes
auto_save	settings auto_save	Yes
version	system info	Yes
name	system name	Yes
runtime	system runtime	Yes
reboot	system reboot	Yes
update	system update	No
identify	system identify	Yes
error	system error	No
volatile	system volatile	Yes
rs485	system rs485	No
prompt	system rs485 mode	Yes

Alias	Full Command Name	ProVideo Command
controller	system rs485 mode 0	No
interactive	system rs485 mode 1	No
rs485_addr	system rs485 device_address	Yes
rs485_bc_addr	system rs485 broadcast_address	Yes
rs485_bc_master	system rs485 broadcast_master	Yes
rs485_baud	system rs485 baudrate	Yes
status_led	system status_led	No
tally	system tally	No
temp	system temp	Yes
max_temp	system temp max	Yes
max_temp_reset	system temp reset	Yes
over_temp_count	system temp count	Yes
audio_enable	system audio	Yes
audio_gain	system audio gain	Yes
timecode_enable	system timecode	No
timecode	system timecode value	Yes
timecode_hold	system timecode pause	Yes
timecode_drop	system timecode drop	No
video_mode	video mode	Yes
flip	video flip	Yes
black_sensor	video black_sensor	No
black_red	video black_sensor red	Yes
black_green	video black_sensor green	Yes
black_blue	video black_sensor blue	Yes
lsc	video lsc	Yes
lsc_preset	video lsc preset	No
wb	video wb	Yes
wb_gain	video wb gain	No
gain_red	video wb gain red	Yes
gain_green	video wb gain green	Yes
gain_blue	video wb gain blue	Yes
wb_gain_offset	video wb gain offset	No
gain_red_offset	video wb gain offset red	No
gain_green_offset	video wb gain offset green	No
gain_blue_offset	video wb gain offset blue	No
color_cross	video wb color_cross	Yes
color_cross_offset	video wb color_cross offset	No
wb_temp	video wb temp	No
wb_preset	video wb preset	Yes
awb	video wb auto	Yes
awb_speed	video wb auto speed	Yes
awb_offset	video wb auto offset	No
bpc	video bpc	No
dpc_auto_load	video bpc calibrate	Yes

Alias	Full Command Name	ProVideo Command
flare	video flare	Yes
flare_red	video flare red	No
flare_green	video flare green	No
flare_blue	video flare blue	No
color_space	video color_space	Yes
sdi_range	video sdi_range	Yes
filter_enable	video filter	Yes
filter_detail	video filter detail	Yes
filter_denoise	video filter denoise	Yes
mcc	video mcc	Yes
mcc_opmode	video mcc mode	Yes
mcc_set	video mcc phase	Yes
mcc_blink	video mcc blink	Yes
black_master	video black_master	Yes
black_master_red	video black_master red	No
black_master_green	video black_master green	No
black_master_blue	video black_master blue	No
knee	video knee	Yes
lut_enable	video lut	Yes
log_mode	video lut mode	Yes
lut_fast_gamma	video lut gamma	Yes
pq_max_brightness	video lut max_brightness	Yes
slog3_master_gain	video lut master_gain	Yes
lut_mode	video lut mode_compat	Yes
post_bright	video post brightness	Yes
post_cont	video post contrast	Yes
post_sat	video post saturation	Yes
post_hue	video post hue	Yes
osd	video osd	Yes

Appendix A: Software Licenses

PROTON OS makes use of software licensed under the following conditions.

Apache License Version 2.0

Parts of PROTON OS are distributed under the Apache License Version 2.0:

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and

wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

“Fast Approx” Library

PROTON OS uses the “Fast Approx” C-code library by Paul Mineiro that provides approximated versions of popular math functions.

The source code is available for download here: <https://github.com/pmineiro/fastapprox/tree/master>

And can be used under the following license:

Copyright (c) 2011, Paul Mineiro
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Paul Mineiro nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Appendix B: Document Revision History

Date	Revision	Chapter	Changes
29. May. 2024	v1.0.0	All	Initial release.
14. Jun. 2024	V1.0.1	3	Reworked firmware update chapter for usage of the new PROTON Updater GUI which is now described in the new chapter 3.3.
		6.1	Updated list of critical settings.
		6.2, 7.2	Updated auto-save chapter and descriptions for the <code>settings reset</code> and <code>auto_save</code> commands: Auto-save is now done immediately after changing a setting.
		7.3	Updated command descriptions for the <code>system runtime</code> , <code>temp max</code> , <code>temp reset</code> , <code>temp count</code> , and <code>error</code> commands: Runtime and temperature logging are no longer persistent.
		7.3.5	Added <code>system update</code> command description.
		7.3.13.1	Updated default <code>audio gain</code> from 1.0 to 0.5.
		8	Updated alias reference.
11. Jul. 2024	V1.1.0	2.3	Updated maximum temperature, is now 100°C.
		3.3	Updated firmware update GUI chapter.
		7.4.5	Added chapter for new auto exposure control commands.
		7.5.5	Added auto white balance commands in section 7.5.5.5 and updated <code>wb</code> , <code>wb temp</code> , <code>wb gain</code> , <code>wb color_cross</code> and <code>wb preset</code> commands descriptions.
		7.3.9.5	Added new supported baud rates: 230400 and 250000 baud.
		7.3.13.1	Changed default <code>audio gain</code> from 0.5 to 1.0.
		8	Updated alias reference.
22. Jul. 2024	V1.1.1	2.2	Improve description of error state.
		3	Updated firmware update chapter.
		5.2.4	Added description of error code “-140”.
		7.3.5	Updated description of <code>system update</code> command.
		7.5.4	Added <code>lsc preset</code> command to LSC chapter.
		8	Updated alias reference.
16. Aug. 2024	V1.1.2	All	Fixed typos, no major changes.
10. Sep. 2024	V1.2.0	2.2	Improved description of error state.
		2.3	Improved description of the over temperature protection. New shutdown temperature is 90°C, restart temperature is 85°C.
		7.4.3, 7.4.4	The <code>low_light</code> command has been replaced with the <code>gain_mode</code> command. Changing the gain mode may also change the maximum gain value, so the <code>camera info</code> command description has been updated.
		7.4.5.1	Added missing description of combined auto exposure mode and improved description of other modes.
		7.5.3	Removed <code>black_sensor auto</code> command, it is no longer supported.
		7.5.5	Added new <code>wb gain offset</code> and <code>wb color_cross offset</code> commands and updated description of the other white balance commands accordingly. Note: The minimum color temperature of the <code>wb temp</code> command has been increased from 2400 to 3000 Kelvin and the 2800 Kelvin preset has been removed from the <code>wb preset</code> command.
		8	Updated alias list.